# Anomaly Detection in RFID system

Khaled Garri, Francoise Sailhan, Samia Bouzefrane

# Anomaly detection in RFID system

Khaled Garri, Francoise Sailhan, Samia Bouzefrane

Cedric Laboratory, Conservatoire National des Arts et Métiers, 292 rue Saint Martin, 75141, Paris, France

Name.surname@cnam.fr

**Abstract:** The level of sophistication exhibited by RFID tags is not only affecting their financial cost but also their ability to provide (extensive) cryptographic functionalities. It follows that low-cost tags offer no real control access. To tackle this issue, we propose an anomaly detection system which attempts to identify deviations from the normal behavior. In practice, subjects are equipped with RFID tags in order to be constantly located. Then, a user's profile is built, relying on the Kohonen's maps that constitute an efficient way for automatically categorizing and further compare the tag behavior against the normal user's behavior (as expressed in the user's profile).

**Keywords:** Anomaly detection system, RFID.

**Biographical notes:** K. Garri did his MTech in mobile and embedded system from Conservatoire National des Arts et Métiers (CNAM), Paris, France and is currently doing his PhD in Computer Science in CNAM. His areas of interest are intrusion detection, RFID and java cards.

Francoise Sailhan is associate professor in CNAM. Prior she was professor assistant in the University of Franche Comté, France, and worked as researcher in the Ericsson Ireland Research Center (EIRC), Ireland. In 2005, she successfully defended a PhD thesis at the University Pierre and Marie Curie, France, her Phd work taking place in INRIA, France**.** Her research interests are wireless networks & intrusion detection.

Samia Bouzefrane is presently associate professor in CNAM and prior in the University of Le Havres, France. She received her PhD in Computer Science from the University of Poitiers. Her research interests involve Java cards, components in real-time and embedded system, real-time data base.

## 1. Introduction

Radio Frequency IDentification (RFID) is a technology which is primarily intended to identify automatically any object. As such, RFID is nowadays considered as one of the mostly used wireless technology in security-related domains including, electronic payment, access control and transport. A RFID system consists of tree main components, a RFID tag (basically, a silicon microchip attached to an antennae and possibly enriched with additional functionalities e.g., sensing, storage, encryption), a RFID reader (a transceiver communicating with tags *via* radio frequency and typically containing internal storage and processing capabilities so as to perform tasks on behalf of the tag) and a back-end database (if any) connected to the reader. These unprotected components are naturally subject to various threats favored by the networked nature of RFID. For instance, the clandestine scanning of tags is completed wherever the read range permits to do so. Such scanning remains undetected recall that the tag responds to reader interrogation without alerting its owner. In addition, once a reader powers a tag, another reader may monitor the resulting tag emission without itself outputting a signal, i.e., it eavesdrops the detection range. Such a misbehaving reader that harvests information from a well-behaving tag is the starting point of privacy concerns, especially when the tag serial numbers are combined with personal data. In the other way around, a well behaving reader may also harvest information from misbehaving tags. Note that this threat which is closely related to tag authentication is of paramount importance seeing that identification is the main purpose of RFID.

In order to deal with the above outlined (privacy and security) issues[1], one widely used approach consists in relying on cryptographic methods. As illustration, the Exxon-Mobil Speed pass [2] refers to a payment system for gasoline, which is based on the TI-DST (Texas Instruments DIGITAL Signal Repeater) tags used to authenticate customers. This system relies on short-length cryptographic keys (40-bits) which provide relatively-weak protection [BG05]. Note that, generally speaking, the level of sophistication of RFID tags and readers, not only affects their relative financial cost but also their ability to provide extensive (cryptographic) functionalities (e.g., encryption, strong pseudorandom, number generation, and hashing). It follows that low-cost RFID tags offer no real access control. This circumvents the need for complementing cryptographic methods (if any) with advanced anomaly or intrusion detection. Research tackling anomaly detection in RFID systems still remains in its infancy with a research effort [LMF07,EV10,YGD10] focusing on (i) the threats targeting supply chains, (ii) attacks launched by some rogue readers given no specific use-case scenario [TS08], and control access threats (i.e., changes in the tag ownership) [MH07]. In this last case, anomaly is detected based on a simple statistical method (i.e., standard deviation and mean). We propose an anomaly detection also taking into account a control access scenario. In practice, users (staff) working within a building, are equipped with RFID tags which are used to constantly monitor the user's location. We then record a user's profile including the series of Cartesian coordinates of that user. Nevertheless, instead of relying on simple statistical method, we select the Kohonen's self-organizing maps [TK82] as an advanced neural network architecture permitting to build the user's profile as an ordered representation of spatial proximity among vectors of an unlabelled data set. The reason that motivates this choice is twofold. First, Kohonen's maps are recognized for their ability to automatically categorize the inputs provided during the training phase without supervision and for rating efficiently whether subsequent information fits any of the learned categories. Second, they permit to enrich easily the user profile, i.e., without necessitating major implementation changes. Consequently, based on an advanced Kohonen's map, our detection system detects any spoofing attack wherein an adversary mimics an authentic tag and any usage of a robed tag, because these intrusions, by assumption, will deviate from normal usage of the customer. We further prototyped the proposed anomaly detection and conducted preliminary experiments.

