Developing a Cooperative Intrusion Detection System for Wireless Sensor Networks

Lionel Besson, Philippe Leleu
Thales Communications France
160, boulevard de Valmy – BP 92
92704 Colombes Cedex
FRANCE

Lionel.BESSON@fr.thalesgroup.com / Philippe.LELEU@fr.thalesgroup.com

ABSTRACT

Wireless sensor networks (WSNs) are quickly gaining popularity because they are potentially low-cost solutions that can be used in a variety of application areas. However, they are highly susceptible to attacks and it is very probable that an intruder catches already existing security measures out. AWISSENET (Ad-hoc personal area network & WIreless Sensor SEcure NETwork) is a project funded by the European Union Information and Communication Technologies Program that is focused on security and resilience across ad-hoc personal area networks and wireless sensor networks, and provides a security toolbox for trusted route selection, secure service discovery and intrusion detection. This paper deals with intrusion detection systems for WSNs and how it is used in the AWISSENET project.

1.0 WIRELESS SENSOR NETWORKS AND SECURITY

1.1 Introduction

An ad-hoc wireless sensor network (WSN) is a network made of a large number of simple and low-cost devices called sensor nodes which are monitoring physical or environmental conditions like temperature, sound, pressure, etc using the ad hoc wireless multi-hop media to communicate these measurements to a base station. This cheap and efficient solution can be used in many military and civilian application areas including emergency response, homeland security and environmental monitoring. The open and distributed nature of the network, as well as the limited resources of the nodes makes WSNs highly vulnerable to attacks. Intrusion detection systems (IDS) act as a second line of defence when an intruder might deceive the other security solutions. In this paper, we explain the approach we have followed to develop the AWISSENET [1] distributed IDS.

1.2 Attackers Goals and Security Requirements

The attacker goals regarding WSNs can be multiple, depending on how easy it is for him to launch an attack, and the kind of damages he wants to inflict to the network. Moreover, some attacks can be seen as early steps to wider attacks that rely on some prerequisites. Among them, we can mention overhearing data (especially easy if communications are not or weakly encrypted, but can also be used for traffic analysis attacks [10]), injecting fake data (in order to fake the measurements, or attack the network protocols), reduce the performance of the network (which already has limited resources), breaking parts of the network links or damaging the whole network operation (usually done via the routing protocol).

A secure WSNs should be robust and reliable (the failure of a small set of nodes should not break the entire security of the network), but also ensure data authenticity, integrity, confidentiality and freshness.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **NOV 2010**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
Developing a Cooperative Intrusion Detection System for Wireless Sensor Networks

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Thales Communications France 160, boulevard de Valmy BP 92 92704 Colombes Cedex FRANCE

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADA564697. Information Assurance and Cyber Defence (Assurance de l’information et cyberdefense). RTO-MP-IST-091

14. ABSTRACT

Wireless sensor networks (WSNs) are quickly gaining popularity because they are potentially low-cost solutions that can be used in a variety of application areas. However, they are highly susceptible to attacks and it is very probable that an intruder catches already existing security measures out. AWISSENET (Ad-hoc personal area network & WIreless Sensor SEcure NETwork) is a project funded by the European Union Information and Communication Technologies Program that is focused on security and resilience across ad-hoc personal area networks and wireless sensor networks, and provides a security toolbox for trusted route selection, secure service discovery and intrusion detection. This paper deals with intrusion detection systems for WSNs and how it is used in the AWISSENET project.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**SAR**	**8**	

1.3 Securing Wireless Sensor Networks

They are two main approaches for securing a WSN: adapt the existing protocols to counter the possible attacks, or use existing security frameworks that provide security functions.

Securing existing protocols usually means less integration work, but you need to consider the security of each protocol your network relies on in order to achieve the global security of the network. The secured version of the AODV protocol (SAODV, [14]) is an example of such an approach.

Security frameworks aim at providing a generic security package that covers the basic security needs for WSNs and can be integrated into sensor network applications. The protocols must be adapted to use these frameworks. TinySec [11], ZigBee [12] or MiniSec [13] enjoy significant attention in the community.

As presented in the introduction, the open and distributed nature of communications of WSNs makes it impossible to concentrate all security functions in a central point, so that each node of the network needs to execute several security functions. Combined with the limited resources of sensor nodes, this requires to carefully considering the cost of the security mechanisms that are deployed. As a consequence, it is very probable that a node gets compromised or a fake node forged at some point. Intrusion Detection Systems for WSNs act as a second line of defence. Their role is to detect attacks before they are successful and compromise the security of the network, and to expel intruders and compromised nodes from the network.

