Unity is strength! Combining Attestation and Measurements Inspection to handle Malicious Data Injections in WSNs

Vittorio P. Illiano*
Department of Computing
Imperial College London
London, U.K.
v.illiano13@imperial.ac.uk

Rodrigo V. Steiner*
Department of Computing
Imperial College London
London, U.K.
r.v.steiner@imperial.ac.uk

Emil C. Lupu
Department of Computing
Imperial College London
London, U.K.
e.c.lupu@imperial.ac.uk

ABSTRACT

Attestation and measurements inspection are different but complementary approaches towards the same goal: ascertaining the integrity of sensor nodes in wireless sensor networks. In this paper we compare the benefits and drawbacks of both techniques and seek to determine how to best combine them. However, our study shows that no single solution exists, as each choice introduces changes in the measurements collection process, affects the attestation protocol, and gives a different balance between the high detection rate of attestation and the low power overhead of measurements inspection. Therefore, we propose three strategies that combine measurements inspection and attestation in different ways, and a way to choose between them based on the requirements of different applications. We analyse their performance both analytically and in a simulator. The results show that the combined strategies can achieve a detection rate close to attestation, in the range 96-99%, whilst keeping a power overhead close to measurements inspection, in the range 1-10%.

ACM Reference format:

Vittorio P. Illiano, Rodrigo V. Steiner, and Emil C. Lupu. 2017. Unity is strength! Combining Attestation and Measurements Inspection to handle Malicious Data Injections in WSNs. In *Proceedings of WiSec '17, Boston, MA, USA, July 18-20, 2017*, 11 pages. DOI: 10.1145/3098243.3098249

1 INTRODUCTION

The current state of the art in Wireless Sensor Networks (WSNs) includes a substantial number of security mechanisms [5, 25, 26]. Nevertheless, an adversary can still compromise sensor nodes by exploiting software vulnerabilities or through physical attacks. Compromised sensors pose a severe threat to data integrity as they can report arbitrary data instead of real measurements. Such threat also propagates to IoT applications that build on top of compromised WSNs, such as emergency management, health monitoring, and vehicle control. As the system's behaviour relies on the measurements data, integrity violations can result in wrong analyses and erroneous decisions which can cause severe damage, and even loss

of life. For instance, compromised nodes could silence the presence of fires preventing an immediate reaction.

Given the large scale such networks can achieve, it is inefficient and impractical to monitor them through external surveillance and manual inspection. Therefore, the health of a WSN should be measured from within itself. This can be done by taking advantage of the sensors' capabilities as most of the time they are idle waiting for the transmission or reception of data. However, this choice has an unavoidable overhead, and further effort is required to keep it low. In the end, there is a strict trade-off between the security level achieved and the cost to obtain it. Current solutions are highly biased towards one of the two.

In this paper, we combine two approaches at different ends of the security/cost spectrum: *Measurements inspection* and *Attestation*. Measurements inspection detects malicious measurements by inspecting the measurements themselves to find changes in their internal correlation structures. When performed in a centralized way, for instance by the base station, this approach requires no additional computations from the sensor nodes themselves. From this perspective, it can be considered very lightweight. However, the accuracy in distinguishing genuine from compromised nodes is limited by the unpredictability of the sensed phenomenon, which introduces uncertainty in the measurements correlations. Attestation, on the other hand, ascertains the integrity of a node by verifying its memory contents through a challenge-response protocol. This approach has a considerable overhead on sensor nodes, which have to exchange additional messages and for a period cannot do anything other than calculating a computationally intensive response. Nonetheless, attestation can be very reliable if the challenge can only be responded in time by the genuine nodes.

Although compromised nodes may perform attacks to undermine the network confidentiality and availability, which could be detected by attestation but not by measurements inspection, it is in attacks to data integrity that both schemes overlap. Moreover, while attestation is good at telling if a node has been compromised or not, it has no saying in which nodes should be attested, whereas measurements inspection is good at pointing the finger to suspicious nodes but less effective in determining maliciousness. In this sense, the techniques complete each other. In real applications, we envision the coexistence of both mechanisms and based on a comparative study, which to our knowledge has not been done before, we propose novel methods to maximise the benefits of their complementary capabilities. Our goal is to combine them while keeping the high security level of attestation and the low cost of measurements inspection. Nevertheless, there are many ways to integrate the two approaches, and it is not obvious which combinations are

*These authors contributed equally to this work.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

WiSec '17, Boston, MA, USA

© 2017 Copyright held by the owner/author(s). 978-1-4503-5084-6/17/07...\$15.00
DOI: 10.1145/3098243.3098249

better beforehand. The main contributions of this paper are: (I) to present different ways of combining these two integrity verification mechanisms; and (II) a performance evaluation of each combination both analytically, at the techniques' components abstraction level, and numerically, through in-depth simulations.