[1] An overview on RFID threats and privacy concerns can be respectively found in [ABK10] and [ML09].

This paper is organized as follows. We first define the use-case targeted by our anomaly detection system and evaluate existing systems against the requirements driven by our use case (Section 2). Then, we present the proposed anomaly detection system (Section 3) and further evaluate its performances (Section 4). We finally conclude with a summary of our contribution along with directions of future work (section 5).

## 2. Control Access based on RFID

We consider a scenario that consists in controlling access to the buildings of a Computer Science laboratory, and in detecting possible intruders. Each researcher of the laboratory owns a cell phone rather than a badge, used to access the laboratory building. The cell phone is endowed with a RFID tag. This constitutes the first level of security. In order to detect an intruder who enters in the building despite of the access check, we add to our RFID system, a traceability system built thanks to the geo-localization of the tags carried by the staff. This constitutes the second level of security.

In this scenario, only one category of tag attacks is considered: the stolen, cloned and spoofed tags. If an intruder accesses a laboratory building, her/his behaviour differs from a legitimate person that owns some habits when she/he is working within the laboratory. The idea is to build a reference model based on the habits of the laboratory researchers within a time period and then to build another model when we want to detect intrusions. The new model is then compared to the reference model. If a significant deviation is observer/computed, a possible intrusion is consequently detected.

Within such a scenario, the trajectory of any subject is sampled at discrete time intervals $t_{1,}$ ..., $t_{k,}$ ..., $t_m$ with $m$ defining the trajectory length. Any observation is expressed as a set of $m$-dimensional real *vectors* $(x(t_1), ..., x(t_k), ... x(t_m))$. Note that we assume more samples than rows in the observation (i.e., $m >> n$). A trajectory is hence composed of spatio-temporal records, each record being primarily composed of:

- A geographical location within a 1D, 2D, 3D plan,
- A temporal attributed, i.e., a timestamp. Note that records are collected at arbitrary time interval.

In addition to the above, extra pieces of information may be added or inferred from the spatio-temporal records defined above. They relate to the e.g., duration separating two samples, maximum speed, (estimated) attractor point, direction, movement pattern (e.g., loop, u-turn) and the average, or standard deviation of the aforementioned parameters.

### *3.1 Related Work*

Research tackling anomaly detection in RFID systems still remains in its infancy. Two anomaly detection systems [MH07, TS08] have been initially proposed. In order to find an abnormal behavior (e.g., a change in the tag ownership), both rely on a statistical method (i.e., standard deviation and mean) inspirited by the pioneering[1] work of Denning [DE87]. Precisely, the former measures the number of times a user logs into a system (i.e., the number of tag reads) at different locations, whereas the latter also encompasses the number of tag writes, the time interval between two readings (versus 2 writings) and the received signal strength. Based on the aforementioned indicators, the former identifies changes in the ownership (as it is the case with e.g., cloned or robed tags) whereas the latter introduces the notion of watchdog reader, i.e., a reader dedicated to

[1] For a survey on anomaly detection, interested reader may refer to [CBK09].

monitoring tags and readers in its reading range so as to detect a MIM (Man In the Middle), i.e., a malicious reader that either writes false data on writeable tag or intercepts reading request and relays it to a malicious tag emulator so as to provide to a malicious user an access to the tag reader. The observations cached by any reader and watchdog reader are forwarded to the anomaly detection system and any deviation is defined as an anomaly. Still based on statistical data, the intrusion detection system proposed in [YGD10] makes use of the rate of command matching, password succeeding and Cyclic Redundancy Check (CRC) fails in order to detect intrusions relating to password guessing, DOS (Denial Of Service) based on e.g., RF signal interfering and MIM. The basic idea is that attacks are made of test operations that usually fail. Thus, a *ratio* e.g., number of succeeding passwords over the total number of attempts, is used to define a danger signal, which is further collaboratively detected relying on artificial immune system. In [EV10], the notion of location is refined by distinguishing physical and semantic location (i.e., a geographical location/area wherein a RFID reader operated and the meaning of that location/region, e.g., a room number). The interpretation of the (physical and semantic) location information is further facilitated by relying on an ontology-based intrusion detector which makes use of an inference system in order to automatically reason on anomalies. In practice, an anomaly refers to a RFID tag that is either read to many times within a fixed duration with respect to the usage condition or not moving according to the static path in the supply chain (as predefined in the object profile). Note that, similarly, this last indicator is used in [LMF07] in order to pinpoint illicit players that inject counterfeits tagged objects in a licit supply chain. Nevertheless, in this latter case, the intrusion detection is obtained relying on hidden Markov chains rather than rules.