2.0 INTRUSION DETECTION FOR WSNS

2.1 Specificities and Challenges

IDS for sensor networks differ in many ways from the one used in legacy networks. The challenges that IDS have to take up in the particular field of WSNs include:

- *Automated decision:* nodes must be truly autonomous and adapt to the evolution of the network and environment.
- *Limited resources:* security functions must take into account the scarce bandwidth, memory, energy and computational power.
- *Localize auditing:* a node can only see what is happening in its immediate neighborhood.
- *No node is trustworthy:* nodes can be quite easily compromised, and should not be trusted.
- *Distributed IDS:* intrusion detection must happen on several nodes in order to detect distributed attacks.
- *Security of the IDS itself:* malicious nodes should not be able to deceive the IDS.

2.2 Network Architecture

Usual IDS are typically *stand-alone IDS*, where each node runs an independent intrusion detector. This is particularly true for network-based intrusion detection systems, which often consist in a powerful server that has access to the whole traffic (Figure 1). Such systems cannot perform satisfyingly in WSNs, since local audit data are not enough to have a good comprehension of what is happening in the network. Cooperation between the different nodes is needed in order to achieve efficient detection.

Hierarchical IDS are systems where specific nodes are in charge of monitoring their neighbours, with various level of cooperation between cluster heads, as presented in [2]. They are particularly suited for multi-layered network architectures.

Distributed IDS meet the decentralized nature of ad-hoc wireless sensor networks, where each node is responsible for collecting local audit data, and share this knowledge globally in order to carry out a global intrusion detection system [3], [4].

Mobile Agent Based IDS use pieces of mobile code charged with a specific mission and sent to other nodes in order to analyse the local audit data of other nodes and bring back the results to the originator [5], or to run a specific attack detection on a node for distributing the detection tasks amongst the network [6].

The AWISSENET architecture is a hybrid one between the hierarchical and the distributed approach. The network is partitioned into several multi-hop clusters. Inside each cluster, and at the global level between cluster heads, we use a distributed architecture. The intrusions detections and assumptions, and the other IDS messages are exchanged inside a cluster, and the cluster members cooperatively take the decisions. The cluster head is then responsible for iterating the same process at the global network level. This approach enables more scalability, since having a completely distributed IDS would flood the network when they are too many nodes. It also minimizes the drawbacks of the hierarchical architectures by introducing a high cooperation between the nodes.

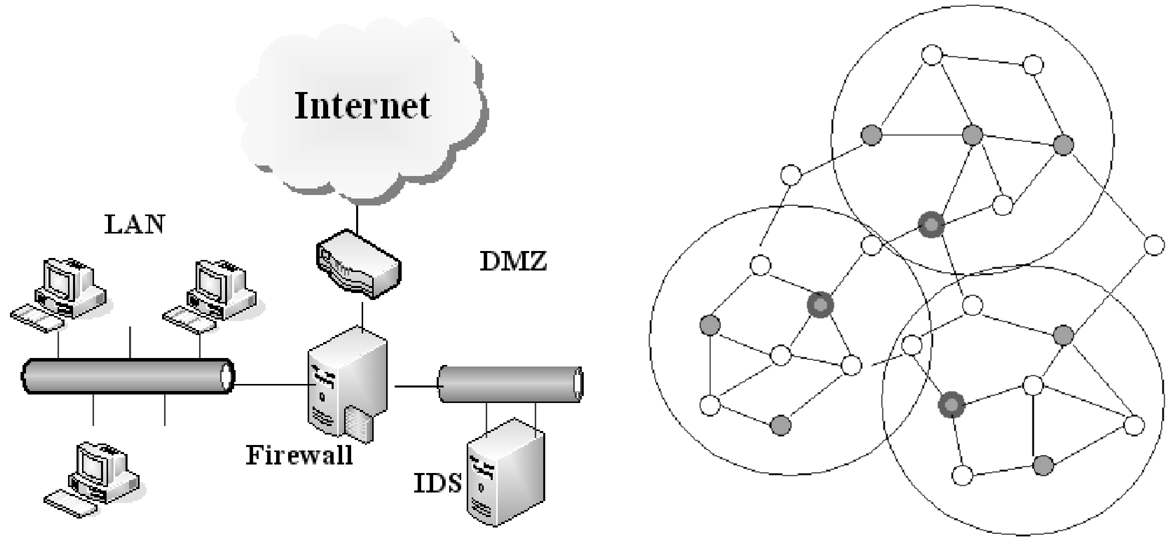

Figure 1 The left figure represents a typical IDS use in a classical infrastructure network. The right one illustrates the network of the AWISSENET distributed IDS. Gray nodes are IDS agents, circled ones are cluster heads.

