5

Secure Platform Over Wireless Sensor Networks

Marco Pugliese, Luigi Pomante and Fortunato Santucci
Center of Excellence DEWS, University of L'Aquila
Italy

1. Introduction

Homeland security and monitoring of critical infrastructures, such as buildings, bridges, nuclear power plants, aircrafts, etc., represent challenging application domains for modern networking technologies. In this context Wireless Sensor Networks (WSNs) are gaining interest as a fundamental component of an advanced platform that embeds pervasive monitoring, networking and processing. Indeed, recent literature has addressed the perspectives of WSNs for monitoring structural and functional health of industrial plants, e.g. in (Akyildiz, et al., 2002; Bai et al., 2004; Barbaràn et al., 2007; Cho et al., 2008; Flammini et al., 2008; Kim et al., 2007): nevertheless, we can observe that the dominating paradigm is to exploit WSNs features in terms of a "network of small sensors", while almost unexplored is the more advanced paradigm of "networked smart sensors" and the underlying opportunity to actually support autonomous (anomaly) detection processes. A large body of specialized literature deals with this topic and several ad-hoc solutions can be found. On the contrary, we try to develop a different approach in this context: resorting to security mechanisms that are made available in traditional networks can provide a suitable and reliable framework, while smart adaptations are targeted to meet tight resource constraints and possible performance degradation.

Therefore we argue to demonstrate experimentally that, under certain limitations, a WSN can operate as a functionally "autonomous entity" not only for sensing operations. Despite the hard constraints on HW and the computation limitations, a WSN node is not just a sensing device (such as a magnetic contact or an infrared source): it is indeed a smart micro-device equipped with CPU and memory and is able to perform some autonomous data pre-processing, coding and transmission. Moreover the peculiar feature of a WSN with respect to a traditional sensor network is not to rely on fixed devices and cabling: nevertheless this comes at the cost of the availability of the so-called "ad-hoc" network properties (e.g. a sophisticated topology rearrangement mechanism is mandatory to achieve fault tolerance) as well as peer-to-peer frameworks, which imply enhanced protocol complexity and further computational and memory resource.

However, if proper design approaches (Pugliese et al., 2009; Sangiovanni-Vincentelli & Martin, 2001) are adopted, also the provision of fundamental security services (Hu et al., 2004; Law et al., 2005) can be pursued, which is a fundamental step towards the development of WSNs in critical applications; indeed the typical WSN deployment scenarios depicted above are highly exposed to physical capture or signal interception by external attackers much more than traditional sensors, which can be monitored by an extra-surveillance service.

Therefore providing security in a WSN system cannot be restricted to providing a robust cryptographic scheme, also because this kind of schemes are heavy demanding in terms of computational power and memory. Indeed a smart intrusion detection service should be provided also with ciphering and authentication in order to build up a "security service" package that will enhance the typical middleware services provided by an Application Execution Environment (AEE): this service package is the core feature of the proposed "secure platform" that is proposed, analyzed and tested in this chapter.

This chapter is organized as follows: Sec. 2 deals with the security services provided by the "Secure Platform", Sec. 3 and Sec. 4 describe fundamental algorithms and architectures supporting those security services, Sec. 5 reports the design approach of the platform while Sec. 6 is concerned withy a prototype of implementation and related tests. Sec. 7 deals with a viable conformance path to the trusted computing guidelines (TCG, n.d.).

2. Secure platform functions

Fig. 1 shows the main functional blocks of the proposed Secure Platform: apart from the block providing the typical middleware services (MW Services) and shared memory, other specific services (in this case security-oriented) are implemented as customizations of specific SW component and provided to the AEE via different APIs. It is very important to note that the "secure platform approach" offers a promising guideline to design and implement "integrated security" over WSN in a "application-oriented" approach which is aligned to the current SW development paradigms over resource constrained devices (Gay, 2003; Kliazovich, 2009; Sangiovanni-Vincentelli & Martin, 2001).

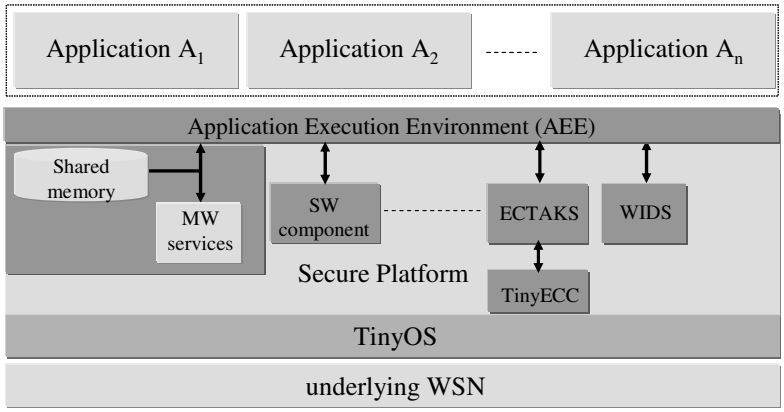


Fig. 1. Secure Platform Architecture

In this case at least two functional blocks are provided: the cryptography module, which implements ECTAKS (Elliptic Curve-based Topology Authenticated Key Scheme), and the intrusion detection module, which implements WIDS (Weak process model-based Intrusion Detection System): the former one represents a novel contribution that enhances the capabilities of the approach in (Pugliese & Santucci, 2008) by exploiting the advanced security features of elliptic curves, while the latter one integrates the developments proposed in (Pugliese et al., 2008, 2009).

TinyECC module (Liu, 2008) represents the ECC security package in WSN as it natively integrated with TinyOS (TinyOS, n.d.), the widely used operating system over WSN: ECTAKS, as we will show in next sections, rely on TinyECC security services to encrypt / decrypt messages.

Next sections, Sec. 3 and Sec. 4, deal with ECTAKS and WIDS modules respectively as well as with security and cost evaluations; however further details, especially about the mathematical proofs of theorems and computation expressions, can be found in (Pugliese et al., 2008, 2009; Pugliese & Santucci, 2008).

3. Elliptic curve-based topology authenticated key scheme (ECTAKS)

3.1 Motivations

In traditional networks such as the Internet, Public Key Cryptography (PKC) has been the enabling technology underlying many security services and protocols (e.g., SSL, IPsec). However, in WSNs PKC has not been widely adopted due to the resource constraints on sensor platforms, in particular the limited battery power and storage capacity. There has been intensive research aimed at developing techniques that can bypass PKC operations in sensor network applications. For example, there has been a substantial amount of research on random key pre-distribution for pair-wise key establishment, e.g. (Eschenauer & Gligor, 2002). However, these alternative approaches do not offer the same degree of security or functionality of PKC. For instance, none of the random key pre-distribution schemes can guarantee key establishment between any two nodes and tolerate arbitrary node compromises at the same time. Pair-wise key establishment can always be achieved, e.g. by resorting to the Diffie-Hellman key exchange protocol (Diffie & Hellman, 1976) without suffering from the node compromise problem and without requiring time synchronization.

Thus, it is desirable to explore the application of PKC on resource constrained sensor platforms (Malan, 2004; Menezes, 1996). Elliptic Curve Cryptography (ECC) has been the top choice among various PKC options due to its fast computation, small key size, and compact signatures: for example, to provide equivalent security to 1024-bit RSA, an ECC scheme only needs 160 bits on various parameters, such as 160-bit finite field operations and 160-bit key size (Gura et al., 2004). TinyECC, targeted at TinyOS, includes almost all known optimizations for ECC operations.

Taking into account the above considerations, we will show how the "hybrid" topology-based authentication logic (Topology Authenticated Key Scheme, TAKS) we proposed in (Pugliese & Santucci, 2008) can be enhanced using an ECC-based vector algebra (and, therefore, we now denote as ECTAKS) and be compatible with TinyECC.

3.2 EC Extensions to vector algebra over GF

Before starting with ECTAKS description, it is necessary to introduce some new algebraic tools and, specifically, the extension to elliptic curves of vector algebra over GF(q). Let $GF(q_E)$ be a finite field and let $x^3 + ax + b$, where $a,b \in GF(q_E)$, be a cubic polynomial with the condition that $4a^3 + 27b^2 \neq 0$ (this ensures that the polynomial has no multiple roots); an elliptic curve E over $GF(q_E)$ is the set of points (x,y) with $x,y \in GF(q_E)$ that satisfies the

condition $y^2 = x^3 + ax + b$ and also an element denoted O called the "point at infinity": the point at infinity is the point of intersection where the y-axis and the line at infinity (the collection of points on the projective plane for which z=0) meet. The elements over $E(GF(q_E))$, or the point in E, are denoted #E which results to be a function of q_E. An elliptic curve E can be made into an Abelian group by defining an additive operation on its points (Koblitz, 1987). As the elements of a group can be generated starting from a base element, or generator, by successive multiplications with scalars, we introduce a supplementary field GF(q) with $q \geq \#E$, (therefore q is function of q_E) and, as in TAKS, $q > N$ where N represents the total number of nodes in the network (Pugliese & Santucci, 2008). It is important to note that ECTAK results to be a point on E.

Let V be a vector space over GF(q) with the generic element $\underline{v} \in V$ represented through the 3-pla (v_x, v_y, v_z) with $v_x, v_y, v_z \in$ GF(q), let V_E be a vector space over E with the generic element in $V \in V_E$ represented through the 3-pla (V_1, V_2, V_3) with $V_1, V_2, V_3 \in$ E; let P, Q be points in E. We will denote elements in V as "scalar vectors" because their components are scalars in GF(q), and elements in V_E as "point vectors" because their components are points in E. ECC algebra introduces the "scalar by point product" (the operator symbol is usually omitted) which coincides with the addition of a point by itself many times the value of the scalar.

