Wireless Sensor Data Fusion for Critical Infrastructure Security

Francesco Flammini[1 2], Andrea Gaglione[2], Nicola Mazzocca[2], Vincenzo Moscato[2], Concetta Pragliola[1]

[1] ANSALDO STS - Ansaldo Segnalamento Ferroviario S.p.A.
Via Nuova delle Brecce 260, Naples, Italy
{flammini.francesco, pragliola.concetta}@asf.ansaldo.it

[2] Università di Napoli "Federico II"
Dipartimento di Informatica e Sistemistica
Via Claudio 21, Naples, Italy
{frflammi, andrea.gaglione, nicola.mazzocca, vmoscato}@unina.it

Abstract. Wireless Sensor Networks (WSN) are being investigated by the research community for resilient distributed monitoring. Multiple sensor data fusion has proven as a valid technique to improve detection effectiveness and reliability. In this paper we propose a theoretical framework for correlating events detected by WSN in the context of critical infrastructure protection. The aim is to develop a decision support and early warning system used to effectively face security threats by exploiting the advantages of WSN. The research addresses two relevant issues: the development of a middleware for the integration of heterogeneous WSN (SeNsIM, Sensor Networks Integration and Management) and the design of a model-based event correlation engine for the early detection of security threats (DETECT, DEcision Triggering Event Composer & Tracker). The paper proposes an overall system architecture for the integration of the SeNsIM and DETECT frameworks and provides example scenarios in which the system features can be exploited.

1. Introduction

Several methodologies (e.g. risk assessment [5]) and technologies (e.g. physical protection systems [4]) have been proposed to enhance the security of critical infrastructure systems.

The aim of this work is to propose the architecture for a decision support and early warning system used to effectively face security threats (e.g. terrorist attacks) based on wireless sensors. Wireless sensors feature several advantages when applied to critical infrastructure surveillance [8], as they are:

- Cheap, and this allows for fine grained and highly redundant configurations;
- Resilient, due to their fault-tolerant mesh topology;

- Power autonomous, due to the possibility of battery and photovoltaic energy supplies;
- Easily installable, due to their wireless nature and auto-adapting multi-hop routing;
- Intelligent, due to the on-board processor and operating systems which allow for some data elaborations being performed locally.

All these features support the use of WSN in highly distributed monitoring applications in critical environments. The example application we will refer to in this paper is railway infrastructure protection against external threats which can be natural (fire, flooding, landslide, etc.) or human-made malicious (sabotage, terrorism, etc.).

Examples of useful sensors in this domain are listed in the following: smoke and heat – useful for fire detection; moisture and water – useful for flooding detection; Pressure – useful for explosion detection; movement detection (accelerometer or GPS based shifting measurement) – useful for theft detection or structural integrity checks; gas and explosive – useful for chemical or bombing attack detection; vibration and sound – useful for earthquake or crash detection. WSN could also be used for video surveillance and on-board intelligent video-analysis, as reported in [7].

Theoretically, any kind of sensor could be interfaced with a WSN, as it would just substitute the sensing unit of the so called "motes". For instance, it would be useful (and apparently easy) to interface on WSN intrusion detection devices (like RFID readers, volumetric detectors, active infrared barriers, microphonic cables, etc.) in order to save on cables and junction boxes and exploit an improved resiliency and a more cohesive integration. With respect to traditional connections based on serial buses, wireless sensors are also less prone to tampering, when proper cryptographic protocols are adopted [6]. However, for some classes of sensors (e.g. radiation portals) some of the features of motes (e.g. size, battery power) would be lost.

The heterogeneity of network topologies and measured data requires integration and analysis at different levels (see Figure 1).

As first, the monitoring of wide geographical areas and the diffusion of WSNs managed by different middlewares have highlighted the research problem of the integrated management of data coming from the various networks. Unfortunately such information is not available in a unique container, but in distributed repositories and the major challenge lies in the heterogeneity of repositories which complicates data management and retrieval processes. This issue is addressed by the SeNsIM framework [1], as described in Section 2.

