Application of a Game Theoretic Approach in Smart Sensor Data Trustworthiness Problems

Konstantinos Maraslis, Theodoros Spyridopoulos, George Oikonomou, Theo Tryfonas, Mo Haghighi

Application of a Game Theoretic Approach in Smart Sensor Data Trustworthiness Problems*

Konstantinos Maraslis, Theodoros Spyridopoulos, George Oikonomou, Theo Tryfonas, and Mo Haghighi

Cryptography Group, University of Bristol, UK
k.maraslis@bristol.ac.uk, th.spyridopoulos@bristol.ac.uk,
g.oikonomou@bristol.ac.uk, theo.tryfonas@bristol.ac.uk,
m.haghighi@bristol.ac.uk

Abstract. In this work we present an Intrusion Detection (ID) and an Intrusion Prevention (IP) model for Wireless Sensor Networks (WSNs). The attacker's goal is to compromise the deployment by causing nodes to report faulty sensory information. The defender, who is the WSN's operator, aims to detect the presence of faulty sensor measurements (ID) and to subsequently recover compromised nodes (IP). In order to address the conflicting interests involved, we adopt a Game Theoretic approach that takes into consideration the strategies of both players and we attempt to identify the presence of Nash Equilibria in the two games. The results are then verified in two simulation contexts: Firstly, we evaluate the model in a middleware-based WSN which uses clustering over a bespoke network stack. Subsequently, we test the model in a simulated IPv6-based sensor deployment. According to the findings, the results of both simulation models confirm the results of the theoretic one.

1 Introduction

Wireless Sensor Networks (WSNs) have been playing a major role in the field of monitoring and controlling complex processes remotely. Their application on industry has facilitated the automation of large, complex and distributed industrial control processes. However, the plethora of applications that they can be used for, including their utilisation in sensitive fields such as military and medical applications, or the Critical National Infrastructure (CNI), renders WSNs an attractive target for a variety of cyber-attacks. Therefore, protecting such networks is of topmost importance. In order to maintain their security, WSNs require a set of policies effectively implemented in an automated fashion, so that faster and more efficient security-related decisions can be made.

Much research has been conducted into the security aspect of WSNs in an attempt to address the aforementioned issue [1, 2]. However, due to the significant resource constraints in WSNs' hardware and long unsupervised operations, key challenge in the protection of WSNs is the development of lightweight methods that will be able to efficiently detect and confront attacks under constrained computational resources.

* The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^o 609094. This work has also been supported by Bristol's Systems Centre and Fraser-Nash Consultancy.

In this work, we utilise the principles of Game Theory to develop two discrete models for the protection of WSNs; an Intrusion Detection System (IDS) and an Intrusion Prevention System (IPS). Our models provide optimal cost-efficient strategies for the detection of an intrusion (IDS model) and the protection of the system (IPS model). Additionally, their ability to take into account all related costs (sensor price, cost of recovery, attack cost etc.) along with the ability to apply their results as a security policy in the WSN without constantly updating the network, renders them an energy-efficient protection solution. Our models provide automated procedures based on which, network operators can either disregard bogus data from compromised sensors or find the optimal way to recover the compromised sensors depending on their capabilities.

To validate our findings, we run two sets of simulations. Initially, we simulate a middleware-based WSN that uses network clustering and a bespoke network stack using Sensomax [3], an agent-based WSN middleware, which supports executing multiple applications with regards to their operational paradigms. Subsequently, we use the Cooja simulator, which is distributed as part of the Contiki[1] open source Operating System for the Internet of Things.

The next section offers some basic background knowledge in the area of WSN protection and Section 3 discusses the key related work in this field. Section 4 presents the models for both the IDS and the IPS, including the results of their simulations. The validation of our models in a cluster-based deployment is provided in Section 5, while Section 6 presents a validation of our models utilising an IPv6-based Deployment. Finally, Section 7 provides the conclusions of our work and some paths for further work.

2 Basic Background

Wireless Sensor Networks (WSNs) are typically composed of self-powered sensors (nodes) that communicate with each other and/or with a base station. The topology of a WSN varies from a simple star-shaped network to more advanced multi-hop wireless mesh networks, while the number of sensors that comprise them can fluctuate between different deployments. Sensors have inherent limitations in terms of storage capacity, processing power, energy availability and network bandwidth. Thus, implementing encryption-based security mechanisms is a challenging task [4].

Various threats like eavesdropping, lack of physical protection of the sensors, Denial of Service (DoS) attacks and injection of malicious data necessitate their protection [5]. This is a challenging task, especially when users and/or nodes take autonomous decisions, which in turn raise non-cooperative behaviours and conflicting interests [6].

Conventional WSN security systems mainly use Rule/Signature-based detection mechanisms, which can effectively detect known attacks based on predefined rules, or anomaly-based mechanisms that can detect new attacks by comparing patterns or resource utilization [1]. However, the application of such methods should be driven by security policies that take into account the related resource constrains.