ECC vector algebra introduces two new operators: the "scalar vector by point product" (denoted by the symbol $\circ$) and the "scalar vector by point vector product" (denoted by the symbol $\otimes$). Identity elements are $0 \in GF(q)$, $\underline{0} = (0,0,0) \in V$ and $\underline{O} = (O,O,O) \in V_E$.

The operator "scalar vector by point product" is a function formally represented as $\circ : V \times E \to V_E$ and defined by

$$\underline{v} \circ P = (v_x, v_y, v_z) \circ P \equiv (v_x P, v_y P, v_z P) \tag{1}$$

It is straightforward to show that

$$\begin{aligned} &\underline{0} \circ P = (0,0,0) \circ P = (0P, 0P, 0P) = (O,O,O) \\ &\underline{v} \circ O = (v_x, v_y, v_z) \circ O = (v_x O, v_y O, v_z O) = (O,O,O) \end{aligned} \tag{2}$$

and the distributive of $\circ$ respect to + and vice-versa:

$$\begin{aligned} &(\underline{a} + \underline{b}) \circ P = \\ &= ((a_x + b_x)P, (a_y + b_y)P, (a_z + b_z)P) \\ &= ((a_x P + b_x P), (a_y P + b_y P), (a_z P + b_z P)) \\ &= \underline{a} \circ P + \underline{b} \circ P \end{aligned} \tag{3}$$

$$\begin{aligned} &\underline{v} \circ (P + Q) = \\ &= (v_x, v_y, v_z) \circ (P + Q) \\ &= (v_x (P + Q), v_y (P + Q), v_z (P + Q)) \\ &= ((v_x P + v_x Q), (v_y P + v_y Q), (v_z P + v_z Q)) \\ &= \underline{v} \circ P + \underline{v} \circ Q \end{aligned} \tag{4}$$

The operator "scalar vector by point vector product" is a function formally represented as $\otimes : V \times V_E \to E$ and defined by

$$\underline{v} \otimes \underline{V} = (v_x, v_y, v_z) \otimes (V_1, V_2, V_3) \equiv v_x V_1 + v_y V_2 + v_z V_3 \tag{5}$$

It is straightforward to show that

$$\begin{aligned} &\underline{0} \otimes \underline{V} = (0,0,0) \otimes (V_1, V_2, V_3) = 0V_1 + 0V_2 + 0V_3 = O \\ &\underline{v} \otimes \underline{O} = (v_x, v_y, v_z) \otimes (O, O, O) = v_x O + v_y O + v_z O = O \end{aligned} \tag{6}$$

and the distributive of $\otimes$ respect to + and vice-versa:

$$\begin{aligned} &(\underline{a} + \underline{b}) \otimes \underline{U} = (\underline{a} + \underline{b}) \otimes (U_1, U_2, U_3) \\ &= (a_x + b_x)U_1 + (a_y + b_y)U_2 + (a_z + b_z)U_3 = \underline{a} \otimes \underline{U} + \underline{b} \otimes \underline{U} \end{aligned} \tag{7}$$

$$\begin{aligned} &\underline{v} \otimes (\underline{V} + \underline{W}) = (v_x, v_y, v_z) \otimes (V_1 + W_1, V_2 + W_2, V_2 + W_2) \\ &= v_x(V_1 + W_1) + v_y(V_2 + W_2) + v_z(V_3 + W_3) = \underline{v} \otimes \underline{V} + \underline{v} \otimes \underline{W} \end{aligned} \tag{8}$$

The following identity $\underline{u} \otimes (\underline{v} \circ P) \equiv (\underline{u} \cdot \underline{v})P$ holds:

$$\begin{aligned} &\underline{u} \otimes (\underline{v} \circ P) = \\ &= \underline{u} \otimes (v_x P, v_y P, v_z P) = (u_x, u_y, u_z) \otimes (v_x P, v_y P, v_z P) \\ &= u_x v_x P + u_y v_y P + u_z v_z P = (\underline{u} \cdot \underline{v})P \end{aligned} \tag{9}$$

where the operator $\cdot$ denotes the usual scalar product between two vectors of scalars.

3.3 The scheme

Along what done for TAKS, ECTAKS is pair-wise, deterministic, shared keys are not pre-distributed but instead generated starting from partial key components. It exploits the impracticability in solving the Elliptic Curve Discrete Logarithm Problem (EDLP), the analogous of the discrete logarithm problem (DLP) applied to integers on GF(q) (Menezes et al., 1996).

Let V be a vector space over GF(q), V_E be a vector space over E, $f()$ be a function defined on GF(q) and $F()$ defined on E satisfying the following requirements:

- R1. Both $f()$ and $F()$ are one-way functions
- R2. $f(\underline{u}) * f(\underline{u}') = f(\underline{u}') * f(\underline{u}) \neq 0$ for $\forall \underline{u}, \underline{u}' \in V$ and for any commutative operator $*$
- R3. $F(\underline{u}, \underline{U}) = F(\underline{U}, \underline{u})$ for $\forall \underline{u} \in V$ and $\forall \underline{U} \in V_E$.

Let $G(.)$ a function defined on E satisfying the following requirements:

- R4. It must be a one-way function
- R5. $G(\underline{u}, \underline{U}) =$ O must hold only for $\underline{u} \in V' \subset V$ and $\underline{U} \in V'_E \subset V_E$, with V' and V'_E predefined sub-spaces of V and V_E respectively.

Definitions stated for TAKS in (Pugliese & Santucci, 2008) still hold true: each node stores the following information:

- Private Key Component (PRKC) which is a vector of scalars over GF(q)
- Public Key Component (PUKC) which is a vector of points over E
- Local Topology Vector (LTV) which is a vector of scalars over GF(q).

Information is classified according to the following definitions:

- *Public*: any information anyone can access (attackers included)
- *Restricted*: any information any node in the network can access
- *Private*: any information only a single node in the network can access
- *Secret*: any information only the planner can access.

According to Kerkhoff's principle, the explicit expressions for both $f()$ and $G(.)$ are public. Fig. 2 reports the conceptual representation of the proposed scheme.

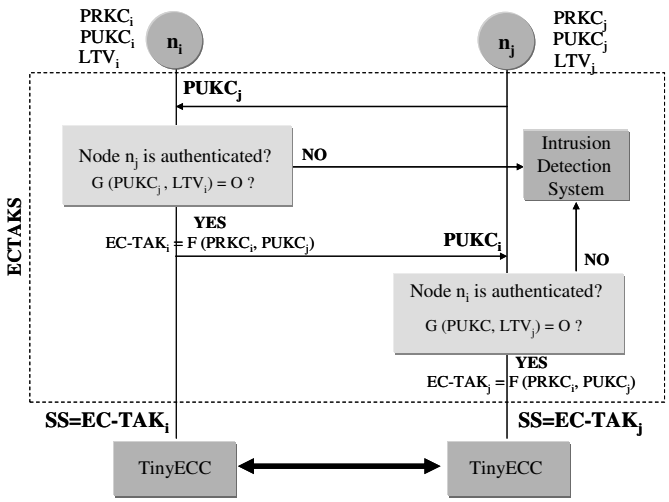


Fig. 2. Conceptual representation of the proposed cryptographic scheme

Node n_j broadcasts $PUKC_j$ and, among the others, node n_i receives it and starts the authentication procedure by executing the verification function $G()$ with inputs LTV_i and $PUKC_j$: if the result is the point at infinity O then node n_j has been successfully authenticated by node n_i and $ECTAK_i$ is generated. The same steps are performed by node n_j and, in case of successful authentication, $ECTAK_j$ is generated. If $f()$ and $F()$ are compliant to requirements R1, R2 and R3, then $ECTAK_i$ and $ECTAK_j$ coincide and ECTAK is a symmetric key shared between nodes n_i and n_j. Therefore ECTAK defines the Shared Secret (SS) which is a mandatory common information shared by parties to encrypt and decrypt messages in standard ECC schemes, such as ECDSA, ECDH, ECIES implemented in TinyECC.

Let n_i and n_j be a nodes pair. The following definitions are assumed:

a. Let $A \subseteq V$, $M \subseteq V$. Elements in A are defined as follows: $\forall \underline{a}_i, \underline{a}_j \in A$ if $\underline{m} \cdot (\underline{a}_i \times \underline{a}_j) \neq 0$ with $\underline{m} \in M$ an arbitrary predefined vector over GF(q): this information is *secret*
b. Let $b \in B \subseteq GF(q)$ be an arbitrary predefined scalar in B but not generator of GF(q): this information is *secret*
c. Let $\underline{c} \in C \subseteq V$ be an arbitrary predefined vector over GF(q): this information is *secret*
d. Let $f(\,) = kb^{\underline{m} \cdot (.)}$ where $\underline{m} \in M$ satisfies (a) and $k \in GF(q)$. This definition for $f(\,)$ is compliant to specified requirements R1, R2 and R3 because for $\forall \underline{v}, \underline{v}' \in V$ and $\forall k \in GF(q)$ is $kb^{\underline{m} \cdot \underline{v}} * kb^{\underline{m} \cdot \underline{v}'} = kb^{\underline{m} \cdot \underline{v}'} * kb^{\underline{m} \cdot \underline{v}} = |k|^2 b^{\underline{m} \cdot (\underline{v} + \underline{v}')} = |k|^2 b^{\underline{m} \cdot (\underline{v}' + \underline{v})}$, where $*$ is the mod q product (commutative) operator. Hereinafter the symbol $*$ will be omitted
e. Let $\underline{k}_{li}$, $\underline{k}_{lj} \in KL \subseteq V$ (this information is *private*)
f. Let $\underline{K}_{ti}$, $\underline{K}_{tj} \in KT \subseteq V_E$ (this information is *public*)
g. Let $LTV_i \in V$. Elements LTV_i are defined to be co-planar to $\underline{m}$ and $\underline{a}_j$ if n_j is an admissible neighbor of node n_i , or is "topology authenticated" (this information is *private*)
h. Let $\alpha, \beta \in GF(q)$ be a random scalars in GF(q) generated by n_i and n_j respectively (this information is *secret*)
i. Let $E : y^2 = x^3 + ax + b$ and $P \in E$ be respectively an elliptic curve E and a point in E both compliant to security requirements in (Certicom Research Standards, n.d.) (this information is *public*).