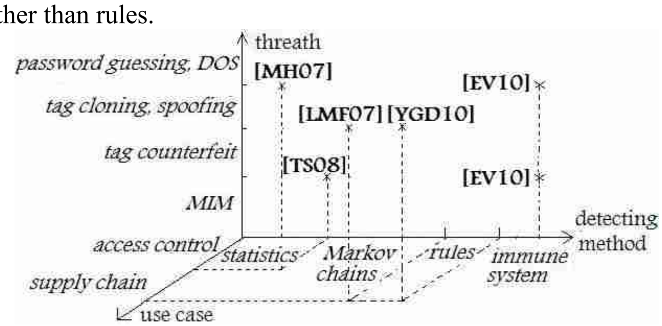


**Figure 1: Taxonomy**

| *reference* | *threat* | *use case* | *detection method* | *Profile* | *anomaly indicators* |
|---|---|---|---|---|---|
| MH07 | tag cloning & spoofing | simulated attack given a real-world test-bed: the access control in a computer science department | statistics | tag profile: profile name, read/write operation, value of current observation, past observations | number of times a tag has been used |
| TS08 | man in the middle | simulated RFID network based on RFIDSim [MIL06] | statistics | The tag profile is coupled with the reader profile and is divided into a read operation profile (tag id, read operation, location, timestamp) and a write | read/write frequencies, time interval between two consecutive operations, RSS-based location |

| | | | | | |
|---|---|---|---|---|---|
| | | | | operation profile (tag id, read operation, timestamp) | |
| YGD10 | password guess, RFID skimming, DOS | Simulated supply chain | artificial immune system | Following the EPC C1G2 standard [EPC07] of EPC Global, the profile is given by (timestamp, location, reader command, EPC code, operation return flag code) | rate of: operation matches, password matches, tag responses, CRC errors |
| EV10 | Tag counterfeit | simulated supply chain | Ontology -based inference system | Timestamps & locations in the supply chain | time interval between 2 read operations and difference with the normal static path in the supply chain |
| LMF07 | Tag counterfeit | simulated supply chain | Hidden Markov chains | Timestamps & locations in the supply chain | Difference with the normal static path in the supply chain |

**Table 1. Classification of anomaly detection**

Overall, two main usages of RFID system have been considered in the aforementioned literature (see Figure 1 and Table 1 for a summary): the control access in a building [MH07] (actually the computer science department of the Tasmania University) and the supply chains [TS08,YGD10,EV10,LMP07]. The performance of the proposed anomaly detectors has been evaluated based on some simulated attacks operating either over a real test bed (for the former) or a simulated RFID system (for the latter). Mostly focused on the tag-reader relation, envisioned threats include (i) the tag cloning/spoofing attack which may lead to e.g., the insertion of a counterfeit objects in a supply chain or an access granted to the computer department, and, (ii) the MIM or DOS attack launched by a rogue reader. Indicators of such threats fall into two categories:

- *Operational:* indicators refer to some repeated commands/operations (e.g., read/write, password check) that are either failing or differing from their normal usage (i.e., deferring from the user's habits),
- *Spatial:* indicators correspond to (i) the path followed by a tag which may differ from the well-established one or the reader position which is identified based on the signal strength and indicates the potential presence of a rogue reader.

Once recorded into an object/user's profile, one or a combination of the above indicators is used to detect intrusion, relying for this purpose on different methods ranging from statistics, hidden Markov chains, rules, up to artificial immune system.

In this context, we propose a self-classifying anomaly detection system. As in [MH07], we focus on a control access scenario wherein users (staff) working within a building, are equipped with active RFID tags, which are used to constantly monitor the user's location based on the signal strength. Thus, fine-grained and continuous user localization can be provided. Given this specific use case, anomalies are primarily[1] detected based on spatio-temporal indicators rather than operational indicators. Instead of relying on a simple statistical method, we select an advanced neural network architecture permitting to build automatically, i.e., without user's/expert's supervision, the user's

[1]

profile. Note that such automatic training permits to add easily additional indicators (i.e., operational indicators), i.e., without modifying the core implementation. Consequently, based on an advanced Kohonen's map, our detection system detects any spoofing/cloning attack wherein an adversary mimics an authentic tag and any usage of a robed tag, because these intrusions, by assumption, will deviate from normal usage of the customer.