In solving decision making problems of this kind, *Game Theory* seems to be a suitable approach as it suits situations where adversarial interests are included. Every participant can choose an action within a set of predefined actions and for every possible

[1] http://www.contiki-os.org

combination of those there is a reward/utility that occurs for each participant. These situations are called *games*, the participants are called *players* and the actions are called *strategies*. The notion of "solution" that a game theoretic approach can offer can have many forms, most of which require that the players are rational, meaning that they *only* seek the maximisation of their personal reward/utility. In addition, there are many kinds of games, depending generally on the piece of information that is disclosed to the players, their relationship and the type of their communication, if any. In this work we only deal with two-player games where every player's loss is equal to the other's reward (zero-sum game). The concept of game solution adopted here is the detection, at the beginning of the (static) game, of strategies such that every player's chosen strategy is the best response to the other player's chosen strategy. In other words, when both players follow their computed optimal strategies, none of them would be tempted to unilaterally change it, because that would only lead to a reduced individual reward, compared to the existing. The concept described, is known as *Nash Equilibrium* of the game [7–9].

3 Related Work

Game Theory has been used in the past for simulating and solving security-related problems in WSNs. For example, the authors of [2] investigate the case where a clustered WSN is under attack. In this project, the attacker targets the cluster heads in an attempt to crowd the data flow or drop it. The underlying IDS monitors the data transfers and attempts to keep the WSN functioning by detecting malicious nodes in the forward path. This situation is modelled as a two-player, non-cooperative, zero-sum game and it is proved that the game has no pure Nash Equilibria. This means that the game is unstable, and therefore does not provide a state at which we would expect the game to be stabilised after a large number of iterations. In the resource-constraint environment of WSNs this instability is translated into increased power demands.

In [10] the behaviour of a system under a Distributed Denial of Service (DDoS) attack is under investigation based on previous work of [11]. The target environment there is a network, however the model is generic and based on the same networking principles that apply to WSNs. The attacker aims to perform a DDoS attack at a system that has implemented a firewall. The attacker's strategy is defined by the number of occupied nodes and the distribution according to which they transmit malicious traffic. The defender on the other hand can control the settings of the system's firewall. This situation was modelled as a two-player, static, non-cooperative, zero-sum game. The research concludes with suggestions for the strategy of the defender which maximize the minimum payoff of the defender regardless of attacker's decision and behaviour.

Authors in [12] try to improve the security and energy efficiency of a WSN by applying a reputation system on its nodes where low-reputed ones are shut down. Every node can improve its reputation by forwarding incoming packages. However, this forwarding causes draining of their batteries. Since conflicting interests are present, a game theoretic model is adopted in order for the maximum possible battery life of the nodes to be assured while sustaining an unproblematic operation. In addition, there are malicious nodes that can cause package drops, making the proper flow of data even more difficult. On all scenarios of the WSN games of this work, the authors solve the problem by find-

ing the network's Nash Equilibrium. Under the assumption that the involved players are rational, the authors find the optimal strategies for both the defender and the attacker that ensure an upper limit for the expected losses when they are followed. As far as the security and power conservation are concerned, the network improves significantly in all three cases, comparing to the scenario were the model was not applied.

Compared to those works, our is about a different scenario. Additionally, the aforementioned works do not address cases where more than one parameters affect each player's strategy simultaneously. Our IPS and IDS models incorporate this capability making the problem multi-dimensional, which not only adds to its complexity but, most importantly, offers the ability to describe a far wider range of problems. A method similar to the one of this work, can be used for even more parameters. However, the complexity and computational workload will increase considerably.

4 Examining Smart Sensor Data Trustworthiness

In our models, the attacker modifies compromised nodes in order to make them report erroneous values. We make the following assumptions about the deployment: 1) All network traffic is encrypted, 2) all sensor measurements are signed and 3) the deployment's topology is not publicly available, which is a reasonable assumption since the logical network topology (e.g. routing topology or cluster membership) is created and maintained at runtime by an algorithm that relies on criteria which can change over time and are not known a-priori, such as the quality of radio links.

For the attacker, we make the following assumptions: 1) S/he is external to the system and highly motivated. 2) S/he can actively initiate attacks against nodes (the firmware running on nodes is susceptible to bugs already discovered by the attacker and therefore, an attack against a node is always successful). 3) S/he has high time availability, but not enough to break cryptography and signature schemes. 4) Due to the inability to break signatures, the attacker can introduce neither her/his own traffic nor malicious nodes. Therefore, the only option is to compromise existing, legitimate nodes. 5) Due to her/his inability to break cryptography, the attacker can passively overhear traffic but cannot understand the contents of network packets. As such, s/he cannot synthesise the deployment's topology from passive eavesdropping. 6) Lastly, the attacker has high, yet not unlimited financial resources. Thus, s/he can choose to attack the entire network, but the criterion is to optimise the financial benefit of an attack. 7) The attacker can choose the number of nodes to attack, but being oblivious about the network topology has no way of identifying which nodes would maximise damage to the network.