Setting $k \equiv b^{\underline{m} \cdot \underline{c}}$ in the definition of $f(\,)$:

$$\begin{cases} \underline{k}_{li} \equiv \alpha \underline{a}_i f(\underline{a}_i) = \alpha \underline{a}_i b^{\underline{m} \cdot (\underline{a}_i + \underline{c})} \\ \underline{K}_{ti} \equiv \alpha \underline{k}_{ti} \circ P = \alpha (\underline{s}_i \times \underline{a}_i) \circ P \end{cases} \tag{10}$$

$$\begin{cases} \underline{k}_{l_j} \equiv \beta \underline{a}_j f(\underline{a}_j) = \beta \underline{a}_j b^{\underline{m} \cdot (\underline{a}_j + \underline{c})} \\ \underline{K}_{t_j} \equiv \beta \underline{k}_{t_j} \circ P = \beta (\underline{s}_j \times \underline{a}_j) \circ P \end{cases} \tag{11}$$

where setting now $k \equiv 1$ in the definition of $f(\,)$:

$$\begin{cases} \underline{s}_i = \underline{m} f(\underline{a}_i) = \underline{m} b^{\underline{m} \cdot \underline{a}_i} \\ \underline{s}_j = \underline{m} f(\underline{a}_j) = \underline{m} b^{\underline{m} \cdot \underline{a}_j} \end{cases} \tag{12}$$

According to Kerkhoff's principle, the explicit expressions for $\underline{k}_l$ and $\underline{K}_t$ are *public*.

Given **m**, **c**, b and for $\forall \underline{a}_i, \underline{a}_j \in A$, the following properties hold true:

1. Always $ECTAK \neq \mathrm{O}$. This follows from the condition $\underline{m} \cdot (\underline{a}_i \times \underline{a}_j) \neq 0$ assumed in (a) with $\forall P \neq O$
2. Elements in KL are always distinct, i.e. for $\forall \underline{k}_{l_i}, \underline{k}_{l_j} \in KL$ is $\underline{k}_{l_i} \times \underline{k}_{l_j} \neq 0$ which can be derived from $\underline{m} \cdot (\underline{a}_i \times \underline{a}_j) \neq 0$ assumed in (a)
3. Elements in KT are always distinct, i.e. for $\forall \underline{K}_{t_i}, \underline{K}_{t_j} \in KT$ is $\underline{k}_{ti} \times \underline{k}_{tj} \neq 0$ with $\forall P \neq O$ which can be derived from $\underline{k}_{li} \times \underline{k}_{l_j} \neq 0$ and $\underline{k}_{ti} \,//\, \underline{m} \times \underline{k}_{li}$ and $\underline{k}_{tj} \,//\, \underline{m} \times \underline{k}_{lj}$ (compare (10), (11) and (12))
4. In each node is $\underline{k}_l \otimes \underline{K}_t \equiv \mathrm{O}$ that is $\underline{k}_l \cdot \underline{k}_t \equiv 0$ with $\forall P \neq O$ which can de derived from the vector identity $\underline{s} \cdot (\underline{a} \times \underline{a}) \equiv 0$ for $\forall \underline{s}$.

Theorem (ECTAK Generation). In a node pair n_i and n_j, given $\underline{m} \in M$ and $\underline{a}_i, \underline{a}_j \in A$ as defined in (a), $b \in B$ as defined in (b), $\underline{c} \in C$ as defined in (c), $\underline{k}_{l_i}$, $\underline{k}_{l_j}$ as defined in (e), $\underline{K}_{t_i}$ and $\underline{K}_{t_j}$ as defined in (f), α, β as defined in (h), and if $ECTAK_i$ and $ECTAK_j$ are defined as:

$$ECTAK_i \equiv \left| \underline{k}_{l_i} \otimes \underline{K}_{t_j} \right| \tag{13}$$

and

$$ECTAK_j \equiv \left| \underline{k}_{l_j} \otimes \underline{K}_{t_i} \right| \tag{14}$$

then ECTAK is a symmetric key defined as follows:

$$ECTAK = ECTAK_i = ECTAK_j = \alpha\beta b^{\underline{m} \cdot (\underline{a}_i + \underline{a}_j)} \, k \left| \underline{m} \cdot (\underline{a}_i \times \underline{a}_j) \right| P \tag{15}$$

Proof. The proof is straightforward: putting (10) into (13), exploiting the vector algebra property $\underline{a} \cdot (\underline{s}' \times \underline{a}') \equiv \underline{s}' \cdot (\underline{a}' \times \underline{a})$ and the property (9) then

$$\begin{aligned} & ECTAK_i \equiv \underline{k}_{l_i} \otimes \underline{K}_{t_j} \\ & = \underline{k}_{l_i} \otimes (\underline{k}_{t_j} \circ P) = (\underline{k}_{l_i} \cdot \underline{k}_{t_j}) P \\ & = \alpha \underline{a}_i k b^{\underline{m} \cdot \underline{a}_i} \cdot (\underline{s}_j \times \beta \underline{a}_j) \, P = \alpha\beta b^{\underline{m} \cdot \underline{a}_i} k \underline{s}_j \cdot (\underline{a}_j \times \underline{a}_i) \, P \end{aligned} \tag{16}$$

Putting (11) into (14), exploiting the property $\underline{a}_j \cdot (\underline{s}_i \times \underline{a}_i) \equiv \underline{s}_i \cdot (\underline{a}_i \times \underline{a}_j)$ and the property (9) then

$$\begin{aligned} & ECTAK_j \equiv \underline{k}_{l_j} \otimes \underline{K}_{t_i} \\ & = \underline{k}_{l_j} \otimes (\underline{k}_{t_i} \circ P) = (\underline{k}_{l_j} \cdot \underline{k}_{t_i}) P \\ & = \beta \underline{a}_j k b^{\underline{m} \cdot \underline{a}_j} \cdot (\underline{s}_i \times \alpha \underline{a}_i) \, P = \beta\alpha b^{\underline{m} \cdot \underline{a}_j} k \underline{s}_i \cdot (\underline{a}_i \times \underline{a}_j) \, P \end{aligned} \tag{17}$$

Putting (12) into (16) and (17), the expression (15) is obtained in both cases and the proof is completed. Q.E.D.

Theorem (Node Topology Authentication). In a node pair n_i and n_j, if $LTV_i \otimes \underline{K}_{tj} = \mathrm{O}$ then node n_j is an admissible neighbor of node n_i or, node n_j is authenticated by n_i.

Proof. By definition (g) if node n_j is an admissible neighbor of node n_i (or "topology authenticated" by n_i) then LTV_i must be co-planar to $\underline{m}$ and $\underline{a}_j$, hence $LTV_i \cdot (\underline{m} \times \underline{a}_j) \equiv 0$ and therefore $LTV_i \cdot \underline{k}_{tj} = 0$; by multiplying both terms by $P \neq O$, it turns out $LTV_i \otimes \underline{K}_{tj} = O$. It is straightforward to show that function $G(\underline{u}, \underline{U}) \equiv \underline{u} \otimes \underline{U}$ is compliant to requirements R4 and R5. QED.

Node authentication by topology information introduces an important security improvement in the family of TinyECC cryptographic schemes because only the integrity check (by means of the Key Derivation Function) of the received crypto-text is actually implemented there.

3.4 Security and cost analysis

We will show how ECTAKS can enhance the security level provided by TAKS: the relevant questions and related answers are as follows:

1. Which is the entropy per binit associated to ECTAK? ECTAK entropy per binit is $\cong 1$ which is the same result for TAKS (Pugliese & Santucci, 2008) as uncertainty about $\underline{K}_t = \underline{k}_t \circ P$ is the same as it is about $\underline{k}_t$ being P a known point.
2. How much complex is the inverse problem to break ECTAKS (security level in a single node)? For the EDLP over E to be intractable, it is important to select an appropriate E (it must be a non-supersingular curve) and q_E such that #E is divisible by a large prime or such that q_E is itself a large prime. Most significantly, no index-calculus-type algorithms are known for EDLP as for the DLP (Menezes et al., 1996). For this reason, the EDLP is believed to be much harder than DLP in that no subexponential-time general-purpose algorithm is known.

The cost is measured in terms of computational time. We assume to employ 128 bit ECTAK keys (i.e. $q = 2^{128}$): it can be shown that (15) can be computed through ~60000 16-bit operations (additions and products). If MicaZ motes are employed (8-bit processor MPR2400 @ 7.4 MHz), and assuming 10 clock cycles / operation, the cost in terms of computation time for the calculation of a 128-bit ECTAK is estimated to be about ~80 ms.