Secondly, there is the need for an on-line reasoning about the events captured by sensor nodes, in order to early detect and properly manage security threats. The availability of possibly redundant data allows for the correlation of basic events in order to increase the probability of detection, decrease the false alarm rate, warn the operators about suspect situations, and even automatically trigger adequate countermeasures by the Security Management System (SMS). This issue is addressed by the DETECT framework [2], as described in Section 3.

The rest of the paper is organized as follows: Section 4 discusses about the SeNsIM and DETECT software integration; Section 5 introduces an example railway security application; Section 6 draws conclusions and hints about future developments.

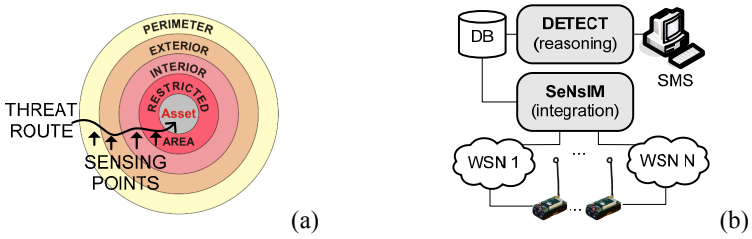

Figure 1. (a) Distributed sensing in physical security; (b) Monitoring architecture.

2. The SeNsIM framework

The main objectives of *SeNsIM* are:

- To integrate information from distributed sensor networks managed by *local middlewares* (e.g. *TinyDB*);
- To provide an unique interface for local networks by which a generic user can easily execute queries on specific sensor nodes;
- To ensure system's scalability in case of connection of new sensor networks.

From an architectural point of view, the integration has been realized by exploiting the *wrapper-mediator* paradigm: when a sensor network is activated, an apposite *wrapper* agent aims at extracting its features and functionalities and to send (e.g. in a XML format) them to one or more *mediator* agents that are, in the opposite, responsible to drive the querying process and the communication with users.

Thus, a query is first submitted through a user interface, and then analyzed by the mediator, converted in a standard XML format and sent to the apposite wrapper. The latter, in a first moment executes the translated query on its local network, by means of a low-layer middleware (*TinyDB* in the current implementation), and then retrieve the results to send (in a XML format) to the mediator, which show them to the user.

According to the data model, the wrapper agent provides a local network virtualization in terms of objects, network and sensors. An object of the class *Sensor* can be associated to an object of *Network* type. Moreover, inside the same network one or more sensors can be organized into objects of *Cluster* or *Group* type. The state of a sensor can be modified by means of classical getting/setting functions, while the measured variables can be accessed using the sensing function.

Figure 2 schematizes the levels of abstraction in the data management perspective provided by SeNsIM using TinyDB as low-level middleware layer and outlines the system architecture. The framework is described in more details in reference [1].

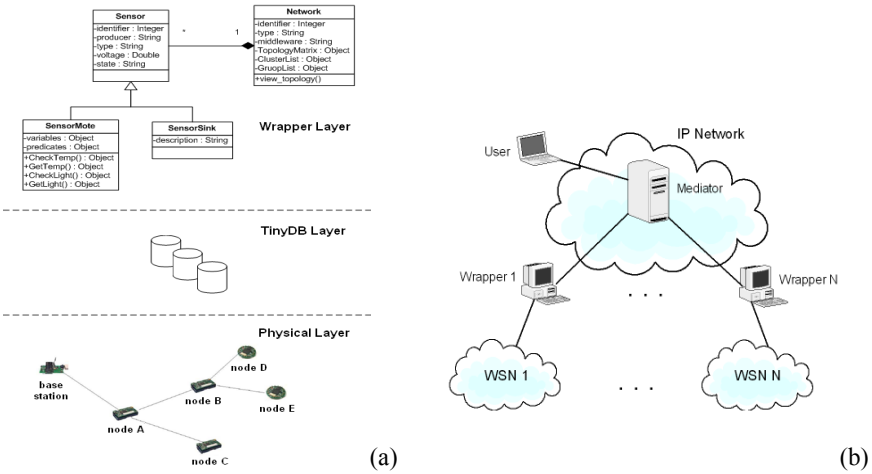

Figure 2. (a) Levels of abstraction; (b) SeNsIM architecture.