The remaining of this paper is organized as follows. Section 2 gives a comparison with related work. Section 3 defines the attacker model and summarises the characteristics of measurements inspection and attestation along with their performance. Section 4 compares advantages and disadvantages of the two approaches. Section 5 presents different combination strategies and their effect on the detection of malicious sensor nodes. Section 6 shows the detection performance of each combination scheme in terms of accuracy and attestation frequency, which is the main factor to affect the energy consumption. Section 7 presents simulation experiments that model the whole data generation and transmission process in the WSN to obtain an accurate detection performance and energy consumption. Finally, in Section 8, we give our conclusions and possible directions for future work.

2 RELATED WORK

The vast majority of works in the literature regarding attestation focus on the safety of the integrity verification [3, 7, 14, 17, 19, 21, 24]. How often should attestation be performed, which nodes should be attested, and the impacts of performing it, especially in terms of energy consumption, are not well covered [22].

Chen *et al.* [9] investigate how often attestation should be triggered in order to optimize the network lifetime without degrading the detection rate of compromised nodes. They conclude higher compromise rates require higher attestation frequencies. However, it is difficult to know how fast an adversary can compromise network nodes. When attestation is used to ascertain the integrity of the measurements, the attestation frequency should be even higher, since there is a time lapse between the time a sensor node is attested and the time when the measurement is taken, in which the sensor node could potentially be compromised. This problem has not been addressed yet, and we solve it through the combination with measurements inspection, which decreases the attestation frequency by two orders of magnitude, regardless of the compromise rate.

With the ever increasing number of devices permeating our daily lives, scalability also becomes an issue. Asokan *et al.* [4], Ambrosin *et al.* [2], and Carpent *et al.* [6] examine how to scale attestation to efficiently verify a large number of devices. Our combination scheme allows us to target the problem from a different perspective by attesting only a subset of devices, thus incurring less overhead on nodes, while still achieving a good detection rate.

The literature about measurements inspection is broad, but usually considers malicious interference analogous to genuine faults [11]. Instead, the adversary may seek to stay undetected. This is analysed only in a few works [8, 12, 18, 23]. The overhead in computations and communications introduced by such techniques is always kept low [11]. However, measurements inspection techniques have significantly poorer detection performance when several malicious sensor nodes collude in the injection of malicious data. For this reason, Tanachaiwiwat and Helmy [23] propose to deploy tamper-resistant sensor nodes that authenticate suspicious sensors. This

paper, is concerned with integrity problems rather than authentication, and detects malicious activity in highly compromised networks by assuming that only the data sink is tamper-resistant.

To the best of our knowledge, no work in the literature presents a direct comparison between attestation and measurements inspection in WSNs, let alone their combination. This paper analyses and compares both approaches in detail, focusing on the aspects that make them complementary.

3 BACKGROUND

In this section, we describe the characteristics and performance of measurements inspection and attestation and discuss our assumptions and attacker model. Table 1 summarizes the notation used.

Table 1: Notation Summary.

Term	Description
N	Number of sensor nodes
C	Number of malicious sensor nodes
A_{DT}	Anomaly Detection triggers
A_{CF}^S	Sensor S fails Anomaly-based Characterisation
A_{DS}^M	Anomaly-based Diagnosis output for sensor S is "Malicious"
MI_F^S	Sensor S fails Measurements Inspection
A_{TF}^S	Sensor S fails attestation
A_{TP}^S	Sensor S passes attestation
S_G	Sensor S is genuine
S_M	Sensor S is malicious
T_E	Total number of examined sensors
T_M	Total number of malicious sensors
T_{MF}	Total number of malicious sensors that fail a given test
TPR	True Positive Rate
FPR	False Positive Rate

3.1 Measurements Inspection

Measurements inspection refers to the detection of malicious data injections through analysis of the measurements themselves. The idea is that to disrupt the information gathered from the measurements, the injected and genuine values should not hold some internal data properties introduced by *inter-measurements correlation*.