4. Weak process-based intrusion detection system (WIDS)

4.1 Motivations

The further security service component in our Secure Platform is the intrusion detection logic (IDS). Its main function is to identify abnormal network activity that differs from the expected behavior (Kaplantzis, 2004; Karlof & Wagner, 2003; Roosta et al., 2006; Sharma et al., 2010). We will show how a light state-based anomaly-based detection logic can be suited to be implemented over WSN (Ioannis et al., 2007; Jangra et al., 2011; Kalita & Kar, 2009).

Smart nodes are typically provided with mechanisms to identify changes in system parameters or anomalous exchange of information: such data can be used as relevant observations to predict the hidden state of the system and infer whether it is under attack. An Hidden Markov Model (HMM), see e.g. (Ephraim & Merhav, 2002), is a doubly stochastic finite state machine with an underlying stochastic process that represents the real state of the system: the real state of the system is hidden but indirectly observable through another stochastic process that produces a sequence of observable events. The relationships between hidden states and observable data are stochastic as well as the transitions between states. HMMs (Doumit & Agrawal, 2003; Rabiner & Juang, 1986) have been widely used in network-based IDS for wired systems (Al-Subaie & Zulkernine, 2006; Khanna & Liu, 2006; Luk et al., 2007; Sheng & Cybenko, 2005; Yin et al., 2003) as well as for modeling Internet traffic (Dainotti et al., 2008). The Baum-Welch algorithm as likelihood criterion and technique for parameter estimation in HMM is extensively used in (Doumit & Agrawal, 2003) but some training data should be available and still result expensive in terms of computational and memory costs. Some conventional intrusion detection systems perform cross-correlation and aggregation of data, e.g. by analyzing fluctuation in sensor readings (Loo, 2005) or by detecting abnormal traffic patterns (Law, 2005). In general, the application of traditional IDSs to sensor networks is challenging as they require intense computation capability or too limited to a restricted number of threats. The implementation of an effective IDS over a WSN leads to the problem of finding a trade-off between the capability of identifying threats (i.e. with a bounded false alarm rate), the complexity of the algorithms and memory usage (Baker & Prasanna, 2005; Bhatnagar et al., 2010; Jiang, 2005; Kumari et al., 2010).

Our contribution proposes a novel network-layer anomaly detection logic over WSN exploits the Weak Process Models (WPM) and is here simply denoted as WIDS (WPM-based Intrusion Detection System): WPM are a non-parametric version of HMM, wherein state transition probabilities are reduced to rules of reachability in a graph representing the abnormal behaviors (Jiang, 2005). The estimation of a threat in the case of weak processes is greatly simplified and less demanding for resources. The *most probable* state sequence generated by the Viterbi algorithm (Forney, 1973) for HMM becomes the *possible* state sequence generated by simplified estimation algorithms for WPM. The intensity of the attack is evaluated by introducing a threat score, a likelihood criterion based on weighting states and transitions (Pugliese et al., 2008).

4.2 The scheme

As stated before, if WPM are used to model behavior, the algorithm to estimate the possible state sequences (instead of the most probable ones) is much easier than Viterbi estimator (Forney, 1973). But this comes at a cost: due to the cut of lower probabilities (approximated to zero) the expressiveness in WPM could be reduced with respect to HMM and false negatives can increase. However, it has been shown that adding a certain number of further states to WPM, expressiveness could be recovered (Pugliese et al., 2008). Indeed a sort of "state explosion" can require added memory for storage but the binary matrices describing WPM are very low dense (sparse matrix) and some algebraic tricks can be adopted. Given the choice of WPM as behavior model, the question becomes: which behavior should be modeled? Our solution is based on two basic ideas: first, the adoption of an anomaly-based

IDS and, second, a "hierarchical" model for abnormal behaviors. However, even anomaly-based detection algorithms are of lower complexity than misuse-based ones, the problem to model a behaviour still remains (Debar et al., 1999): usually the question is approached by defining different regions in the observation space associated to different system behaviors. Further we apply a "state classification", i.e. we associate each defined region to a specific sub-set (class) of WPM states (not single states) according to WPM topology. State classification can reduce false negatives and false positives in anomaly detection because different state traces (therefore different behavior patterns) contain the same information leading to a useful redundancy. In (Pugliese et al., 2008, 2009) we introduced two classes: LPA (Low Potential Attack) and HPA (High Potential Attack).

Definition 1. Low Potential Attack, LPA. An attack is defined in a "low potentially dangerous" state (or in a LPA state) if the threat is estimated to be in state x_j which is at least 2 hops to the final state.

Definition 2. High Potential Attack, HPA. An attack is defined in a "high potentially dangerous" state (or in a HPA state) if the threat is estimated to be in state x_j which is 1 hop to the final state.

WIDS identifies any observable event correlated to a threat by applying a set of anomaly rules to the incoming traffic. An example of anomaly can be the event of a node receiving multiple "setup" messages in a short time, or two "topologically far" nodes (i.e. nodes whose path length is >>1 hop) to receive similar message sequences. We will show how attacks can be classified into low and high potential attacks according to specific states in the corresponding WPM-based threat model. Alarms are issued as soon as one or more high potential attacks are detected. Considered threats are "hello flooding" and the generalized version of "sinkhole" and "wormhole": we will show that any possible attack against WSN network layer protocols can be derived from these models. The security performance analysis will be carried out by computing the probability of false positives and negatives. However, WPMs technique introduces the following drawback: as very low state transition probabilities are reduced (approximated) to zero, it results an increase of false negatives as some (hazardous) sequences could be classified as not possible when instead in a probabilistic model would be achievable. The number of false negatives decreases if we add states (Pugliese et al., 2008) but the drawback is a larger memory requirement. As it will be shown in the dedicated sections, Boolean matrices that describe the models are sparse and can be compacted for faster computation. The intensity of the attack is evaluated by introducing a threat score, a likelihood criterion based on weighting states and transitions. Intrusions and violations are classified into low potential attacks (LPA) and high potential attacks (HPA) depending on their distance from the state corresponding to a successful attack. When at least one HPA occurs, an alarm is issued. Moreover we introduce a score mechanism to weight state sequences where LPA and HPA contribute differently so that it becomes possible to derive how many LPA and / or HPA states have been experimented.

Definition 3. Threat Score s at Observation Step k [s^k]. It is a weighting mechanism we apply to states and transitions in a given WPM. Weights are represented by a square $n \times n$ matrix (we denote "Score Matrix" S) whose elements are defined as follows (n is the number of states in the WPM): s_{ij} is the score assigned to the transition from x_j to x_i and s_{jj} is the

score assigned to the state x_j. In (Pugliese et al., 2008) it has been shown that $s^k = Hn^k_{hpa} + Ln^k_{lpa}$ where n^k_{hpa} and n^k_{lpa} are the number of HPA and LPA states that the system is supposed to have reached up to observation step k, and L, H are values to be assigned depending on LPA and HPA state topology in WPM graph (Pugliese et al., 2008) respectively. Last we introduce the concept of Anomaly Rule, the logic filter applied to incoming signaling messages, which gives two possible results: "no anomalies", resulting in the message being processed further, or "anomaly detected" resulting in a "threat observable".

The main objective of IDS is to detect attacks coming from insider intruders, i.e. only combinations of "hello flooding", "sinkhole" and "wormhole" threats (Debar et al., 1999; Roosta et al., 2006; Singh et al. 2010; Whitman & Mattord, 2011). These attacks are based on well-formed messages generated by authenticated nodes where control information is chosen to generate malicious network topologies. IDS monitoring domain is restricted to observables associated to any combination of these threats (we denote the Aggregated Threat Model).

In summary the process can be stated through the following steps: 1) Analyze the behaviour of the threat; 2) Derive the Anomaly Rules; 3) Derive the WPM-based threat model and 4) Assign weights to WPM states and transitions. WPM-based models for single threats are shown in (Pugliese et al., 2008). Following these steps, we obtain the WPM aggregated model in Fig. 3: ovals represent states, grey border indicate the final states (X_9 and X_10), numbers into brackets the associated threat observables. LPA states are X_1, X_2, X_5 and X_6; HPA states are X_3, X_4, X_7 and X_8. A positive effect of aggregation is to enhance model effectiveness: this is due to the possible sharing of "threat observables" among different threats (as it is for "sinkhole" and "wormhole") and scores can be different. The observable $o^k = o_9$ is produced (defining a RESET state) when no threat observables are produced after K consecutive observation steps, with K a tunable threshold.

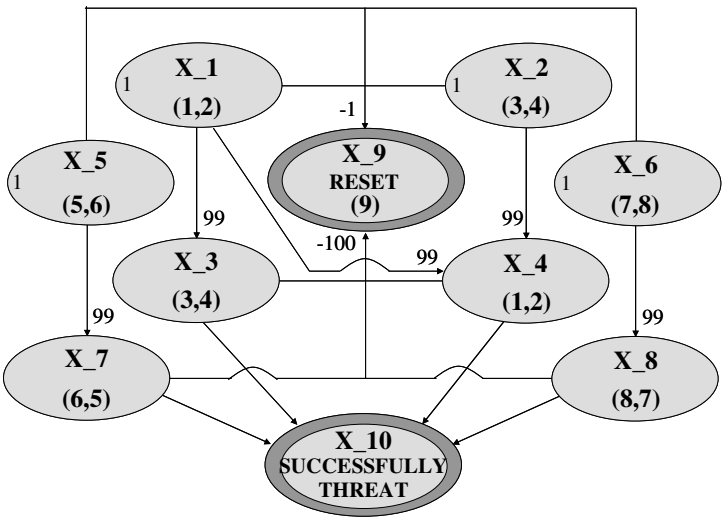


Fig. 3. The Aggregated Threat Model

4.3 Security and cost analysis

The security analysis will verify the effectiveness of the proposed IDS in terms of probabilities of false negatives and false positives. We introduce the false-negative rate (P_{neg}) which represents the rate that the detection algorithm is not able to identify an existing threat (*alarm mis-detections*), and the false-positive rate (P_{pos}) which represents the rate that the detection algorithm identifies undue threats (*false alarm detections*).