4.1 Modification Detection Model

In this model a game between the defender who wants to monitor a specific area and the attacker who randomly chooses which sensors to attack and tries to make the network transmit as much incorrect information as possible, is deployed. The first question that rises for the defender is what should be the density of sensors (i.e. number of sensors per area unit) that should be chosen, as this affects directly the strategy of the attacker. Since the area under investigation is predefined, it is only the number of sensors that

can affect the density. Hence, the number of sensors is part of defender's strategy and in the game s/he tries to find the most beneficial value within a set of possible choices.

In addition, every sensor has a coefficient of significance. This coefficient is proportional to the level of trust that is related to the information transmitted by this particular sensor and echoes the probability that the measurements provided by the sensor are true. The reasons that this coefficient differs from sensor to sensor vary from the type of measurements that are taken, to the structural features of the sensing elements.

Tolerance is another strategy of the defender and it is a property of the whole network. Having defined untrusted / trusted / total information as the sum of significance coefficients of untrusted / trusted / all sensors respectively, tolerance denotes the minimum portion of the total information that the untrusted information should be, in order for the latter to be believed by the defender. In other words, it denotes the minimum value that the following fraction can have in order for the incorrect information that has been injected into the network to be treated as correct. We call this fraction Attack Coefficient: $AC = Untrusted\ Information/Total\ Information$. This is part of the defender's strategy since s/he is the one to decide which piece of information is treated as valid. The choice of tolerance can directly affect players' tactics due to formula (1).

At this point it is essential that some basic assumptions of the model are presented: 1) Players are rational. 2) Full area coverage is desired. 3) Two sensors of the same network with identical specifications, operating under identical conditions can still report slightly different values. 4) A compromised sensor cannot affect the information that other sensors transmit. 5) The attacker's goal is to make a sensor transmit faulty data that demonstrate noteworthy deviation from the data that uncompromised sensors transmit (otherwise the attack is pointless). 6) Compromised network is the network into which the injected faulty information is believed by the defender.

Under those assumptions, the network operators try to take into account only the non-compromised data without knowing in advance which piece of data is compromised. Therefore, if the attack coefficient is greater than $tolerance$ then the incorrect information is considered to be accurate, correct data is disposed and the attempt for compromising the network is considered successful, which in turn increases attacker's payoff. Otherwise, the network is not considered compromised, which implies a lower payoff for the attacker. Thus, the algorithm and, in turn, the defender can judge whether the network is under attack by the percentage of the believed information out of the total information which justifies its inclusion in the IDM category. Intuitively, tolerance should only be a value greater than 0.5 (50%) and of course less or equal to 1 (100%). In this way, the weighted information that will be ultimately "believed" by the defender will correspond to at least half of the total weight. Our goal is to help the defender choose the best options (i.e. options that will lead to the highest possible payoff) about the number of sensors that will constitute the network and the tolerance adopted.

The attacker can only affect the number of sensors attacked considering that each one of these attacks bears a cost. Therefore, the optimal strategies are not obvious and a game theoretic approach would be suitable. The payoff function (1), with the help of which a payoff matrix will be populated, is affected by the aforementioned parameters.

$$AP = (\frac{is}{ts} \geq t) \times rcn + \ s \times cps \ - \ a \times cpa + \ t \times tc \tag{1}$$

where, AP = Attacker's Payoff, is = incorrect sum (i.e. the sum of significance coefficients of the actually compromised sensors), ts = total sum (i.e. the sum of significance coefficients of all sensors), t = tolerance, rcn = reward for compromising the network, s = number of sensors, cps = cost per sensor, a = attacks, cpa = cost per attack, tc = tolerance cost and:

$$(\frac{is}{ts} \geq t) = \begin{cases} 1 & \text{if inequality holds} \\ 0 & \text{if inequality does not hold} \end{cases} \tag{2}$$

As the formula denotes, the attacker will only be rewarded with rcn if s/he manages to compromise the network ($is/ts \geq t$) which is equivalent to $[(is/ts \geq t) = 1]$, whereas s/he bears the cost of attacks, regardless their impact. Since the players are antagonistic, the attacker takes advantage of the defender's expenses. Thus, everything that has a cost for the defender, like the total cost of sensors $(s \times cps)$ or the total tolerance cost $(t \times tc)$, is added to the attacker's reward in formula 1. The necessity of tolerance cost lies in the fact that the greater the tolerance is, the greater part of the whole information, should be faulty in order for it to be "believed". That motivates the attacker for a more comprehensive attack and therefore a less possible recovery by the operators of the network. Under this perspective, it could be preferable for the network to suffer a mild assault that will compromise the network temporarily, than risk suffering a massive one that will render it totally useless or unaffordable to be fixed. It should be noted that the payoff function has no units of measurement. It is just a necessary quantification of the advantage derived for each player due to the actions taken so that the problem can be solved and resembles the role of a utility function.

It is worth noting that although the defender is not aware of which piece of information is compromised, it is still possible to use the outcome of formula 1. In other words, although the defender cannot distinguish between correct and faulty data, s/he is aware of the payoff that s/he receives when both players choose specific strategies. Furthermore, there is a chance that $(cs/ts) < (is/ts) < t$, where cs = correct sum. In this case, the compromised information will not be believed although it is greater portion of the total information than the correct information is and therefore no reward for compromised network is given to the attacker. This is only possible for $t > 0.5$ (50%).