## 3. Anomaly Detection

When attempting to detected anomaly, the main difficulty lies in defining what a normal *versus* abnormal behaviour is. An advantage of self-organising map is that it learns to discriminate normal behavior from abnormal behavior based on examples (i.e., training samples). Thus, no explicit definition of normal/abnormal behavior is required to the user. Our anomaly detection system is based on the Kohonen map [TK82]. In a nutshell, a Kohonen map is a neural network that distinguishes itself by its unsupervised learning. Another convenient aspect is related to the fact that this map reduces the dimensionality of the input data from a (potentially) high dimension into 2- or 3-dimensional space (herein 2-dimensional), hence allowing an easy and instinctive interpretation of the results. In practice, a Kohonen-map-based detection of anomaly involves the following three phases:

- The pre-processing phase (Section3.1) consists in filtering the raw data provided by the RFID system,
- The training phase (Section 3.2) aims at learning the habits of the subjects in order to build Kohonen maps, and,
- The anomaly detection phase (Section 3.3) makes use of Kohonen maps in order to detect anomalies.

### *3.1 Raw Data Preprocessing*

Anomaly detection is intended to identify activities that vary from an established pattern. This necessitates to (i) create a knowledge database constituted of the (previously) monitored activities and to (ii) subsequently categorize the variety of stored data relying for this purpose on the Kohonen maps. Prior being provided as input to the Kohonen maps, data is pre-processed, following a two steps process:

- *Data filtering* - Data provided by the RFID system are filtered so as to extract information that is relevant to anomaly detection.
- *Data normalizing* - Normalising input samples consists in scaling the initial data set so as to fall in the specific [0,1] range. In practice, a set of input samples that are collected at $t_1, \ldots, t_k, \ldots, t_m$ is expressed as a set of m-dimensional vectors $(x(t_1), \ldots, x(t_k), \ldots x(t_m)) \in R^m$, with each vector $x(t_k)$ describing a monitored activity. Such activity is defined as a vector of $n$ dimensions $x^T(t_k) = (x_1(t_k), \ldots, x_i(t_k), \ldots, x_n(t_k))$, which, once normalised, verifies:

$$x'^T(t_k) = (x'_1(t_k), \ldots, x'_i(t_k), \ldots, x'_n(t_k)), \qquad (1)$$

$$with\ x'_i(t_k) = \frac{x_i(t_k)}{\max_{j\in[1,n]}\left(x_{ij}(t_k)\right)}$$

As a result of these filtering and normalisation processes, workless samples are removed and each filtered sample is of equal footing and can be exploited during the training phase in order to create a Kohonen map.

### *3.2 Training phase*

The result of the training phase is a Kohonen map that corresponds to a topological 2-dimensional array of neurons originally initialised with random values. This map results from the categorisation of the samples $x'(t_1), \ldots, x'(t_k), \ldots x'(t_m)$ provided as input.
In practice, each input vector $x'(t_k)$ with $k \in [1, m]$, is compared with each neuron forming the Kohonen map and a distance between the input vector and this neuron is computed. Finally, the closest neuron is selected as the winning neuron. Then, the topological structure of the Kohonen map is updated: neurons that are topologically close to the winner move towards its direction. Consequently, the resulting Kohonen map reflects a categorisation (clustering) of the samples.
More particularly, assuming a measure, whose norm is noted ||, the distance between an input vector $x(t_k)$ and the synaptic vector of all the neurons $w_i\ (t_k)$ in the map is computed and the winner $g(x'(t_k))$ is selected according to the following law:

$$g\ (x'(t_k)) = min_{\ i \in [1, s]}\ ||\ (x'(t_k)\ ,\ w_i\ (t_k))\ ||, \qquad (2)$$

with $s$ defining the Kohonen map size.

Next, the neurons that are topologically close to the identified neuron move towards the direction of the winner. For this purpose, the neurone $w_i$ is updated as follows:

$$w_i(t_k+1) = w_i(t_k) + \pi_{i,\ g(x(tk))}\ (t_k)\ .\ \eta\ (t_k)\ .\ [(x'(t_k)\ -\ w_i\ (t_k)], \qquad (3)$$

with $(i,\ j,\ k) \in [1,\ s]^2 \times [1,\ m]$ and $\eta(t_k)$ defining an adaptation factor that controls the degree of change imposed to the neuron vector and $\pi_{i,\ g(x'(tk))}\ (t_k)$ a neighbouring function centred around the winner $g(x'(t_k))$