Definition 3. WPM Memory Length, WML. It is the length of the state sequence trace considered for alarms detection.

Test for alarm mis-detections (False Negatives). We will compute the false negatives probability P_{neg} by applying "ad-hoc" observables sequences to WPM model in Fig. 3 originated by "hello flooding", "sinkhole" and "wormhole" randomly aggregated. If WPM representation is structurally well-defined we should experiment always $s^k \neq 0$ (and therefore $P_{neg} \to 0$) for increasing observation steps ($k \to \infty$), for any combinations of threat behaviors. Here we report the case study with $k = 32$ observables, $WML = 10$ and $K = 3$: suppose an "hello flooding" attack has been engaged against the WSN: in this case the Anomaly Rules would produce an observable sequence of the type

$$\{5;5;*;8;7;*;*;6;6;8;8;*;*;*;8;*;\ *;5;7;*;*;7;*;6;8;*;*;*;5;5;*;*\} \tag{18}$$

and, in case of an attacking "sinkhole" / "wormhole", observable sequences like:

$$\{2;1;*;*;*;1;*;1;2;2;*;*;*;1;2;*;\ *;1;2;*;*;*;2;2;*;*;1;*;1;*;*;*\} \tag{19}$$

$$\{2;4;*;*;*;3;*;3;4;2;3;*;*;1;3;*;\ *;4;4;*;*;*;2;2;*;*;4;*;3;*;*;*\} \tag{20}$$

The symbol * means "no observable related to this threat". According to the previous considerations, we preliminarily note that:

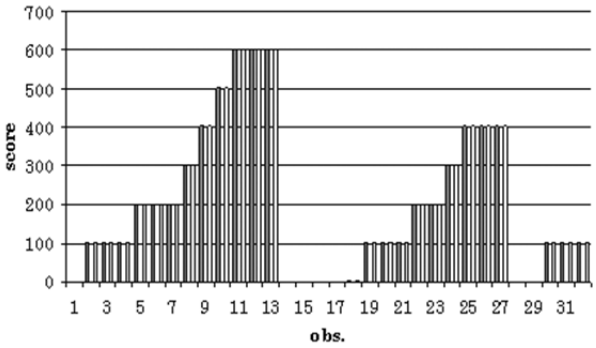
- There are no observable sharing between "hello flooding" and "sinkhole" or "wormhole";
- Observables for "sinkhole" are also observables for "wormhole" but not vice-versa.

Simulations results are graphically reported in Fig. 4 where dark grey bars refer to scores produced by individual threat models and light grey bars refer to aggregated threat models. As expected the same outputs from both models are obtained only for threats not sharing any observable (Fig. 4 a) while different outputs are obtained for threats, as "sinkhole" and "wormhole", sharing at least one observable (Fig. 4 b).

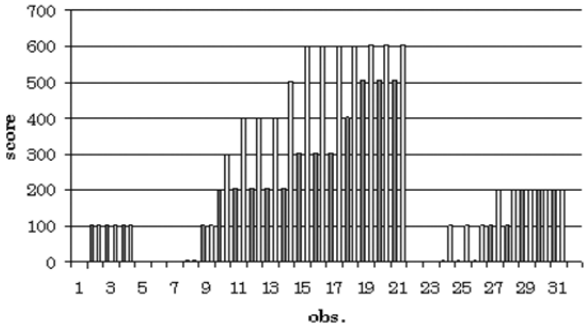
Test for false alarm detections (False Positives). Not well-defined Anomaly Rules can produce "undecided" threat observables which lead to potential false positives, hence $P_{pos} \neq 0$. False positives are structurally zeroed if no "undecided" threat observables are associated to HPA states. Two approaches can be adopted:

1. Insert further states associated to truly "threat observables into WPM paths where states associated to "undecided" threat observables are leaves: this approach can

decrease the probability for false positives ($P_{pos} \to 0$) because the joint probability to reach the leaf state can be very low in long paths; however a drawback is that long paths could reduce the system reactivity to threats.



(a)



(b)

Fig. 4. Scores from single (dark grey) and aggregated model (light grey) when (a) the sequence (18) is applied and (b) the sequences (19), (20) are applied

2. Introduce a further class of states associated to "undecided" threat observables: this approach cannot decrease the probability for false positives, but "ad-hoc" lighter countermeasures can be applied to nodes where alarms from "undecided" observables are generated (e.g. node quarantine rather than link release).

The cost is measured in terms of computational time. If n are the states in the Aggregated Threat Model, we can derive that the upper bound complexity in the computation of scores and alarms is $\approx 6WML \cdot n^2$ if $WML >> n$. If MICA2 motes (CROSSBOW, n.d.) are employed (8-bit processor ATMega128L @ 7.4 MHz), and assuming 20 clock cycles per arithmetic / logic operation, the average computation time per 32-bit operation is ~ 3 μs. If IMOTE motes (MEMSIC, n.d.) are employed (32-bit processor PXA271Xscale@{312, 416} MHz), and assuming 5 clock cycles per arithmetic / logic operation, the average computation time per 32-bit operation is ~ 0.03 μs (assuming 300 MHz for the clock). Suppose the case $n = 10$ and $WML = 100$. For MICA2 the estimated computation time is ≈ 200 ms, for IMOTE ≈ 2 ms.

5. Secure platform design

The adopted architectural design (Roman et al., 2006) will be cross-layered (Kliazovich et al., 2009) and platform-based (Sangiovanni-Vincentelli & Martin, 2001). Cross-layer (CL) results in the interplay between network layer (topology management and routing protocol) and presentation layer (mobile agent based execution environment for distributed monitoring applications): applied to security, an important benefit of CL mechanism is the exploitation of the interplay between different security measures in different layers to provide an enforced security service to applications. Platform-based design (PBD) results in the availability of a software platform where the internal structure is composed by "interconnected" SW components, which represent abstractions of the wired hardware components. Achievements of research goals are sought by taking care of the following major topics: selection of the right layers in the architectural design (a middleware layer is an essential component), application of the platform-oriented concepts for service mappings between layers, enhancement of the middleware layer with security services offered by lower layers entities and, on top, the creation of a flexible AEE by means of agents.

Fig. 5 depicts WIDS functional blocks: the Threat Model (TM) block implements the WPM-based model for abnormal system behavior and the Anomaly Detection Logic (ADL) block implements detection and alarm generation functions. The Intrusion Reaction Logic (IRL) schedules the intervention priority toward the compromised nodes according to specific criteria (defense strategy); IRLA applies the countermeasures against attacks to compromised nodes, including node isolations (quarantine), key revocations, link release or inclusions in black lists / grey lists (Roman et al., 2006).

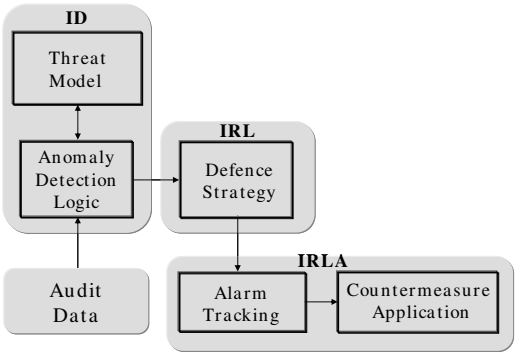


Fig. 5. WIDS functional blocks

6. Mobile agent-based middleware

A key characteristic of mobile agent-based middleware is that any host in the network is allowed a high degree of flexibility to possess any mixture of code, resources, and processors. Its processing capabilities can be combined with local resources. Code (in the form of mobile agents) is not tied to a single host but it is available throughout the network . Moreover, the mobile agent paradigm supports data-centric applications because the implementation code can migrate towards data no matter about node addressing (Hadim & Nader, 2006). Therefore in a mobile-agent application execution environment (Szumel et al.,

2005), each agent implements a sub-set of application components which can be proactively aggregated through agent mobility (code mobility across the network). Among the agent-based middleware solutions available from literature, we will refer to AGILLA (Fok et al., 2006), developed at the Washington University in St. Louis. There are different motivations for this choice. Some of these are listed in the following:

- it is developed using NesC (Gay et al., 2003) which is a component-based programming language (used to develop TinyOS): this occurrence simplifies the integration of further components in AGILLA code
- it is lighter than other mobile agent middleware solutions, e.g. Maté (Levis & Culler, 2002)
- agent mobility is selective, i.e. no code broadcast, e.g. Impala (Liu & Martonosi, 2003)
- agents hosted on adjacent nodes can share memory (through the "Tuple Space")

AGILLA middleware provides two components that facilitate inter-agent coordination: a *Tuple Space* and a *Neighbors List*, both maintained on each node by the middleware services. A Tuple Space is shared by local agents and is remotely accessible and offers a decoupled style of communication where one agent can insert a tuple, another can later read or remove it using pattern matching via a template. The Neighbors List is on every node and contains the location of all one-hop neighbors. Local agents can access it by executing special instructions. The agent architecture is described in (Fok et al., 2006). Code migration is implemented by moving or cloning an agent from one node to another. Migration can be strong or weak dependently if the current execution state is ported on the other node or not. When an agent moves, it carries its code and, if strong move, also state and resumes executing on the new node. When it clones, it copies its code and, if strong clone, state to another node and resumes executing on both the old and new nodes. Multi-hop migration is handled by the middleware and is transparent to the user. It is important to remember that AGILLA can initially deploy a network without any application installed: agents that implement the application can later be injected, actually reprogramming the network.