3. The DETECT framework

Among the best ways to prevent attacks and disruptions is to stop any perpetrators before they strike. DETECT is a framework aimed at the automatic detection of threats against critical infrastructures, possibly before they evolve to disastrous consequences. In fact, non trivial attack scenarios are made up by a set of basic steps which have to be executed in a predictable sequence (with possible variants). Such scenarios must be precisely identified during the risk analysis process. DETECT operates by performing a model-based logical, spatial and temporal correlation of basic events detected by sensor networks, in order to "sniff" sequence of events which indicate (as early as possible) the likelihood of threats. In order to achieve this aim, DETECT is based on a real-time detection engine which is able to reason about heterogeneous data, implementing a centralized application of "data fusion". The framework can be interfaced with or integrated in existing SMS systems in order to automatically trigger adequate countermeasures (e.g. emergency/crisis management).

Attack scenarios are described in DETECT using a specific Event Description Language (EDL) and stored in a Scenario Repository. Starting from the Scenario Repository, one or more detection models are automatically generated using a suitable formalism (Event Graphs in the current implementation). In the operational phase, a model manager macro-module has the responsibility of performing queries on the Event History database for the real-time feeding of detection model according to pre-determined policies.

When a composite event is recognized, the output of DETECT consists of: the identifier(s) of the detected/suspected scenario(s); an alarm level, associated to scenario evolution (only used in deterministic detection as a linear progress indicator);

a likelihood of attack, expressed in terms of probability (only used as a threshold in heuristic detection).

DETECT can be used as an on-line decision support system, by alerting in advance SMS operators about the likelihood and nature of the threat, as well as an autonomous reasoning engine, by automatically activating responsive actions, including audio and visual alarms, unblock of exit turnstiles, air conditioned flow inversion, activation of sprinkles, emergency calls to first responders, etc.

DETECT is depicted as a black-box in Figure 3 and described in more details in [2].

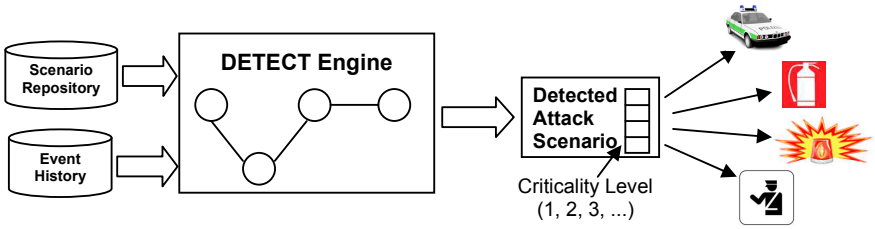

Figure 3. The DETECT framework.

4. Integration of SeNsIM and DETECT

The SeNsIM and DETECT frameworks need to be integrated in order to obtain an on-line reasoning about the events captured by different WSNs. As mentioned above, the aim is to early detect and manage security threats against critical infrastructures. In this section we provide the description of the sub-components involved in the software integration of SeNsIM and DETECT.

During the query processing task of SeNsIM, user queries are first submitted by means of a *User Interface*; then, a specific module (*Query Builder*) is used to build a query. The user queries are finally processed by means of a *Query Processing* module which sends the query to the appropriate wrappers. The partial and global query results are then stored in a database named *Event History*. All the results are captured and managed by a *Results Handler*, which implements the interface with wrappers.

The Model Feeder is the DETECT component which performs periodic queries on the Event History to access primitive event occurrences. The Model Feeder instantiates the inputs of the Detection Engine according to the nature of the model(s).

Therefore, the integration is straightforward and mainly consists in the management of the Event History as a shared database, written by the mediator and read by the Model Feeder according to an appropriate concurrency protocol.