. Note that both $\eta\ (t_k)$ and $\pi_{i,\ g(x'(tk)}\ (t_k)$ depend of the time $t_k$. The basic idea is that the adaptation factor $\eta\ (t_k)$ decreases monotonically as the learning phase progresses so as to guarantee a convergence of the weighted neuron's vector towards a stable state [LB91]. For that purpose, $\eta\ (t_k) = \eta_0\ exp\ (t_k/\ t_m)$. Similarly, the neighbouring function $\pi_{\ i,\ g\ (x'(tk))}\ (t_k)$ decreases as $t$ evolves until the winning neuron is the only neuron that has its weight significantly updated. For this purpose, $\pi_{\ i,\ g\ (x'(tk))}\ (t_k)$ is defined as a symmetric function following a Gaussian form[1] with a standard deviation *$\sigma\ (t_k)$* decaying exponentially with time:

$$\pi_{i,g(x'(t_k))}(t_k) = \exp\left(\frac{\left\|\left(x'(t_k), w_i\right)\right\|^2}{2\sigma^2(t_k)}\right), \text{ and} \quad (4)$$

$$\sigma(t_k) = \sigma_0 . \exp\left(\frac{-t_k \log(\sigma_0)}{t_m}\right)$$

Overall, the Kohonen map is updated based on this neighbouring notion which permits to classify the observations, i.e., to group neurons into clusters characterised by their high training density. The training phase can be expressed as follows:

***Algorithm 1. Training phase***

*Parameters:*

*Given,*

- *A set of pre-processed and n-dimensional input samples noted $x'(t_1)$, …, $x'(t_k)$, … $x'(t_m)$), collected at $t_1$, …, $t_k$, …, $t_m$ , with each of these samples characterised by $x'^T(t_k) = (x'_1(t_k), \ldots, x'_i(t_k), \ldots, x'_n(t_k))$*
- *A measure and its related norm noted ||*
- *The size s of a 2-D Kohonen map,*
- *A function random function Rand ([a, b]) that provide as output a random number belonging to [a, b]*
- *a threshold α*

***-- Map initialisation***

*$\forall i, j \in [1, n] . [1, s], w_{ij} = rand([0,1])$*

***-- Winner selection & kohonen map update***

*$\forall t_k \in [t_1, t_m]$*

*$g(x'(t_k)) = \min_{i \in [1, s]} || (x'(t_k), w_i(t_k)) ||$,*

*$\forall w_i \ni: || wi - g(x'(t_k)) || < \alpha$*

*$\eta(t_k) = \eta_0 \exp(t_k / t_m)$*

*$\sigma(t_k) = \sigma_0 \exp(- t_k . \log(\sigma_0) / t_m)$*

[1] A Gaussian form facilitates the ordering of the neighboring set, yielding to faster convergence [LB91].

$$\pi_{i,\, g\,(x'(tk))}\,(t_k) = exp\,(-\,||\,(x'(t_k)\,,\, w_i)\,||^{\,2} / \,2\,\sigma^{\,2}(t_k))$$
$$w_i(t_k{+}1) = w_i(t_k) + \pi_{i,\, j,\, g(x(tk))}\,(t_k)\,.\,\eta\,(t_k)\,.\,[(x'(t_k) - w_i\,(t_k)]$$

It is noteworthy that Kohonen algorithm is applicable to large dataset because (i) the computational complexity scales linearly with the number $m$ of samples and (ii) limited memory (i.e., the memory necessary to record the set of training vectors $x'(t_1)$ , ... $x'(t_m)$ and the Kohonen map $w_1 ... w_s$). Nevertheless, complexity is quadratic, hence causing a time-consuming training phase. As a 2D-grid, this Kohonen map is of great help for visualising and inspecting the user behaviour recall that the structure of Kohonen map reflects the structure of the original training samples. Based on the trained Kohonen map, which reflects the normal activity of a subject, any deviation from that normal activity can be detected and identified as an anomaly.

## *3.3 Anomaly Detection*

Central to the notion of anomaly detection is the decision threshold. Intuitively, if the distance between the observed and normal behavior is greater than the threshold, then the observed behavior is defined as anomalous. Given our use case - a RFID-enabled study of the user location within the control access domain - we distinguish two sources of potential anomalies, the user's position and its trajectory. Intuitively, a position is said to be anomalous if it does not belong to any of the classification defined during the training, i.e., if it does not pertain to any of the clusters centered on the wining neurons defined as part as the training phase (Figure 1 and Algorithm 2).