From the function decomposition shown in Fig. 5, the mapping between WIDS functions and SW components and mobile agents is shown in Fig. 6: ADL and TM blocks are mapped into SW components while IRL and IRLA blocks into a mobile agent, which is denoted by Intrusion Reaction Agent (IRA). SW components are indicated with smoothed squares. This design allows the optimal allocation and code distribution for those functions that should not be implemented anywhere.

6.1 Enhancements to AGILLA middleware

Current version of AGILLA foresees that only the AGILLA Manager can read and write into the Neighbor List and only the AGILLA Manager and Mobile Agents can read and write into Tuple Space.

As stated before, Neighbors List contains the location of all one-hop neighbors but topology authentication provided in ECTAKS should update this list with admissible neighbors only: therefore it would be preferred if ECTAKS could read and write into the Neighbor List as the AGILLA Manager does.

Moreover, WIDS should read and write into Tuple Space in order to manage IRA agents mobility according to the functional mapping shown in Fig. 6.

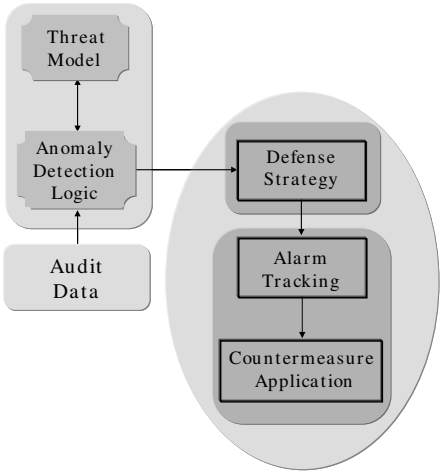


Fig. 6. Mobile Agent-based WIDS architecture

These enhancements have been designed as NesC stubs (Gay et al., 2003) embedded into AGILLA code (Pugliese et al., 2009). Fig. 7 schematically represents this added interfaces as bold arrows.

The first issue that has been actually addressed is related to the interface with the Communication Unit. In particular, the first enhancement made to AGILLA has been to add some basic mechanisms to let the agents able to retrieve some information about the radio traffic from the nodes. More in detail:

- the node-resident part of the middleware has been modified in order to allow the evaluation of some indicators, customizable by the designer, based on the analysis of the radio traffic
- the interface of the middleware towards the agents has been modified to allow an agent to retrieve the value of such indicators by pushing them on its stack.

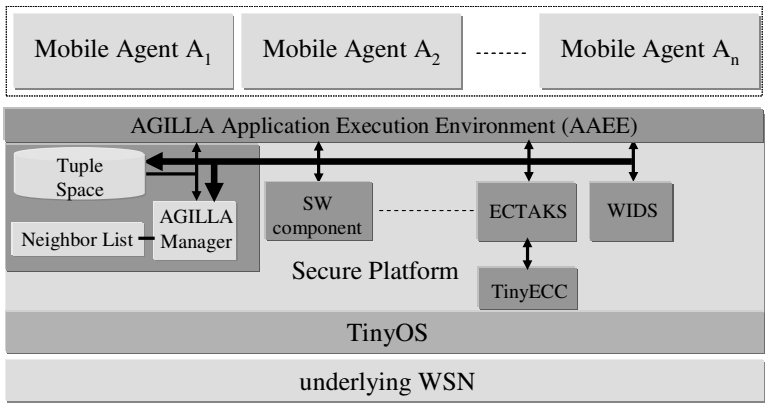


Fig. 7. Enhanced AGILLA Mobile Agent-based Secure Platform Architecture

In this way, the agents are able to check for anomalous values (i.e. alarms), as described in the previous sections. Moreover, this possibility has been added while keeping the existing interaction mechanisms between agents and nodes: the agent sees the added indicators as *virtual sensors* (Fig. 8) accessible as if they were normal sensors (i.e. light, temperature, etc…) by means of the *sense* instruction.

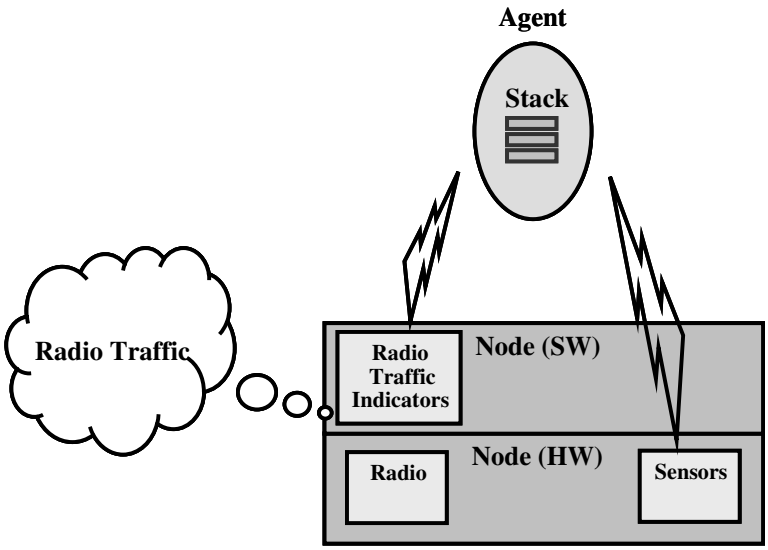


Fig. 8. Virtual Sensor Architecture

As a very simple example, if each node evaluates the number of received packets, an agent could retrieve such an information as shown in Fig. 8. It is worth noting that the approach is exactly the same as the one used to read the temperature sensor. In the sample code the agent turns on the red led when the number of received packets is larger than 10.

In order to make such a first extension to the AGILLA framework, a deep study (that will be very useful for future work) of the original architecture has been performed. First of all, it has been needed to understand the mapping mechanisms between AGILLA instructions and nesC components: each instruction is implemented by a component, stored in the *opcodes* directory, that offers the *BytecodeI* interface that includes the *execute* command. Such a command is called to execute an instruction that is identified by the codes stored in *AgillaOpcodes.h* used as parameters for the interface.

6.2 Validation

In order to validate the first AGILLA extensions and to give the flavor of its exploitation in building up the IDS proposed in the previous sections, the following example has been selected and concerns a sample agent-based application. More in details, as discussed before, by means of the middleware the agents can access to some information about the radio traffic (i.e. in this case just the number of the packets received by a node) as if they were sensor readings and can react if required.

The demo application (Fig. 9) is then based on a sample WSN composed of 4 *MicaZ* nodes and a *MIB510* board (connected to a PC) where 3 Agilla agents are injected for monitoring purposes. Such agents exploit the proposed middleware extensions and the Agilla reaction mechanism while moving on the WSN. The final goal is to detect the nodes that present one or more radio traffic indicators out of standard value (i.e. in this case the agents checks for a number of received packets larger than a defined threshold). The agents developed for such a purpose, called *TupleOut*, *Dynamic* and *Alarm*, are described in the following.

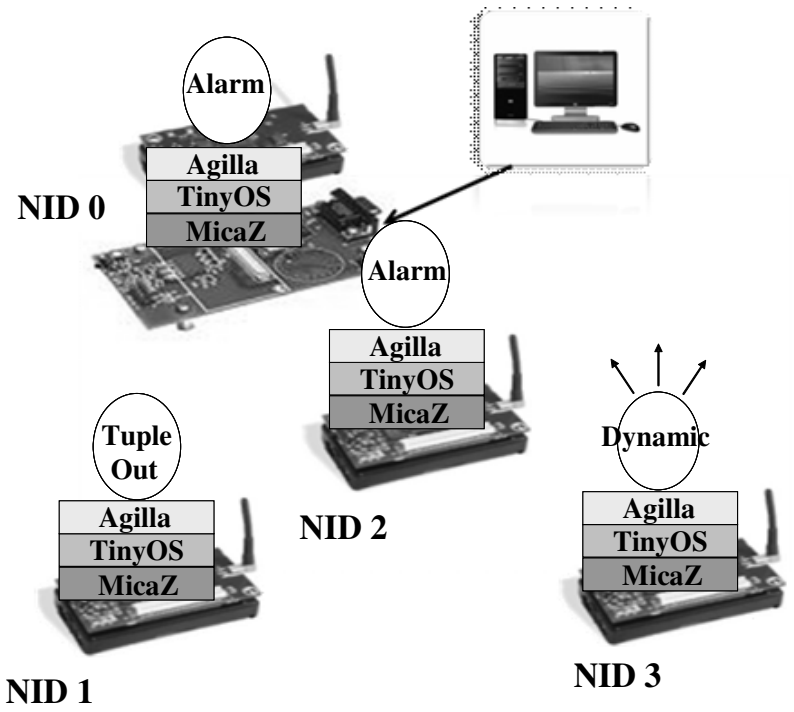


Fig. 9. Demo Application Architecture

TupleOut is a static agent, virtually present on each node to be monitored, that access to the radio traffic indicators evaluated by the node-side middleware, checks for anomalous values and insert a proper tuple on the tuple space of the node to signal an eventual alarm. In the proposed example the agents injected on node #1 checks for a number of received packets larger than 5 and, when the condition met, it inserts the alarm tuple on the Tuple Space of node #1 (Fig. 10). *Dynamic* is a dynamic (strong move in the whole WSN nodes) agent that looks for alarm tuples in the nodes tuple spaces. It exploits a template-based match by type reaction (Fig. 11) to detect an alarm tuple and then to eventually send to the Alarm agent the alarmed node ID. Finally, the *Alarm* agent is a static one that resides in the base station of the WSN (node #0). It receives alarm signals and alarmed node IDs and manages them. In the example, it simply displays by means of the leds the alarmed node ID and sends also such an information to the connected PC.

```
        // INIT
        pushc 0
        setvar 0
        // MANAGEMENT
BEGIN   pushc 25
        putled        // Red led on
        getvar 0
        copy
        inc
        setvar 0
        // CHECK
        pushc num_packets  // ID of the virtual sensor
        sense              // Read the virtual sensor
        pushcl 5           // Threshold
        cgt
        rjumpc OUT         // If > Threshold go to OUT
        pushc BEGIN        // Else go to BEGIN
        jumps
        // ALARM
OUT       pushc num_packets    // ID of the virtual
              sensor
        sense
        pushc 2             // Number of tuple fields
        out                 // Insert the alarm tuple
        rjumpc REDTOGGLE
        // EXIT (ERROR)
        halt
        // EXIT (OK)
REDTGL  pushc 8
        sleep
        pushc 25
        putled
        halt
```