Correlations occur in time, space, and between different sensed phenomena. We focus on spatial correlations, defined as the relationships existing between measurements at different points in space because in this context remaining undetected requires compromising more sensors. Hence, the first step of measurements inspection is a test for anomalies in the measurements correlation, i.e. their *detection*. However, if malicious nodes *collude*, i.e. act in concert according to a common goal, malicious measurements may be correlated and more difficult to detect. Moreover, even if the presence of malicious measurements can be detected, colluding nodes could make genuine sensors appear responsible for it. Thus, besides detection of malicious interference, the system also requires a *characterisation* step, to identify the sensors responsible for it. Finally, faulty sensors may also disrupt inter-measurements correlations and need to be distinguished from malicious ones. To make such distinction, a further step, known as *diagnosis*, is required. Nevertheless, this step is missing in most measurements inspection

algorithms [11]. Figure 1 illustrates the high-level architecture of a complete measurements inspection scheme. In this paper, we use the scheme described in [12], which provides detection of sophisticated collusion attacks and includes also a characterisation and a diagnosis procedure.

Figure 1: Measurements Inspection Scheme.

The *detection* step searches for anomalies, identified as measurements that introduce an anomalous variations compared to the measurements of other sensors. In particular, the anomaly is examined in the individual variation that a measurement introduces in a small neighbourhood, contextualised into the overall variation that is observable in larger neighbourhoods. So, the detection step does not impose any constrain to the reported values but imposes a threshold on the variations at the low scale with respect to the higher scales events. Thus, remaining undetected (i.e., below the detection threshold) prevents the attacker from causing high damage (e.g. spoof a fire) and vice versa. The characteristics of the physical phenomenon, the deployment, and the environment, determine the cross-scale relationships, which are learnt and tested through the analysis of *wavelet* coefficients at multiple scales [12].

Characterisation identifies malicious sensors through a group-wise analysis that considers correlated measurements together, where correlations are still based on their effect on the neighbouring measurements. For instance, if a group of measurements is consistently showing an increasing trend, they will be assigned to the same group. Consequently, colluding sensors are presumably assigned to the same group, while genuine correlated sensors are assigned to different groups. Sensors are grouped together if they are within the area characterised by the same spatial behaviour. The anomalies observed in the detection step are then used to identify conflicts between groups, which reveal which group is responsible for the anomaly. A conflict solving algorithm finds the anomalous groups. The last characterisation task is to filter out sensors in the same group that have a borderline behaviour, i.e. their measurements neither endorse nor reject the group's spatial behaviour.

Finally, by searching for characteristics typical of faulty sensors in the anomalous measurements, the *diagnosis* step can infer whether each anomalous sensor is likely faulty or malicious.

Detection, characterisation, and diagnosis are different tasks, where the probability of succeeding in one of them depends on the success of the previous steps. Success is determined by two factors: identifying as malicious only malicious scenarios (true positives) and not genuine scenarios (false positives).

Moreover, the performance of detection is different from the performance of characterisation and diagnosis. Indeed, detection operates at the granularity of the network, or of a cluster of nodes, while characterisation and diagnosis operate at the granularity of sensors. For this reason, we can link the performance of detection to the random variable C , representing the number of malicious nodes in the network. Instead, the performance of characterisation and diagnosis are linked to the random variable S_M , representing that a generic sensor S is malicious (or dually S_G , the event that a generic sensor S is genuine, where $P(S_G) = 1 - P(S_M)$).

Considering the whole measurements inspection process, a true positive is a malicious sensor that is diagnosed as malicious and characterised as anomalous, after anomaly detection triggered. A false positive, instead, is a genuine sensor diagnosed as malicious, after being characterised as anomalous. This can occur, e.g., because the anomaly is falsely detected or the characterisation falsely blames that sensor. Considering the three steps separately, the probability of having a true positive from measurements inspection is:

$$P(MI_F^S | S_M) = P(A_{D_T} A_{C_F}^S A_{D_M}^S | S_M) = P(A_{D_M}^S | A_{C_F}^S A_{D_T} S_M) P(A_{C_F}^S | A_{D_T} S_M) P(A_{D_T} | C > 0) \quad (1)$$

Where MI_F^S is the event that sensor S fails measurements inspection, A_{D_T} denotes the event that the anomaly detection algorithm triggers, $A_{C_F}^S$ the event that characterisation fails, and $A_{D_M}^S$ the events that the sensor is diagnosed as malicious.