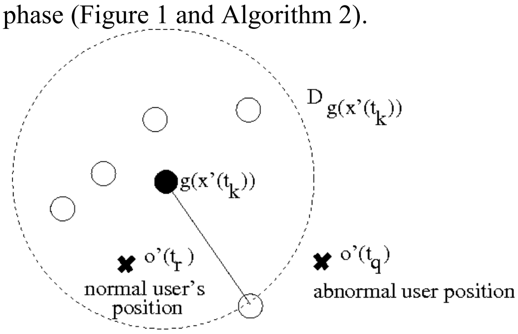


**Figure 2. Detection of a abnormal and normal user position, $o'(t_q)$ and $o'(t_r)$, in a one dimension training set (m = 1) plotted into a 2 dimensional plan: a winning neuron (black circle) is surrounded by some neighboring neurons (white circles) located within a disk (dashed circle).**

By extension, we define that a trajectory is anomalous if a great percentage of the user's position is anomalous, i.e., if the pre-processed observation $o'^T(t_k) = (o'_1(t), ..., o'_i(t), ..., o'_n(t))$ does not pertain to any of the clusters centred around the winning neurons $g\,(x'(t_k))$ and circumvented by the radius defined as the maximum distance $max_{\,w_i\,\epsilon\,D_{g\,(o'(t))}}\,||\,g\,(x'(t_k))\text{-}\,w_i\,||$ separating the winning node $g(x'(t_k))$ from its neighbouring neurons, i.e., the neurons that belong to $D_{g\,(x'(tk))}$. By extension, a trajectory $o'(t), ..., o'(t{+}p)$ is anomalous if the ratio of anomalous position exceeds a given threshold defined by $\beta.p$.

***Algorithm 2. Anomaly detection***

*Parameters:*
*Given*

- *the set of wining neurons* $g(x'(t_1)), \ldots, g(x'(t_k)), \ldots g(x'(t_m))$ *defined during the training phase, with* $g(x'(t_k)) = \min_{i \in [1,s]} \| (x'(t_k), w_i(t_m)) \|$
- *the trajectory* $o'(t), \ldots, o'(t+q), \ldots, o'(t+p)$ *composed of a set of novel pre-processed observation* $o'^T(t+q) = (o'_1(t+q), \ldots, o'_i(t+q), \ldots, o'_p(t+q))$ *with* $t > t_m$ *and* $p > 0$

***-- Detection configuration***

$\forall\, g(x'(t_k))$ *with* $t_k \in [1, m]$

Let $D_{g(x'(t_k))} = \{ w_i \ni : \| w_i - g(x'(t_k)) \| < \alpha \text{ and } w_i \notin D_{g(x'(t_k))} \}$

Let $| D_{g(x'(t_k))} | = \max_{w_i \in D_{g(x'(tk))}} \| wi - g(x'(t_k)) \|$

***-- On detecting an anomalous user's position***

$\forall\, g(x'(t_k))$ *with* $t_k \in [1, m]$

*if* $\| o'(t) - g(x'(t_k)) \| < | D_{g(x'(t_k))} |$
*then* $o'(t)$ *is normal*
*else* $o'(t)$ *is abnormal*

***-- On detecting an anomalous user's trajectory***

$$\text{if} \sum_{q=0}^{p} \sum_{k=0}^{m} \mathbb{1}_{\{\| o'(t_q) - g(x'(t_k)) \| < | D_{g(x'(t_k))} |\}} < \beta . p$$

*then* $o'(t), \ldots, o'(t+p)$ *is normal*
*else* $o'(t), \ldots, o'(t+p)$ *is abnormal*

The computational complexity related to detecting a position and then its trajectory scales linearly the number of winning vectors $g(x'(t_k))$ (bounded by $m$) and with $p \,.\, g(x'(tk))$ (bounded by $p.m$). In addition to the memory allocated to the training phase, little additional memory (basically, the index $i$ of the winning neurons in the Kohonen map and their established radius) is used during the anomaly detection.

## 4. Implementation and Experiments

In order to assess the proposed solution, we implemented the prototype of an anomaly detector (Figure 1). The overall architecture includes: a RFID system that consists of RFID tags, RFID readers and a back-end database fed by a RFID middleware connected to the readers. In order to detect any anomaly, information provided by the RFID middleware is recorded leading to the creation of a backend database.

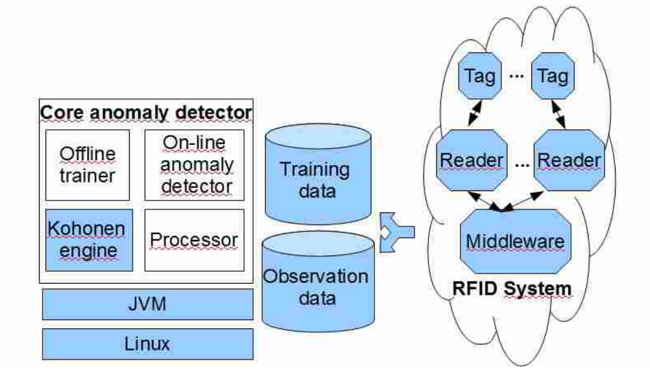


**Figure 3. Anomaly detection in a RFID system**

For the sake of clarity, we distinguish two backend databases, one devoted to the training and one to gathering observations. Both are constituted of the data collected from the RFID system according to the scenario defined in Section , but, their usage differs: the former is used in order to train the anomaly detector whereas the latter serves so as to identify threats. These two activities are performed by the core anomaly detector that can be broken down into:

- A processor which extracts the information from the database in order to parse, filter and normalize it. In practice, data is stored in the database as XML files. The resulting information is then provided either to the trainer so as to build the user's normal behavior or to the anomaly detector in order to detect intrusion attempt.
- A trainer that takes as input the processed training data in order to create a Kohonen Map. This training phase which is performed off-line, permits to classify the user's behavior whereas the anomaly detection is typically performed online, i.e., during the RFID system's run-time.
- An anomaly detection sub-system which identifies abnormal behaviors based on the comparison between the Kohonen Map and the processed samples provided by the RFID system.

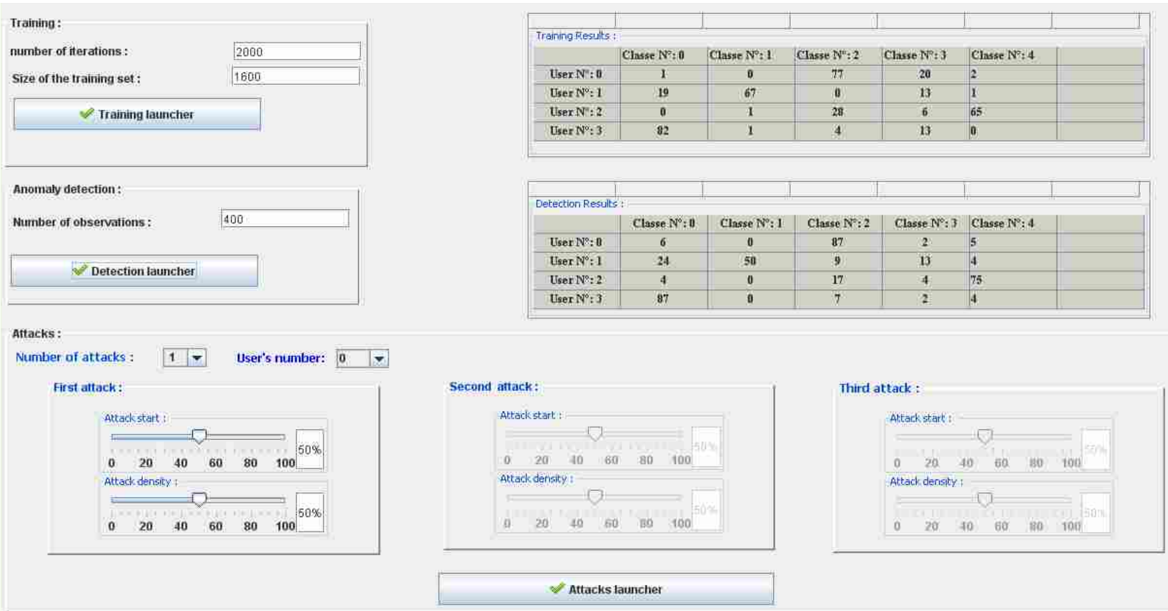


**Figure 4. User Interface**

- *The overall anomaly detection system has*

been developed using Java in conjunction with JVM 1.6, relying on a customized version of the Kohonen engine developed as part as [REN06]. The main goal of this prototype is to prove that the proposed architecture works efficiently on a RFID system. Towards this goal, a simulation of the RFID system was conducted as follows:

- *Step 1* – A self- training-test is generated manually according to the application scenario presented in Section 2. In practice, users' positions are recorded in a XML file.
- *Step 2* – The Kohonen map is trained based on the training record expressed in the XML file,
- *Step 3* – The resulting trained Kohonen map is used in order to detect attack attempts. Towards this goal, a range of anomalies are simulated. In order to facilitate the testing, parameters, e.g., the number of attacks, starting of an attack, degree of density, can be customized through a user interface (Figure 3). In addition, results can be observed using this user interface.