Fig. 10. TupleOut Agent

```
pusht VALUE             // Type
pushrt num_packets      // Sensor ID
pushc 2          // Number of fields
pushc DO
regrxn
```

Fig. 11. Agilla reaction

This simple demo application has been very useful to validate the first extension made to the AGILLA middleware and to give the flavor on how AGILLA agents can be used to implement the presented security framework.

7. Compliance to trusted computing paradigm

As a further aspect in performance assessment and evolution perspectives, it is worth noting that the proposed platform can be compliant to the emerging trusted computing guidelines (TCG, n.d.). Nevertheless, some attention should be paid in the mapping process of roles and functions defined in (TCG Best Practice Committee, 2011) to the underlying technology and application scenario of our Secure Platform: as stated in the Introduction, the main service supported and enabled by the Secure Platform consists in monitoring structural and functional health of industrial plants, which indeed can be configured as an “industrial” service. An item-by-item preliminar analysis of compliance to TCG paradigm has lead to the following results.

- **Security**: security modules embedded into the proposed platform can achieve controlled access to some critical secured data (e.g. monitoring measurements). They also provide reliable measurements and reports of the system's security properties through the ciphered mobile code transfer mechanism among sensor nodes. The reporting mechanism can be fully kept under the owner's control through proprietary format messages feedbacks.
- **Privacy**: data mainly refer to physical quantities related to the industrial plant under monitoring in the considered application scenario. Detection data and observable results are transmitted and stored in ciphered mode among Tuple Spaces and the random nature of some one-shot cryptographic parameters (see Sec. 3.3 h) enhance service confidentiality and reliability, so that the system can be reasonably made compliant to all relevant guidelines, laws, and regulations applicable to this case.
- **Interoperability**: the adoption of both a platform-based design and a cross-layered architecture configures primitives, interfaces and protocols as building blocks of the platform model; therefore, the conformance to TCG specifications [TCG WG, 2007] can be achieved when compatible with resource limitations of the underlying WSN.
- **Portability of data**: it does not completely apply in the considered application scenario, as the definition of alarms and observables is based on limited temporal sequences which are gradually overwritten in each Tuple Space.
- **Controllability**: the analysis of this item requests some clarifications about the role of "owner" and "user": in the considered application scenario, for security and safety reasons, the owner of the platform necessarily coincides with the owner of the system under monitoring and the "user" can ultimately be represented by a specialized operator. User-related information is not present in the system and it never affects service operations, the relationship owner - user thus being strictly hierarchical: therefore, some sub-items cannot apply (e.g. the user be able to reliably disable the TCG functionality in a way that does not violate the owner's policy).
- **Ease-of-use**: usually specialized SW applications (installed at user premises) devoted to post-processing and decision support are comprehensible and usable by specialized trained personnel.

8. Conclusions and perspectives

In this chapter we have proposed novel contributions about definition of cryptography and anomaly detection rules in wireless sensor networks and their implementation in a cross-layered framework design that we denote "Secure Platform". Security functions are executed autonomously by nodes in the network without any support from outside (like servers or database). The proposed schemes have been validated using MATLAB simulations and a prototype implementation through mobile agents supported by a MicaZ wireless sensor network. This work is a partial achievement of the internal project WINSOME (**WI**reless sensor **N**etwork-based **S**ecure system f**Or** structural integrity **M**onitoring and Al**E**rting) at DEWS, whose target is to develop a cross-layer secure framework for advanced monitoring and alerting applications.

Current work is concerned with several developments. One objective is to extend WIDS to detect anomalies in data message content and signaling as well: in this frame bayesian analysis and decision techniques (e.g. the Dempster-Shafer theory) have been successfully

applied in traditional networks where resource availability is not a problem, but in WSNs it might be a big issue. Current research, jointly done with our research partners, deals with this topic and we are extending the Weak Process Models approach to this case and to derive new "threat observables" in WIDS. Another important issue is to consider monitoring as a component in a control process where correlated actuations on the environment can be performed. This vision implies the integration of Hybrid System Control (Di Benedetto et al., 2009) items into the service platform. Another issue consists in the definition of the defense strategy in IDS: rather than listing the possible countermeasures, the question is about how to schedule the priorities in case of multiple interventions on the network. A multi-constraints (hazardousness and distribution of the estimated threat vs. available resources) optimization problem can be a solution.

Finally, from a signal processing and communication viewpoint, some efforts have been already devoted to optimize the information flow on WSNs: the existing correlation among measurement information taken from "contiguous" sensing units should be exploited to increase coding efficiency without losses (the Slepian-Wolf coding theory).

9. Acknowledgment

We would like to thank Dr. Annarita Giani (UC Berkeley) for the joint work and the long discussions, so stimulating and profitable, on our common research items. Thanks to AGILLA Project group and, in particular, Dr. Chien-Liang Fok (Washington University at St. Louis). We also would like to thank Ms. Francesca Falcone and Ms. Catia Maiorani, two Master students of the University of L'Aquila who have actively collaborated to some of the design and implementation activities reported in this chapter during their thesis work.

Furthermore, the research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement n° 257462 HYCON2 Network of excellence and has been motivated and supported by the ESF-COST Action IntelliCIS (Prof. Fortunato Santucci is participating to this Action).

10. References

Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y. & Cayirci, E. (2002). A Survey on Sensor Networks, *IEEE Communications Magazine*, August 2002

Al-Subaie, M. & Zulkernine, M. (2006). Efficacy of Hidden Markov Models Over Neural Networks in Anomaly Intrusion Detection, *Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC)*, vol. 1, 2006

Bai, H.; Atiquzzaman, M. & Lilja, D. (2004). Wireless Sensor Network for Aircraft Health Monitoring, *Proceedings of Broadband Networks'04*, 2004

Baker, Z., & Prasanna, V. (2005). Computationally-efficient Engine for flexible Intrusion Detection, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, n. 10, 2005

Barbaràn, J.; Diaz, M.; Esteve, I. & Rubio, B. (2007). RadMote: A Mobile Framework for Radiation Monitoring in Nuclear Power Plants," *Proceedings of the 21st International Conference on Computer, Electrical, Systems Science and Engineering (CESSE'07)*, 2007

Bhatnagar, R.; Srivastava, A. K. & Sharma, A. (2010). An Implementation Approach for Intrusion Detection System in Wireless Sensor Network, *International Journal on Computer Science and Engineering*, vol. 2, no. 7, 2010

Certicom Research Standards, http://www.secg.org/

Cho, S.; Yun, C.-B.; Lynch, J. P. ; Zimmerman, A.; Spencer Jr B. & Nagayama, T. (2008). Smart Wireless Sensor Technology for Structural Health Monitoring of Civil Structures, *International Journal of Steel Structures, KSSC*, pp. 267-275, 2008

CROSSBOW Inc., http://www.xbow.com/

Dainotti, A.; Pescapè A.; Rossi, P.; Palmieri, F. & Ventre, G. (2008). Internet Traffic Modeling by means of Hidden Markov Models, Computer Networks, Elsevier, vol. 52, no. 14, 2008

Debar, H.; Dacier, M. & Wespi, A. (1999). Towards a Taxonomy of Intrusion-Detection Systems, *International Journal of Computer and Telecommunications Networking*, pp. 805–822, 1999

Di Benedetto, M. D.; Di Gennaro, S. & D'Innocenzo, A. (2009). Discrete State Observability of Hybrid Systems, *International Journal of Robust and Nonlinear Control*, vol. 19, n. 14 2009

Diffie, W. & Hellman, M.E. (1976). New Directions in Cryptography, *IEEE Transactions on Information Theory, IT-22:644–654*, November 1976

Doumit, S. & Agrawal, D. (2003). Self Organized Critically and Stochastic Learning Based Intrusion Detection System for Wireless Sensor Networks, *Proceedings of the Military Communications Conference (MILCOM)*, 2003

Ephraim, Y. & Merhav, N. (2002). Hidden Markov Processes, *IEEE Trans. Information Theory*, vol. 48, no. 6, 2002

Eschenauer, L. & Gligor, V.D. (2002). A key-management Scheme for Distributed Sensor Networks, *Proceedings of the 9th ACM Conference on Computer and Communications Security*, 2002

Flammini, F.; Gaglione, A.; Mazzocca, N.; Moscato, V. & Pragliola, C. (2008). Wireless Sensor Data Fusion for Critical Infrastructure Security, *Proceedings of International Workshop on Computational Intelligence in Security for Information Systems, CISIS'08*, 2008

Fok, C.-L.; Roman, G.C. & Lu, C. (2006). Agilla: A Mobile Agent Middleware for Sensor Networks, *Technical Report, Washington University in St. Louis,* WUCSE-2006-16, 2006

Forney, G. (1973). The Viterbi Algorithm, *Proceedings IEEE*, vol. 61, pp. 263–278, 1973

Gay, D.; Levis, P.; von Behren, R.; Welsh, M.; Brewer, E. & Culler, D. (2003). The nesC Language: A Holistic Approach to Networked Embedded Systems, *Proceedings of ACM SIGPLAN*, 2003

Gura, N.; Patel, A. & Wander, A. (2004). Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs, *Proceedings of the 2004 Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004)*, 2004.