Since every strategy of the defender consists of a pair (m, n) where m is the number of sensors used and n is the acceptable tolerance, we have two-dimensional strategy sets. One way for this to be tackled and thus for the optimal strategies to be found, is the procedure we outline here. The algorithm is described by the following piece of pseudo-code along with Fig. 1. In this figure, green denotes the parameters that are chosen by the defender and constitute their strategy ($Number\ of\ Sensors$ and $Tolerance$) while orange is used for the parameter that is chosen by the attacker and constitute their strategy ($Attacks$ which is the number of attacks performed). $Sensor\ weights$ are the aforementioned significance coefficients of the sensors which can shape defender's strategy, but their value is not chosen by the defender and therefore it is in grey colour. These parameters shape the values of $Attacker's\ Payoff$ (formula (1)) which populate $Attacker's\ Payoff\ Matrix$ that is seen in Fig. 1. This is the matrix of the game, based on which we will later look for Nash Equilibria. Its pseudo-code is:

for $s = S_{min}$ *to* S_{max}
  $SC(all\ sensors) = 1/follow\ Uniform/follow\ Normal$
  given the strategy sets for *number of attacks* and *tolerance level*
    $-$populate APM_s based on formula(1)
    $-$calculate $ne(APM_s)$ and $AR(ne(APM_s))$
end for
$NE = \{ne(APM_s), \forall\, s\}$
$NEG = \{ne(APM_s) \in NE : AR(ne(APM_s)) = min\{AR(NE)\}\}$
find which strategies lead to NEG

where s is the number of sensors in the network, S_{min} and S_{max} are the minimum and maximum possible number of sensors, respectively, $SC()$ denotes the significant coefficient of deployed sensors, APM_s is the Attacker's Payoff Matrix (Fig. 1) that occurred for $number\ of\ sensors = s$, $ne()$ is the Nash Equilibrium/a of a sub-game, $AR(ne())$ is the attacker's reward that corresponds to $ne()$ and NEG is the Nash Equilibrium/a of the whole game.

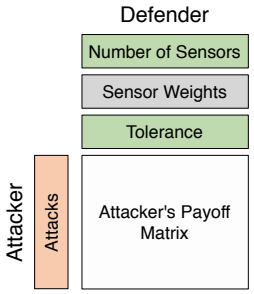

Fig. 1: Schematic description of the IDS model

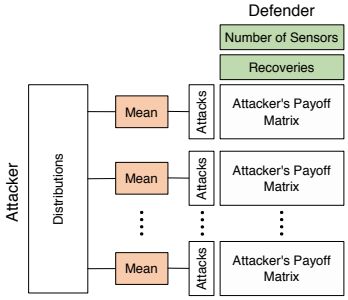

Fig. 2: Schematic description of the IPS model

4.2 Modification Correction Model

In this model's use case there is an attacker who attacks sensors and a defender that protects them, but there are three key differences from the detection instance. Firstly, the defender in this game knows which sensors are compromised and has the ability to recover them. Secondly, there are now two parameters that affect the attacker's strategies, instead of one and thirdly, the game now is repeated for many rounds. However all decisions are made at the beginning and remain unchanged for the whole game, which makes the game static although in a repeated form. Attacker's goal is once more to compromise the network with the least possible cost while defender's is to keep the network uncompromised with the least possible cost. Our scope is to help the defender choose the best options regarding the number of sensors that will constitute the network and the number of recoveries that will be required. Again, the best choices for the attacker are considered the ones that will lead to the highest possible payoff.

The schematic representation of this model is shown in Fig. 2. The logic of the colours is the same as in Fig. 1, therefore the attacker's strategies are defined by *Distributions* (i.e. the distribution that the number of attacks follow) and their *Mean* (i.e. the mean value of the distribution) while defender's strategies are defined by the *Number of Sensors* that constitute the WSN and $Recoveries$ which denotes the maximum number of recoveries performed in each round and remains the same for all rounds. $Attacks$ occur as a result of the choice of $Distributions$ and their $Mean$ values (as in the pseudo-code that follows) and affect formula (3) based on which *Attacker's Payoff Matrix* is populated. We then seek for Nash Equilibria on that matrix. At any round of the game the attacker can only make as many attacks as the uncompromised sensors in the network and the defender can only make as many recoveries as the compromised sensors in the network. The payoff this time is computed as the function:

$$AP = ta \times (rcs - ac) \; + \; tr \times (rcps - rcs) \; + \; s \times sc \; + \; (\frac{cse}{tns} \geq t) \times rcn \quad (3)$$

where, AP = Attacker's Payoff, ta = total attacks, rcs = reward for compromising a sensor, ac = attack cost, tr = total recoveries, $rcps$ = recovery cost per sensor, rcs = reward for compromised sensor, sc = sensor cost, cse = compromised sensor at the end, tns = total number of sensors and for n number of rounds:

$$total\ attacks\ (or\ recoveries) = \sum_{i=1}^{n} attacks\ (or\ recoveries)\ at\ round\ i \quad (4)$$

$$(\frac{cse}{tns} \geq t) = \begin{cases} 1 & \text{if inequality holds} \\ 0 & \text{if inequality does not hold} \end{cases} \quad (5)$$