2.3 Collecting Audit Data

Audit data are collected by local agents analysing local sources of information, which can be hardware or network based. The AWISSENET projects concentrates on the second one.

Hardware audit data include anti-tamper mechanisms, detect when a node is being reprogrammed or watch for abnormal sensor values, like accelerometer.

However, using the vulnerabilities of software (and especially the routing protocol) is often a simpler and easier way for an attacker to break into the network. Thus, the role of the distributed IDS is to analyze the overheard traffic and look for suspicious behaviours. Due to the ad-hoc nature of the network, a single node has only a partial knowledge of what is happening. However, they can still analyze the packets that are directly sent to them or exchanged between their neighbours, thus acting as “spontaneous watchdogs”.

Metrics can then be gathered about badly forwarded packets, or nodes flooding the network with route replies [7], or even more complex behaviours.

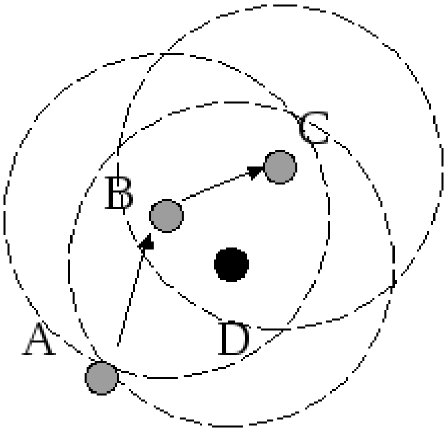

Figure 2 Example of a spontaneous watchdog: node D can watch if B is correctly forwarding messages from A to C

2.4 Intrusion Detection

IDS need to distinguish between normal and abnormal activities in order to detect attacks against the network before they are successful. Detection techniques are usually classified into three categories.

- *Misuse detection* (also known as signature-based detection) consists in comparing audit data with known attack patterns. This technique is the one mainly used for classical IDS, but is not widely suitable for WSNs, due to memory and processing power constraints. It also suffers from a lack of flexibility and is useless against previously unknown attacks.
- *Anomaly detection* systems describe the 'normal' behaviour of the network and detect any activity that differs significantly from it, and are thus potentially capable of detecting new attacks. The normal behaviour is usually established via automated training [8].
- *Specification-based* detection is similar to anomaly detection, but the correct behaviour of the network is manually defined. It allows a smaller rate of false alarms, but is less flexible with regards to the different environments.

Depending on the context (for example which routing protocol is used, or the services deployed in the network) and the capabilities of the heterogeneous nodes, different intrusion detection algorithms can be used in order to offer the best ratio between detection efficiency and resources consumption. The AWISSENET distributed IDS proposes a plug-in based architecture in order to enable an easy and flexible management of the algorithms running on each node. Some algorithms from the three kinds of detection techniques aforementioned have been implemented, amongst which**:**

- *Bad Protocol:* An attacker unaware of the services and protocols used in the network might try to launch fake nodes with widely used protocols, with the hope that the network will understand it. This is the simplest implemented detection algorithm.
- *Black Hole, Grey Hole, Selective Forwarding:* the attackers are selectively or randomly dropping some or all packets that they should be forwarding. An alert is raised if the ratio of non-forwarded packets by a specific node is big compared with the overall ratio of the watched nodes (experience has shown that false positives couldn’t be ignored)

- *Integrity attack:* This attack is launched by a node that selectively or randomly alters the packet that he forwards. Unlike black hole or grey hole attacks, such events are unlikely to be caused by the nature of communications.
- *Flooding attack:* An alert is raised when a node is sending specific messages at an unusual rate.
- *Replay attack:* IDS messages are checked for detecting nodes trying to replay IDS exchanges in order to lure the system into expelling a legitimate node from the network, or keep a malicious one undetected. It uses the mechanisms explained in 2.6.

2.5 Decision Making and Recovery

Once a local IDS agent has raised an alarm internally, the next question that arises is who is going to make the final decision that a node is truly an intruder or not. Independent Decision-making Systems are usually used in cluster-based architectures because they leave the decision that a node is effectively an intruder to specific nodes (usually the cluster heads) [2]. The alternative solution is called Cooperative Intrusion Detection Systems. When an attack seems to have been detected, the node appeals to neighbouring nodes in order to output a global decision.