In Figure 4 we report the overall software architecture as a result of the integration between SeNsIM and DETECT. The figure also shows the modules of SeNsIM involved in the query processing task. User interaction is only needed in the configuration phase, to define attack scenarios and query parameters. According to

the query strategy, both SeNsIM and DETECT can access data from the lower layers using either a cyclic or event driven retrieval process.

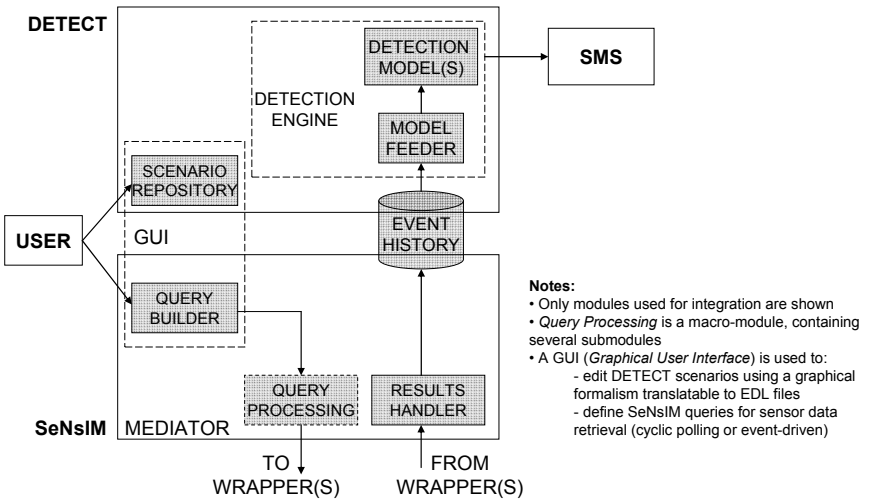

Figure 4. Query processing and software integration.

5. Example application scenario

In this section we report an example application of the overall framework to the case-study of a railway transportation system, an attractive target for thieves, vandals and terrorists. Several application scenarios can be thought exploiting the proposed architecture and several wireless sensors (track line break detection, on-track obstacle detection, etc.) and actuators (e.g. virtual or light signalling devices) could be installed to monitor track integrity against external threats and notify anomalies. In the following we describe how to detect a more complex scenario, namely a terrorist strategic attack.

Let us suppose a terrorist decides to attack a high-speed railway line, which is completely supervised by a computer-based control system. A possible scenario consisting in multiple train halting and railway bridge bombing is reported in the following:

1. Artificial occupation (e.g. by using a wire) of the track circuits immediately after the location in which the trains needs to be stopped (let us suppose a high bridge), in both directions.
2. Interruption of the railway power line, in order to prevent the trains from restarting using a staff responsible operating mode.
3. Bombing of the bridge shafts by remotely activating the already positioned explosive charges.

Variants of this scenarios exist: for instance, trains can be (less precisely) stopped by activating jammers to disturb the wireless communication channel used for radio signaling, or starting the attack from point (2) (but this would be even less precise). The described scenario could be early identified by detecting the abnormal events reported in point (1) and activating proper countermeasures. By using proper on-track sensors it is possible to monitor the abnormal occupation of track circuits and a possible countermeasure consists in immediately sending an unconditional emergency stop message to the train. This would prevent the terrorist from stopping the train at the desired location and therefore halt the evolution of the attack scenario. Even though the detection of events in points (2) and (3) would happen too late to prevent the disaster, it could be useful to achieve a greater situational awareness about what is happening in order to rationalize the intervention of first responders.