Again, as a zero-sum game, everything costly for the defender rewards the attacker. For example, an attack has a cost ac for the attacker but as a result a compromised sensor occurs which rewards the attacker (rcs) since it is harmful for the defender. Thus, the total number of attacks (ta) is multiplied by $(rcs - ac)$. The same logic explains the terms $tr \times (rcps - rcs)$ and $s \times sc$ while the term $(cse/tns \geq t) \times rcn$ is equivalent to the term $(is/ts \geq t) \times rcn$ in formula (1). In this case the algorithm may choose to intentionally allow mild attacks since that will save the defender of the recovery costs and will still bear a cost for the attacker although the latter will not be rewarded with the rcn, causing an overall small damage. Additionally, there should be $rcs < ac$ and $rcps < rcs$. The first inequality ensures that the attacker will seek the additional reward for compromising the network (rcn) and that his optimal strategy is not necessarily to attack as many sensors as possible. The second inequality ensures that the defender will not overspend his resources protecting more sensors than necessary. As in the previous model, if $[(cse/tns) \geq t] = 1$, then the network is considered compromised and the corresponding reward is given to the attacker. The pseudo-code for this model is:

```
for s = S_min to S_max
  for every D ∈ Distributions = {Normal, Poisson, Exponential}
    for every m ∈ MeanValues
      −Generate attacks(i), i = 1,...,5 that follow D(m)
      −Let attacks(i), i = 1,...,5 be possible number of attacks
       Given the strategy sets of attacks and recoveries
         −populate APM_s based on formula(3)
         −calculate ne(APM_s) and AR(ne(APM_s))
    end for
  end for
end for
NE = {ne(APM_s), ∀ s}
NEG = {ne(APM_s) ∈ NE : AR(ne(APM_s)) = min{AR(NE)}}
find which strategies lead to NEG
```

where, $MeanValues$ is the set of all possible mean values and is described later on. By $D(m)$ we mean that the variable follows distribution D with mean m. In the case of Normal distribution, there is also variance (σ^2) needed but is omitted from the pseudocode for simplicity. However, it is taken into account in the execution of the real code. That variance remains unchanged through the model and has been chosen in a way such that all the values that are generated and follow $N(m, \sigma^2)$ lie within the defined range. In addition, the procedure of generating attacks has been designed in a way such that $\{attacks\ that\ follow\ D(m_i)\} \cap \{attacks\ that\ follow\ D(m_j)\} = \emptyset$, for $i \neq j, \forall\ D \in Distributions$.

4.3 Model Results

In this section we present our models' simulation results visualized as a threefold graph. The distributions used to describe attacker's behaviour and significance coefficients, are commonly used to describe various elements of network activity [13, 14].

For the IDS model, the sample values that were used for formula (1) are: Sensors: [500, 600], t: [0.55, 0.9], Attacks: [500, 600], Significance coefficients: All equal to 1, follow Uniform(1,4) and Normal(2.5, 0.25), rcn = 10, cpa = 1.2, cps = 2.3, tc = 10. Conclusions can be extracted by Fig. 3. We interpret the figure, bearing in mind that we help defender to take the best possible decision regarding the maximization of his payoff. In Fig. 3, we can see the Nash Equilibria of all the sub-games that occurred. The horizontal axis in all sub-graphs of the figure is the number of Sensors. A Nash Equilibrium can be seen, as a vertical line that goes through all three sub-figures. If (x, y_1), (x, y_2) and (x, y_3) are the points that this line cuts the blue lines of sub-figures 1, 2 and 3 (starting from the upper one) respectively, that means that the best option for the defender would be to deploy x sensors and tolerance equal to y_3 for the WSN. The best response to that for the attacker is to perform y_2 attacks. That strategy would lead to a payoff for the attacker equal to y_1. The pair (x, y_3) represents the best strategy that the defender can choose in order to respond to attacker's y_2 strategy and vice versa. Since every vertical line that goes through all sub-figures is a Nash Equilibrium (for the values of x that the graphs exist), we want the one that leads to the least payoff for the attacker which is represented by y axis in the top sub-figure. Fig. 3 depicts

the outcome of the game for the scenario where all significance coefficients are equal to 1 and from that we see that the the least possible attacker's payoff is 703.3 which is achieved when the defender deploys 511 sensors (x axis) in a WSN with tolerance equal to 0.8 (bottom sub-figure) and the attacker performs 400 attacks (middle sub-figure). Thus, the defender's optimal strategy is $(x, y_3) = (511, 0.8)$ and the optimal strategy for the attacker is $y_2 = 400$. This leads attacker's payoff equal to 703.3.