Overall, these experiments were carried on a Windows Dell XPS M 1530, Intel Core 2 duo CPU T5550 1.83GhZ, 2 Gb RAM 987 Mhz with the setting up provided in Table 2. The memory footprint of our anomaly detector can be split into 9270Kb for the training component and 318Kb for the detector whereas 6937ms (respectively 250 ms given a trajectory composed of 400 positions) is devoted to the training (respectively anomaly detection).

| n | m | s | $\sigma_0$ | $\eta_0$ | Measure | $\alpha, \beta$ |
|---|---|---|---|---|---|---|
| 3 | 1600 | 400 | 0.9 | 0.1 | euclidian | 4 |

**Table 2. Configuration parameters used during the experiments**

## 5. Conclusion

In this paper, we propose an anomaly detection system that attempts to find patterns in the data provided by a RFID system, which do not conform to the expected behaviour. For this purpose, we rely on the Kohonen map, a powerful tool for automatically categorising a system activity. In practice, the data provided by the RFID system is first pre-processed in order to train a Kohonen map which permits to define a region representing the normal behaviour of the observed subject. Based on the trained Kohonen map, any activity that does not scope with the defined normal behaviour is identified as an anomaly. The main advantage of this approach is that there is no need for defining the pattern of an intrusion. In addition, such a backend method does not necessitate amending the technical specification of the RFID system. We further developed a prototype of an anomaly detection system which serves as a proof of concept. First experiments show that the time and memory related to the training phase and the anomaly detection together is minimal. We are planning to complement our preliminary simulation-based experiments with real-world tests involving the control access of several computer labs. Such a test bed will permit to obtain real-world traces and their related intrusions and hence constitutes a prerequisite to effectively evaluate the performance of the anomaly detection system in terms of false positive, false negative and number of anomalies effectively detected. We are also thinking in enriching the user profiles with location-, trajectory- and context-related information so as to increase the detection ratio. This enrichment implies extending the core Kohonen engine with novel measures that catch with the heterogeneity of the parameters taken into account in the users' profiles.

# Reference

[ABK10] J. Garcia-Alfaro, M. Barbeau, E. Kranakis. Handling security threats to the RFID system of EPC networks. In Security of self-organizing networks: MANET, WSN, WMN, VANET, Auerbach Publications, Taylor & Francis Group, USA, in press.
[BG05] S. C. Bono, M. Green, A. Stubblefield, A. Juels, A. D. Rubin, M. Szydlo.. Security Analysis of a Cryptographically-Enabled RFID Device. Usenix security symposium. pp 1-16, 2005.
[CBK09] V. Chandola, A. Banerjee, V. Kumar. Anomaly detection: a survey. ACM computing survey, vol. 41(3), 2009.
[DE87] D.E. Denning. An Intrusion-Detection Model. IEEE transactions on software engineering, vol 13(2), pp 222-232, 1987.

[EPC07] EPC Global. Class-1 Generation-2 UHF air interface protocol standard Version 1.0.9: “Gen-2”. http://www.epcglobalinc.org/, 2005.
[EV10] M. Esposito and G.D. Vecchia. An ontology-based intrusion detection for RFID system. Tehnological developments in networking, education and automation, 2010.
[LB91] Lo Z.-P. and Bavarian B., “On the rate of convergence in topology preserving neural networks”, *Biomedical Cybernetic*, vol. 65, pp. 55–63, 1991.
[LMF07] M. Lehtonen, F. Michahelles, F. Fleisch. Probabilistic approach for location-based authentication. Technical Report, Auto-ID labs, http://www.autoidlabs.org/publications/page.html, 2007.
[MH07] L. Mirowski and J. Hartnett. Intrusion detection system model to detect change in tag ownership. International journal of computer science and network security. Vol. 7 (7). July 2007.
[MIL06] M.J. Miller. RFIDSim – a simulator for RFID networks, http://mattewjmiller.net/files/rfidsim_doc/html, 2006.
[ML09] M. Langheinrich. A survey of RFID privacy approaches. Personal and ubiquitous computing, vol.13(6), 2009.
[REN06] Rennard J.P. Réseaux neuronaux, introduction accompagnée d’un modèle Java. Book. Published Vuibert Edition, 2006. http://www.rennard.org/irn/som.html.
[TK82] T. Kohonen. Self-organised formation of topologically correct feature maps. Biological Cybernetics vol 43(1), January 1982.
[TS08] G. Thamilarasu and R. Sridhar. Intrusion Detection in RFID Systems. IEEE Military communication conference (MILCOM), Januray 2008.
F. Kerschbaum and N. Oertel. [Privacy preserving pattern matching for anomaly detection in RFID anti-counterfeiting. 6th workshop on RFID security, 2010
[WS04] S.A. Weis, S. E. Sarma, R.L. Rivest and D. W. Engels. Security and privacy aspects of low-cost radio frequency Identification Systems.. In Security in Pervasive Computing, Vol. 2802, pp. 201-212, 2004.
[YGD10] H. Yang, J. Guo and F. Deng. Collaborative RFID intrusion detection with an artificial immune system. Journal of intelligent information systems, Springer, pp 1—26, 2010.