The AWISSENET IDS uses its hybrid architecture in order to output a global decision on the status of a node. Alerts raised by the local IDS agents may be only assumptions that something abnormal is happening, with inconclusive evidence.

Leaving the decision to a single node would imply a high rate of false positives and false negatives, because the data collected locally is often not enough to output an appropriate intrusion detection decision. The AWISSENET uses an open vote mechanism to output a global decision from the DIDS.

Once a node has raised an alert, a voting mechanism is launched between the nodes belonging to the same cluster. The cluster head and a random node are elected as vote authorities, and each node has to send its response to them.

The vote authorities gather the votes to output the decision at the cluster level. The voting mechanism is authenticated but not ciphered, so that it allows the nodes to detect potential intruders trying to lure the IDS.

The voting mechanisms are then issued at the network level between the cluster head to output the global IDS decision. This decision (which can be the identification of an intrusion and / or an intruder, or simply a false alarm) is sent back to the nodes by the cluster heads. The intruder is then isolated from the network via the routing module, and if needed, cryptographic material is updated.

2.6 Secure IDS Exchanges

As the decisions taken by the IDS can expel a node from the network, it should be very careful not being compromised, and needs to ensure the integrity of the messages exchanged between the nodes and that they are sent by legitimate ones.

The AWISSENET DIDS uses timestamps and digests to secure the communications between the IDS agents, which are inspired by the secure OLSR plug-in [9]. Secret keys are shared inside each cluster and between the cluster heads and used to produce and check the digests of the messages. Timestamps are used to determine the freshness of the messages and prevent replay attacks. In order to securely synchronize (or re-synchronize if needed) clocks between two nodes, an exchange of timestamps is done with challenge-response messages. Any message received with an invalid timestamp or digest is then discarded by the IDS agent, and an alert is raised. This mechanism has been chosen in order to have reasonably secured messages without using too expensive cryptographic material.

$$A \rightarrow B : Ch_a\ D(M, K)$$

$$B \rightarrow A : Ch_b\ Ts_b\ D(Id_b, Ch_a, K)\ D(M, K)$$

$$A \rightarrow B : Ts_a\ D\ (Id_a, Ch_b, K)\ D(M, K)$$

Figure 3 Timestamp exchange between nodes A and B. Ch is a nonce challenge. Ts is a timestamp, D a digest function, K the shared key and Id the identifiers of the nodes.

3.0 RESOURCE USAGE AND IMPLEMENTATION

Our experiments have shown that the sensor motes currently available on the market have very limited resources, so that implementing a distributed intrusion detection system on it can be quite discouraging. Before being able to take into account the power consumption, the first challenge is simply to deal with the low memory of the nodes. The AWISSENET test-bed is mainly composed from motes running on TinyOs, which respectively only have 4KB and 8KB of RAM. As the project is also integrating modules for secure geographical routing and secure service discovery, the memory limitation is quickly exceeded. This implies to carefully choose the amount of audit data that is gathered, thus making intrusion detection more difficult. The AWISSENET IDS uses several mechanisms in order to reduce its resources usage.

The first approach has been to optimize the metrics gathering process with regards to the detection algorithms. The metrics shared by the different plug-ins are only collected and computed once, by using a specific module for managing audit data. We have implemented easy-deployment tools that take yaml configuration files as entries in order to load the program code on the motes. These configuration files define which algorithms and metrics are computed on which motes, but also some configuration items that make it possible to have the lowest possible memory used by the algorithms with regards to network characteristics like the frequency of messages, density of the nodes, etc. Only algorithms and metrics gathering processes effectively running on the node are compiled. The following figure shows the relationship between the snooping interface (used to gather audit data), the metrics (used to extract interesting data) and the detection algorithms (used to decide whether something abnormal is happening).

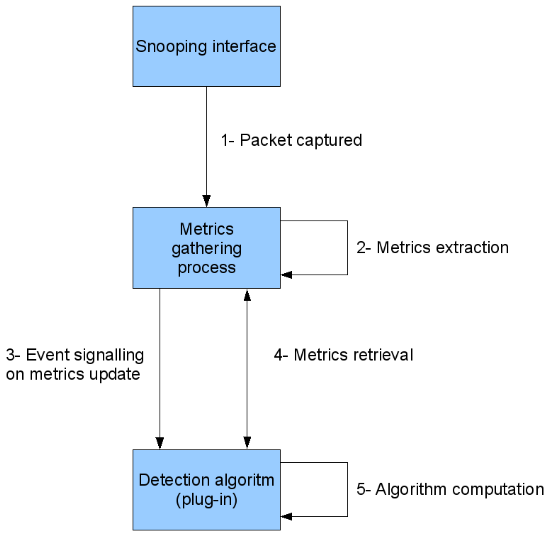

Figure 4 Relationship between algorithms and metrics collection

Anomaly detection algorithms should be the algorithms of choice, since they use less memory and are still efficient in detecting a wide variety of attacks. However, pattern-matching techniques should be used on a per-case basis; they are the most efficient when targeting specific attacks against specific protocols, like the routing one.