Hadim, S. & Nader, M. (2006). Middleware: Middleware Challenges and Approaches for Wireless Sensor Networks, *IEEE Distributed Systems on-line 1541-4922, IEEE Computer Society*, vol. 7, n. 3, 2006

Hu, F.; Ziobro, J.; Tillett, J. & Sharma, N. (2004). Secure Wireless Sensor Networks: Problems and Solutions, *Journal on Systemic, Cybernetics and Informatics*, vol. 1, n. 9, 2004

Ioannis, K.; Dimitriou, T. & Freiling, F. C. (2007). Towards Intrusion Detection in Wireless Sensor Networks, *Proceedings of the 13th European Wireless Conference*, 2007

Jangra, A.; Richa, S. & Verma, R. (2011). Vulnerability and Security Analysis of Wireless Sensor Networks, *International Journal of Applied Engineering Research*, vol. 6, no. 2, 2011

Jiang, G. (2005). Robust Process Detection using Nonparametric Weak Models, *International Journal of Intelligent Control and Systems*, vol. 10, 2005

Kalita, H. K. & Kar, A. (2009). Wireless Sensor Networks Security Analysis, *International Journal of Next-Generation Networks*, vol. 1, n. 1, 2009

Kaplantzis, S. (2004). Classification Techniques for Network Intrusion Detection, *Technical Report*, Monash University, 2004

Karlof, C. & Wagner, D. (2003). Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures, *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, vol. 10, 2003

Khanna, R. & Liu, H. (2006). System Approach to Intrusion Detection Using Hidden Markov Model, *Proceedings of the International Conference on Wireless Communications and Mobile Computing*, vol. 5, pp. 349 - 354, 2006

Kim, S.; Pakzad, S.; Culler, D.; Demmel, J.; Fenves, G.; Glaser, S. & Turon, M. (2007). Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks, *Proceedings of the 6th International Conference on Information Processing in Sensor Networks IPSN 07*, 2007.

Kliazovich, D.; Devetsikiotis M. & Granelli, F. (2009). Formal Methods in Cross Layer Modeling and Optimization of Wireless Networks, Handbook of Research on Heterogeneous Next Generation Networking, 2009, pp. 1-24.

Koblitz, N. (1987). Elliptic Curve Cryptosystems, *Mathematics of Computation*, vol. 48, pp. 203-229, 1987

Kumari, P.; Kumar, M. & Rishi, R. (2010). Study of Security in Wireless Sensor Networks, *International Journal of Computer Science and Information Technologies*, vol. 1, n. 5, 2010

Law, Y.; Havinga, P. & Johnson, D. (2005). How to Secure a Wireless Sensor Network, *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2005

Levis, P. & Culler, D. (2002). Matè: a Tiny Virtual Machine for Sensor Networks, *Proceedings of the 10th International Conference on Architectural support for programming languages and operating systems*, ACM Press, 2002

Liu, A.; Kampanakis, P. & Ning, P. (2008). TinyECC: Elliptic Curve Cryptography for Sensor Setworks (v.0.3), http://discovery.csc.ncsu.edu/software/TinyECC/, 2008

Liu, T. & Martonosi M. (2003). Impala: A Middleware System for Managing Autonomic Parallel Sensor Systems, *Proceedings of ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP 2003)*, 2003

Loo, C.; Ng, M.; Leckie, C. & Palaniswami, M. (2005). Intrusion Detection for Routing Attacks in Sensor Networks, *International Journal of Distributed Sensor Networks*, 2005

Luk, M.; Mezzour, G.; Perrig, A. & Gligor. V. (2007). MiniSec: A Secure Sensor Network Communication Architecture, *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN)*, 2007

Malan, D.; Welsh, M. & Smith, M. (2004). A Public-key Infrastructure for Key Distribution in TinyOS based on Elliptic Curve Cryptography, *Proceedings of IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2004

MEMSIC Inc., http://www.memsic.com

Menezes, A. J.; Van Oorschot, P. & Vanstone, S. A. (1996). Handbook of Applied Cryptography, CRC Press (Ed.), ISBN 0-8493-8523-7, New York, 1996

Pugliese, M. & Santucci, F. (2008). Pair-wise Network Topology Authenticated Hybrid Cryptographic Keys for Wireless Sensor Networks using Vector Algebra, *Proceedings of the 4th IEEE International Workshop on Wireless Sensor Networks Security (WSNS08)*, 2008

Pugliese, M.; Giani, A. & Santucci, F. (2008). A Weak Process Approach to Anomaly Detection in Wireless Sensor Networks, *Proceedings of the 1st International Workshop on Sensor Networks (SN08)*, Virgin Islands, 2008

Pugliese, M.; Giani, A. & Santucci, F. (2009). Weak Process Models for Attack Detection in a Clustered Sensor Network using Mobile Agents, *Proceedings of the 1st International Conference on Sensor Systems and Software (S-CUBE2009)*, Pisa, 2009

Pugliese, M.; Pomante, L. & Santucci, F. (2009). Agent-based Scalable Design of a Cross-Layer Security Framework for Wireless Sensor Networks Monitoring Applications, *Proceedings of the International Workshop on Scalable Ad Hoc and Sensor Networks (SASN2009)*, Saint Petersburg, 2009

Rabiner, L., & Juang, B. (1986). An Introduction to Hidden Markov Models, *IEEE ASSP Magazine*, 1986

Roman, R.; Zhou, J. & Lopez, J. (2006). Applying Intrusion Detection Systems to Wireless Sensor Networks, *Proceedings of the 3rd IEEE Consumer Communications and Networking Conference*, 2006

Roosta, T.; Shieh, S. & Sastry, S. (2006). Taxonomy of Security Attacks in Sensor Networks, *Proceedings of 1st IEEE International Conference on System Integration and Reliability Improvements*, vol. 1, pp. 529-536, 2006

Sangiovanni-Vincentelli, A. & Martin, G. (2001). Platform-based Design and Software Design Methodology for Embedded Systems, *Proceedings of IEEE Computer Design & Test*, vol. 18, n. 6, 2001

Sharma, R.; Chaba, Y. & Singh, Y. (2010). Analysis of Security Protocols in Wireless Sensor Network, *International Journal of Advanced Networking and Applications*, vol. 2, n. 2, 2010

Sheng, Y. & Cybenko, G. (2005). Distance Measures for Nonparametric Weak Process Models, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, 2005

Singh, V. P.; Jain S. & Singhai, J. (2010). Hello Flood Attack and its Countermeasures in Wireless Sensor Networks, *International Journal of Computer Science Issues*, vol. 7, n. 11, 2010

Szumel, L.; LeBrun, J. & Owens, J. D. (2005). Towards a Mobile Agent Framework for Sensor Networks, *2nd IEEE Workshop on Embedded Networked Sensors (EmNetS-TT)*, 2005

TinyOS, http://www.tinyos.net

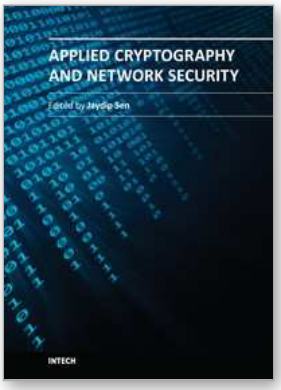
TCG, http://www.trustedcomputinggroup.org

TCG Best Practise Committee (2011). Design, Implementation, and Usage Principles (v.3.0), February 2011

TCG WG (2007). TCG Specification Architecture Overview Design (rev. 1.4), August 2007

Whitman, M. & Mattord, H. (2011). Principles of Information Security, Thomson (Ed.), Fourth Edition, ISBN-13 978-1-111-13821-9, 2011

Yin, Q., Shen, L., Zhang, R., Li, X., & Wang, H. (2003). Intrusion Detection Based on Hidden Markov Model, *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 5, 2003



Applied Cryptography and Network Security
Edited by Dr. Jaydip Sen

ISBN 978-953-51-0218-2
Hard cover, 376 pages
Publisher InTech
Published online 14, March, 2012
Published in print edition March, 2012

Cryptography will continue to play important roles in developing of new security solutions which will be in great demand with the advent of high-speed next-generation communication systems and networks. This book discusses some of the critical security challenges faced by today's computing world and provides insights to possible mechanisms to defend against these attacks. The book contains sixteen chapters which deal with security and privacy issues in computing and communication networks, quantum cryptography and the evolutionary concepts of cryptography and their applications like chaos-based cryptography and DNA cryptography. It will be useful for researchers, engineers, graduate and doctoral students working in cryptography and security related areas. It will also be useful for faculty members of graduate schools and universities.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821