Now, let us formally describe the scenario using wireless sensors and detected events, using the notation "sensor description (sensor ID) :: event description (event ID)":

FENCE VIBRATION DETECTOR (S1) :: POSSIBLE ON TRACK INTRUSION (E1)
TRACK CIRCUIT X (S2) :: OCCUPATION (E2)
LINESIDE TRAIN DETECTOR (S3) :: NO TRAIN DETECTED (E3)
TRACK CIRCUIT Y (S4) :: OCCUPATION (E4)
LINESIDE TRAIN DETECTOR (S5) :: NO TRAIN DETECTED (E5)
VOLTMETER (S6) :: NO POWER (E6)
ON-SHAFT ACCELEROMETER (S7) :: STRUCTURAL MOVEMENT (E7)

Due to the integration middleware made available by SeNsIM, these events are not required to be detected on the same physical WSN, but they just need to share the same sensor group identifier at the DETECT level. Event (a) is not mandatory, as the detection probability is not 100%. Please not that each of the listed events taken singularly would not imply a security anomaly or be a reliable indicator of it.

The EDL description of the above scenario is provided in the following (in the assumption of unique event identifiers):

(((E1 SEQ ((E2 AND E3) OR (E4 AND E5)))
OR
((E2 AND E3) AND (E4 AND E5)))
SEQ E6) SEQ E7

Top-down and left to right, using 4 levels of alarm severity:

a) E1 can be associated a level 1 warning (alert to the security officer);

b) The composite events determined by the first group of 4 operators and the second group of 3 operators can be both associated a level 2 warning (triggering the unconditional emergency stop message);

c) The composite event terminating with E6 can be associated a level 3 warning (switch on back-up power supply, whenever available)

d) The composite event terminating with E7 (complete scenario) can be associated a level 4 warning (emergency call to first responders).

In the design phase, the scenario is represented using Event Trees and stored in the Scenario Repository of DETECT. In the operational phase, SeNsIM records the sequence of detected events in the Event History. When the events corresponding to the scenario occur, DETECT provides the scenario identifier and the alarm level (with a likelihood index in case of non deterministic detection models). Pre-configured countermeasures can then be activated by the SMS on the base of such information.

6. Conclusions and future works

Wireless sensors are being investigated in several applications. In this paper we have provided the description of a framework which can be employed to collect and analyze data measured by such heterogeneous sources in order to enhance the protection of critical infrastructures.

One of the research threads points at connecting by WSN traditionally wired sensors and application specific devices, which can serve as useful information sources for a superior situational awareness in security critical applications (like in the example scenario provided above). The verification of the overall system is also a delicate issue which can be addressed using the methodology described in [3].

We are currently developing the missing modules of the software system and testing the already available ones in a simulated environment. The next step will be the interfacing with a real SMS for the on-the-field experimentation.

References

1. A. Chianese, A.; A. Gaglione, A.; N. Mazzocca, N. & V. Moscato, V. 2008: “SeNsIM: a system for Sensor Networks Integration and Management”. To appear in Journal of System Architecture (JSA), Elsevier.
2. F. Flammini, A. Gaglione, N. Mazzocca &, C. Pragliola 2008. : “DETECT: a novel framework for the detection of attacks to critical infrastructures”. To appear in: Proc. European Safety & Reliability Conference (ESREL’08), 2008.
3. F. Flammini, F.; N. Mazzocca, N. & A. Orazzo, A. 2008: “Automatic instantiation of abstract tests to specific configurations for large critical control systems”. In: Journal of Software Testing, Verification & Reliability (Wiley), DOI: 10.1002/stvr.389
4. Garcia, M.L. 2001. The Design and Evaluation of Physical Protection Systems. Butterworth-Heinemann, USA.
5. Lewis, T.G. 2006. Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation. John Wiley, New York.
6. A. Perrig, A.; J. Stankovic, J. & D. Wagner, D. 2004. Security in wireless sensor networks. In Communications of the ACM, Vol. 47, No. 6, pp. 53-57
7. M. Rahimi, M.; R. Baer, R. & et al. 2005. Cyclops: In situ image sensing and interpretation in wireless sensor networks. In Proc. 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys'05), 2005.
8. Roman, R., Alcaraz, C. &, and Lopez, J. 2007. The role of Wireless Sensor Networks in the area of Critical Information Infrastructure Protection. Inf. Secur. Tech. Rep., Vol. 12, No. 1 (Jan. 2007), pp. 24-31.