Applying the security measures of classical networks to WSNs is difficult, not to say impossible. Cryptographic operations and especially asymmetric cryptography are often very costly and should be avoided as much as possible in sensor networks. Moreover, nodes don't have enough memory to store the public keys of the other members when networks are built with hundreds of nodes. The AWISSENET IDS thus uses symmetric keys and timestamps in order to secure its exchanges.

Finally, the AWISSENET project is using low-cost, low-power programmable logic devices (CPLD, [15]) in order to implement the most resource demanding functions (and especially cryptographic ones) on hardware, thus speeding up the algorithms and reducing the power consumption up to 1 / 100th of that used by a microcontroller.

4.0 CONCLUSION

AWSNs impose new challenges on the design of IDS, which are especially needed due to the unattended operations in open environments. The network owner cannot simply rely on usual security mechanisms to ensure its security. We propose to implement a flexible and efficient intrusion detection system, which can then be used in a variety of wireless network and devices, and easily adapted to the resources available.

5.0 REFERENCES

[1] http://www.awissenet.eu

[2] Kachirski, R. Guba, D. Schwartz, S. Stoecklin, and E. Yilmaz: Case based agents for packet-level intrusion detection in ad hoc networks. In Proceedings of the 17th International Symposium on Computer and Information Sciences. CRC Press, October 2002, pp.315-320

[3] Stamouli: Real-time Intrusion Detection for Ad Hoc Networks. Master of Science dissertation, University of Dublin, 2003

[4] K. Ioannis, T. Dimitriou, and F. C. Freiling: Towards Intrusion Detection in Wireless Sensor Networks. 13th European Wireless Conference, Paris, April 1997

[5] P. Albers, O. Camp; J-M. Percher, B. Jouga, L. M, and R. Puttini: Security in Ad Hoc Networks, a General Intrusion Detection Architecture Enhancing Trust Based Approaches. In Proceedings of the 1st International Workshop on Wireless Information Systems, April 2002

[6] Y. Zhang, W. Lee, and Y. Huang: Intrusion Detection Techniques for Mobile Wireless Networks. ACM/Kluwer Wireless Networks Journal (ACM WINET), Vol. 9, No. 5, September 2003.

[7] R. Roman, J. Zhou, and J. Lopez: Applying Intrusion Detection Systems to Wireless Sensor Networks. Consumer Communications and Networking Conference, 2006, pp. 640-644

[8] V. Bhuse and A. Gupta: Anomaly intrusion detection in wireless sensor networks. Journal of High Speed Networks, Vol. 15, No. 1, pp. 33-51, 2006

[9] A. Hafslund, A. Tonnesen, R.B. Rotvik, J. Anderson, and O. Kure: Secure extension to the OLSR protocol. 1st OLSR Interop & Workshop, San Diego, 2004

[10] J. Deng, R. Han, and S. Mishra: Countermeasures Against Traffic Analysis Attacks in Wireless Sensor Networks. In Proceedings of the 1st Conference on Security and Privacy for Emerging Areas in Communications Networks, Athens, September 2005

[11] C. Karlof, N. Sastry, and D. Wagner: TinySec, A link layer security architecture for wireless sensor networks. In Proceedings of the 2nd ACM conference on Embedded Networked Sensor Systems, November 2004.

[12] ZigBee Alliance: ZigBee Specification. Technical Report Document 053474r06, June 2005.

[13] M. Luk, G. Mezzour, A. Perrig, and V. Gligor: MiniSec, A Secure Sensor Network Communication Architecture. In Proceedings of International Conference on information Processing in Sensor Networks, Cambridge, April 2007.

[14] M. Guerrero Zapata and N. Asokan: Securing Ad Hoc Routing Protocols. In Proceedings of the 2002 ACM Workshop on Wireless Security, pp. 1-10, September 2002.

[15] Xilinx: CoolRunner-II CPLD Family. Product specification, DS090 (v3.0), March 2007.