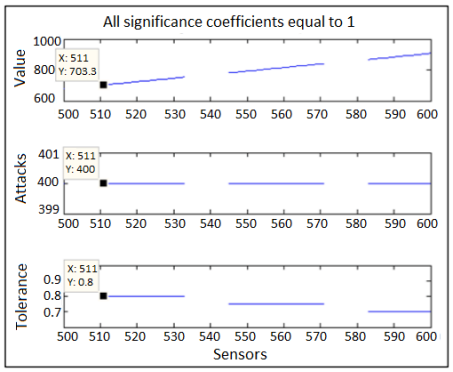

Fig. 3: Attackers Payoff, Number of Attacks and Tolerance for the Nash Equilibrium that occurs for different num. of Sensors when all significance coefficients are equal to 1

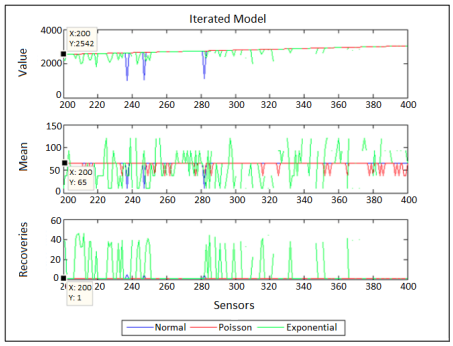

Fig. 4: Attacker's Payoff (Value), Mean values and Number of Recoveries of the Nash Equilibria found in the Iterated model

The gaps of Fig. 3 are observed close to the values that correspond to combinations of strategies that would make the equality $is/ts = t$ from formula (1) to hold. Graphs for the other two scenarios of significance coefficients' distribution, Uniform(1,4) and Normal(2.5, 0.25), are not demonstrated due to their similarity. However, they would be interpreted the same way. The results for all distributions are included in Table 1.

For the IPS model, values for the involved parameters in formula (3) are: Sensors: [200, 400], Recoveries: [1, 70], Distribution of number of attacks: Normal, Poisson, Exponential, rcs = 1.5, ac = 3, Mean values created per distribution = 5, $rcps$ = 5, sc = 4, rcn = 2000, t = 0.5, Attacks: [10, 120]. Interpretation of Fig. 4 is almost identical to the one of Fig. 3. The only difference is that there are now all three distributions in the same figure. Therefore, the Nash Equilibrium of this game will be the one that leads to the minimax price (i.e. the minimum price out of the highest possible ones) of Value. Given that every vertical line that goes through all sub-figures is a Nash Equilibrium of the game and (x, y_1^i), (x, y_2^i), (x, y_3^i), $i \in \{Normal,\ Poisson,\ Exponential\}$ are the 9 points that this line cuts all graphs of all sub-figures then if the defender chooses a specific number of sensors x, the attacker will choose as a response, out of points $\{(x, y_1^i),\ i \in \{Normal, Poisson, Exponential\}\}$ the distribution i for which $max\{y_1^i,\ i \in \{Normal,\ Poisson,\ Exponential\}\}$ is achieved. Thus, assuming that (x, y_1^i) are the points that the vertical line that goes through x cuts all graphs of the first

sub-figure, the defender should choose x for which $min\{max\{ordinate(x, y_1^i)\}\}$ is achieved. The strategies that correspond to the points found that way are Nash Equilibria since they follow the definition of Nash Equilibrium mentioned earlier.

Table 1: Aggregated Results

Intrusion Detection Model			
Significance Coefficients	Optimal # of Sensors	Optimal Tolerance	Optimal # of Attacks
All equal to 1	511	0.8	400
Uniform(1,4)	503	0.85	400
Normal(2.5, 0.25)	500	0.85	400
Intrusion Prevention Model			
Type	Optimal # of Sensors	Optimal # of Recoveries	Optimal # of Attacks
Non-Iterated	200	1	Expon.(mean: 92.5)
Iterated	200	1	Poisson (mean: 65)

5 Validation in a Cluster-based Deployment

In this section we conduct a number of experiments to validate both the IPS and the IDS utilising the clustering facilities offered by Sensomax which allows us to validate our simulation with a hardware-in-the-loop approach. In all our experiments, both models were programmed as two separate applications in every sensor node. Those two applications can be executed concurrently in order to detect and prevent attacks, whilst sensor nodes are carrying out their normal operation and meeting the requirements of their given task. The application itself resides in a single node, known as the cluster-head, where all the top-level executions happen. The IDS and IPS applications (i.e. model logic) are present in every sensor node, whilst being executed only in the cluster-heads.

For the first phase of our experiment a network of 600 virtual nodes was created in SensomaX Companion Simulator (SXCS) [15], incorporating 30 clusters, each containing 20 nodes. As a way of a sensing application, all nodes were programmed to constantly report Temperature readings at 1-second intervals. A second network containing 600 nodes without any clustering mechanism was also created to report false temperature readings. Each experiment reported in this section was repeated 100 times to gain the average values. Fig. 5a demonstrates the average number of attacks required before detection. For a 510-node network, the average number of attacks is 398. This result is on par with the results reported in Fig. 3, given the standard deviation, which covers the 400 attacks reported earlier. Fig. 5b depicts the number of nodes required for the IPS model to operate successfully based on a variable number of attacks. The results reported in this figure are also relatively on par with the results reported in Fig. 4, given the standard deviation around the mean values. The impact on the energy consumption of the network is depicted at Fig.5c.