Whereas, the probability of a false positive is:

$$P(MI_F^S | S_G) = P(A_{D_T} A_{C_F}^S A_{D_M}^S | S_G) = P(A_{D_M}^S | A_{C_F}^S A_{D_T} S_G) P(A_{C_F}^S | A_{D_T} S_G) \quad (2)$$

$$\left(P(C = 0) P(A_{D_T} | C = 0 \mathbb{S}_{\mathbb{R}}) + P(C > 0) P(A_{D_T} | C > 0 \mathbb{S}_{\mathbb{R}}) \right)$$

In the absence of a probabilistic model for both approaches, the probabilities can be approximated with experimental frequencies. In particular, the probabilities listed above can be characterised with True Positive Rates (TPR) and False Positive Rates (FPR). Indeed, the frequency of $(MI_F^S | S_M)$ and $(MI_F^S | S_G)$ correspond to the measurements inspection TPR and FPR, which we refer to as TPR_{MI} and FPR_{MI} . We give a proof for the former, while an analogous proof holds for the latter. Let T_E denote the total number of examined sensors, T_M denote the total number of malicious sensors, and T_{M_F} denote the total number of malicious sensors that fail measurements inspection, then:

$$P(MI_F^S | S_M) = \frac{P(MI_F^S \cap S_M)}{P(S_M)} \approx \frac{\frac{T_{M_F}}{T_E}}{\frac{T_M}{T_E}} = \frac{T_{M_F}}{T_M} = TPR_{MI} \quad (3)$$

With a similar reasoning the $P(MI_F^S | S_G)$ can be approximated with the measurements inspection FPR (FPR_{MI}).

3.2 Software-based Attestation

Attestation is a well-studied integrity verification mechanism that allows a trusted device, named *verifier*, to validate the memory contents of an untrusted device, called *prover* [22]. Software-based attestation mechanisms are capable of attesting untrusted devices without the use of tamper-resistant hardware or hardware that restricts an adversary's control over the prover, such as ROM or a Memory Protection Unit.

Attestation follows a challenge-response protocol as illustrated in Figure 2. The process starts with the verifier generating a challenge, which is essentially a random number, and sending it to the prover. The prover then uses the challenge to traverse its own memory in a pseudo-random fashion computing a checksum of the memory addresses accessed and sends the result to the verifier. It is assumed that the verifier knows in advance the expected memory contents of the prover, so it can validate the response. However, in addition to returning the correct response, the prover has to do so within a time limit after the challenge has been sent. This time limit is imposed to prevent an adversary from performing additional operations to masquerade any possible modifications it may have done to the prover's original memory contents [19, 21].

Figure 2: Attestation overview.

In practice, in addition to the time taken to execute the attestation routine, the timeout also has to incorporate the time necessary to send the challenge and receive the response. While it is possible to specify a time limit for sending the packets over the network, it is not possible to make this time constant. Therefore, an adversary could attempt to reduce its network delay, earning time to execute more operations. In theory, the best adversary has zero delay, meaning its time advantage would be the maximum time allowed to send and receive messages over the network. To prevent this from happening, the attestation routine has to be implemented in a time-optimal way. The routine comprises a main loop, where it handles the memory accesses, which must be executed a number of times. To hide any modifications, an adversary would need to insert new instructions into the loop, introducing a fixed overhead per iteration. Consequently, the number of iterations is defined according to the maximum network Round-Trip Time (RTT) allowed, such that the adversary's overhead is greater than its advantage. Let c be the prover's CPU clock speed, a be the adversary time advantage, and o be the adversary overhead per iteration of the attestation routine, then the number of iterations i can be calculated as [20]:

$$i \geq \frac{(c \cdot a)}{o} \geq m \ln(m) \quad (4)$$

Where m is the size of the memory being attested. Since the memory is traversed in a pseudo-random fashion, a minimum number of iterations is necessary to probabilistically assure (based on the Coupon Collector's Problem [15]) that each memory address is accessed at least once.

Under the assumptions made by software-based attestation mechanisms, that both the maximum network RTT and the minimum adversary overhead per iteration are known a priori, it is possible to define the number of iterations of the attestation routines to be executed such that a malicious sensor S_M has only a negligible chance ϵ to pass attestation. This chance is the possibility of a collision — when the genuine and malicious memory contents output the same checksum value. Thus:

$$P(A_{T_F^S} | S_M) \geq 1 - \epsilon \quad (5)$$

We can characterise the attestation TPR, which we refer to as TPR_{AT} , by analysing the frequency of the event $(A_{T_F^S} | S_M)$. Let T_E denote the total number of examined sensors, T_M denote the total number of malicious sensors, and T_{M_F} denote the total number of malicious sensors that fail attestation, then:

$$P(A_{T_F^S} | S_M) = \frac{P(A_{T_F^S} \cap S_M)}{P(S_M)} \approx \frac{\frac{T_{M_F}}{T_E}}{\frac{T_M}{T_E}} = \frac{T_{M_F}}{T_M} = TPR_{AT} \quad (6)$$

Whereas, a genuine node will always pass attestation:

$$P(A_{T_F^S} | S_G) = 0 \quad (7)$$

By definition $FPR_{AT} = 0$. In practice, $FPR_{AT} \approx 0$, because of unpredictable faults in the network channel.