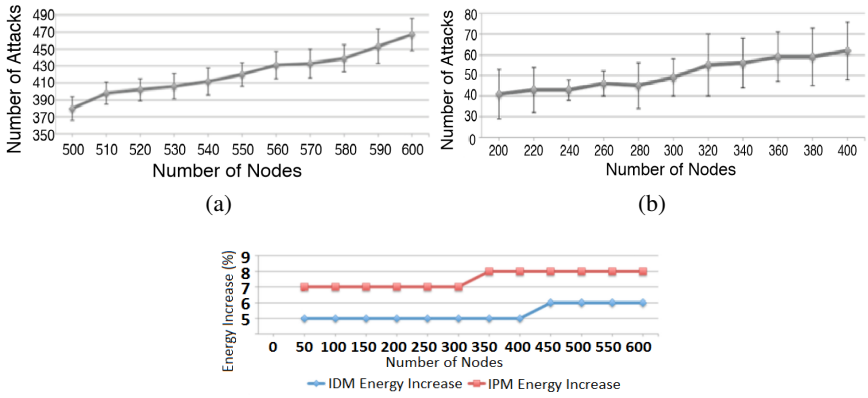

Fig. 5: (a), (b) IDS's & IPS's required number of nodes vs. number of attacks, respectively (c) Impact of IDM & IPM on energy consumption

6 Validation in an IPv6-based Deployment

In this Section we make use of Cooja [16], the network simulator distributed with the Contiki Operating System for the Internet of Things. Within Cooja, we simulate an IPv6-based wireless sensor network. Network nodes use IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) [17] and the Routing Protocol for Low Power and Lossy Networks (RPL) [18]. We simulate a network with 1 traffic sink and 40 traffic sources, distributed in a 200x200 grid. Node distribution is entirely random, with the only limitation being that all sources must have a network path to the sink. We choose to simulate a network of 40 nodes in order to achieve full area coverage, as is the assumption in the model. We use 10 different random topologies and for each topology we repeat the experiment 10 times using a new random seed for each iteration.

In the remainder of the section, we use the following notation: n is the index of a node, $N = \{n : n \in \mathbb{Z}^+ \wedge n \leq 40\}$, $C = \{n : n \in N \wedge \text{node } n \text{ is compromised}\}$, t is the defender's chosen tolerance, $D_n : n \in N$ is the degree of node n discussed below, $S_n : n \in N$ is the significance of node n, also discussed below. In the model, the choice of node significance is based on a random distribution. In our simulations we model node significance as a function of network density. We first calculate the node degree D_n for each each network device, which is calculated as the number of other network nodes within communication range. The significance S_n for node n is subsequently calculated as $S_n = max(\{D_i : i \in N\})/D_n$.

Thus, S_n corresponds to the maximum node degree observed in the network, divided by the node's own degree. Since, all nodes in the network have a path to the sink, they have at least one other node within communication range. Hence, $D_n > 0$ and the significance calculation's denominator is always non-zero. This way, nodes in dense areas will have lower significance, while nodes in sparse areas will have a high one. That

is because the network is used to gather sensory information about an environmental parameter in a geographical region. Even between two identical devices, measurements are likely to be slightly different due to manufacturing inaccuracies and slight fluctuations of environmental parameters even within the same area. Thus, in an area where multiple nodes are reporting, each node's measurement will be of lower significance, whereas in a sparse area where only a few nodes are reporting, it will bear more weight.

According to the model, the optimal attacker strategy is to compromise 78.27% of the total number of nodes in the network (400 out of 511). With this in mind, in each experiment the attacker compromises a random set of 31 nodes ($\mid C \mid = 31$). Furthermore, defender's optimal strategy is to select tolerance level $T = 0.85$. An attack is successful if the defender believes the erroneous value to be accurate and this is only true if Attack's Coefficient $(AC) = \sum_{j \in C} S_j / \sum_{i \in N} S_i > T$.

Fig. 6 illustrates the densities of the ten network deployments under investigation. For all deployments, the minimum node degree D_n was between 1 and 3, whereas maximum node degree was between 7 (topology 1) and 13 (topologies 3 and 5).

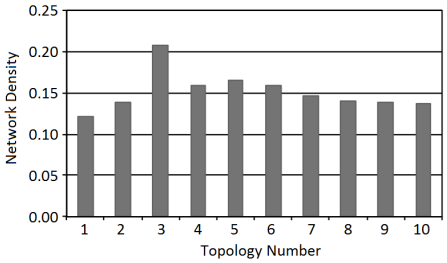

Fig. 6: Topology densities

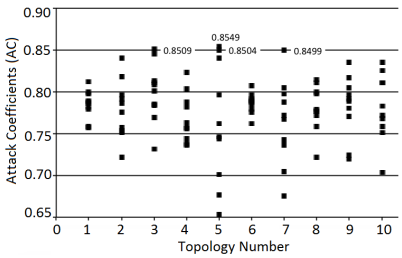

Fig. 7: Attack Coefficients per experiment

Fig. 7 illustrates attack coefficients for each iteration. Across the entire experiment set the attacker was successful only three times. For all other iterations detection was possible. The three successful attacks were observed in topologies 3 and 5, i.e. the ones with the highest network density. This suggests there may be a relation between the model's accuracy and the network density. We shall investigate this in the future.

7 Conclusions

In this paper we show how Game Theory can be used to detect and prevent intrusions in WSNs. These models are applicable to a wide range of use cases, including applications of the Internet of Things, smart metering etc. We demonstrated the effectiveness of the models by two methods of validation. Firstly, with Sensomax where its results matched the ones of the analytical models. Secondly, by using Cooja we investigated the effectiveness of the detection model in an IPv6-connected network of smart objects.

As future work, we aim to extend the model to include quantitative estimation (forecasting), which could be applied on the results of the iterated game with multiple rounds. By fixating the parameters and running the aforementioned game for many

different numbers of rounds, one could apply forecasting methods in order to make an approximation of a player's payoff, given the number of iterations. Additionally, we aim to conduct further validation of the prevention model and investigate its applicability in networks of varying densities as well as its scalability with increasing network size.

References

1. Alrajeh, N.A., Khan, S., Shams, B.: Intrusion detection systems in wireless sensor networks: a review. International Journal of Distributed Sensor Networks **2013** (2013)
2. Reddy, Y.B.: A game theory approach to detect malicious nodes in wireless sensor networks. In: Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on, IEEE (2009) 462–468
3. Haghighi, M., Cliff, D.: Sensomax: An agent-based middleware for decentralized dynamic data-gathering in wireless sensor networks. In: Collaboration Technologies and Systems (CTS), 2013 International Conference on. (May 2013) 107–114
4. Ilia, P., Oikonomou, G., Tryfonas, T. In: Cryptographic Key Exchange in IPv6-Based Low Power, Lossy Networks. Volume 7886 of Lecture Notes in Computer Science. (2013) 34–49
5. Walters, J.P., Liang, Z., Shi, W., Chaudhary, V.: Wireless sensor network security: A survey, in book chapter of security. In: in Distributed, Grid, and Pervasive Computing, Yang Xiao (Eds, CRC Press (2007) 0–849
6. Omic, J., Orda, A., Van Mieghem, P.: Protecting against network infections: A game theoretic perspective. In: INFOCOM 2009, IEEE, IEEE (2009) 1485–1493
7. Spyridopoulos, T., Oikonomou, G., Tryfonas, T., Ge, M.: Game theoretic approach for cost-benefit analysis of malware proliferation prevention. In: Security and Privacy Protection in Information Processing Systems. Springer (2013) 28–41
8. Tambe, M., An, B.: Game theory for security: A real-world challenge problem for multiagent systems and beyond. In: AAAI Spring Symposium: Game Theory for Security, Sustainability, and Health. (2012)
9. Roy, S., Ellis, C., Shiva, S., Dasgupta, D., Shandilya, V., Wu, Q.: A survey of game theory as applied to network security. In: System Sciences (HICSS), 2010 43rd Hawaii International Conference on, IEEE (2010) 1–10
10. Spyridopoulos, T., Karanikas, G., Tryfonas, T., Oikonomou, G.: A game theoretic defence framework against dos/ddos cyber attacks. Computers & Security **38** (2013) 39–50
11. Wu, Q., Shiva, S., Roy, S., Ellis, C., Datla, V.: On modeling and simulation of game theory-based defense mechanisms against dos and ddos attacks. In: Proceedings of the 2010 spring simulation multiconference, Society for Computer Simulation International (2010) 159
12. Asadi, M., Zimmerman, C., Agah, A.: A game-theoretic approach to security and power conservation in wireless sensor networks. IJ Network Security **15**(1) (2013) 50–58
13. Chandrasekaran, B.: Survey of network traffic models. Waschington University in St. Louis CSE **567** (2009)
14. Bedi, H.S., Roy, S., Shiva, S.: Game theory-based defense mechanisms against ddos attacks on tcp/tcp-friendly flows. In: Computational Intelligence in Cyber Security (CICS), 2011 IEEE Symposium on, IEEE (2011) 129–136
15. Haghighi, M.: An agent-based multi-model tool for simulating multiple concurrent applications in wsns. In: Journal of Advances in Computer Networks (JACN), 5th International Conference on Communication Software and Networks. (2013)
16. Österlind, F.: A sensor network simulator for the contiki os. SICS Research Report (2006)
17. Montenegro, G., Kushalnagar, N., Hui, J., Culler, D.: Transmission of ipv6 packets over ieee 802.15. 4 networks. Internet proposed standard RFC **4944** (2007)
18. Winter, T.: Rpl: Ipv6 routing protocol for low-power and lossy networks. (2012)