3.3 Assumptions and Attacker Model

In this paper, we focus on malicious data injection attacks conveyed through malicious software run by compromised nodes. In particular, we consider a sophisticated adversary, whose goal is to evoke fake events or conceal real ones while remaining undetected. We do not seek to address other aspects such as securing the communication between nodes or network membership management. Many existing techniques deal with such issues [25]. Similarly, whilst attestation may be used to detect other threats (e.g., to confidentiality and availability), we focus here on integrity alone.

We assume the adversary is in control of a subset of sensor nodes. The premise that some nodes are not compromised is a consequence of the cost of compromising them. In practice this usually holds given that sensors are not identical (e.g., different generations) even when they monitor the same phenomenon; some sensors may be physically hard to reach while others are easier; and some sensors will be subject to maintenance (e.g. because of fouling). A scenario where all nodes are compromised is, therefore, extreme.

Compromised nodes are assumed to run malicious software. Such malware can: (i) introduce a mismatch between the measurements observed by the sensor and the ones reported, which are referred to as *malicious measurements*, (ii) make the compromised sensors inject malicious measurements according to a common strategy, (iii) overhear the measurements sent by genuine nodes, and (iv) adapt malicious measurements in function of genuine ones to reduce the risk of introducing detectable anomalies.

While we do not restrict how an adversary may compromise a nodes' software we assume it cannot modify its hardware. This is a restriction imposed by the software-based attestation threat model. Measurements inspection is capable of identifying sensor nodes with compromised hardware, as long as a sufficient subset of sensors keeps its hardware unchanged, which is generally the case since physical attacks do not scale. However, in this paper, we

focus on the cooperation between both schemes and leave such special cases for future work.

We also assume the WSN sink cannot be compromised, for instance being equipped with tamper-proof hardware, as it plays the role of the verifier. Fundamentally, if the sink gets compromised, then the whole WSN is compromised. We further assume that provers are within direct communication range and can only communicate with the verifier during attestation. This prevents a prover from relaying the challenge to another, more powerful, device to run attestation on its behalf. In practice multiple tamper-resistant verifiers can be distributed across the network.

Finally, we assume the network maximum RTT can be known a priori and that an authentication system is in place. All of these assumptions are requirements well-known in literature for the application of software-based attestation [22], and while we could relax them, by using hardware-based or hybrid attestation, that would mean equipping each and every sensor in the network with the specific hardware demanded by such techniques.

4 COMPARISON OF THE TWO APPROACHES

Attestation and measurements inspection are very different in nature. Besides the difference in performance, they also vary significantly with respect to constraints introduced, test frequency, and power overhead. We analyse these aspects below.

Constraints Introduced. Measurements inspection introduces the need for a trusted *inspector*, i.e. an entity which can run the anomaly detection algorithm and be trusted for its output. Moreover, good performance is achieved when data from many sensors is available. Therefore, it is often cheaper to make a centralized device in charge of running the algorithm, such as the base station or a tamper-proof local coordinator node. Similarly, software-based attestation introduces the need for a trusted *verifier* to attest untrusted nodes. Furthermore, attestation requires the network maximum RTT to be a known constant and works under the assumptions that the adversary cannot modify the prover's hardware.

Test Frequency. Measurements inspection ascertains the measurements integrity at a specific time. If the application layer aggregates multiple time samples, then it is convenient to run anomaly detection on such aggregates. This results in a better aimed protection of the application task and in a reduced detection frequency. Anomaly detection can be run on aggregates even when the measurements of each time instant are used separately, provided a mechanism that extends the validity of a detection check to future samples. Software attestation has analogous requirements. Measurements produced during the time of a successful attestation are genuine. However, the sensor node may misbehave before or after the verification is complete. The larger the time gap between attestation and measurements transmission, the smaller the reliability.

Power Overhead. When anomaly detection is run by the sink, there is no communication overhead. On the contrary, such entity is subject to a computational overhead that is $O(N \log N)$ in the number of measurements [12]. Software attestation instead, introduces a communication overhead, due to the attestation protocol, as well as a computational overhead to the nodes, which cannot do anything other than compute a response while under attestation, and to the verifier, which needs to validate every prover's response.

This step is deliberately computational intensive, otherwise a malicious node would be able to forge a valid response in time. Both in computation and communication, the overhead of attestation is noticeably higher than measurements inspection.

4.1 Trade-off Tuning

Highly Reliable Software Attestation. Software attestation is mostly reliable when the Attestation Frequency (AF) is high. Each individual sensor has its own attestation frequency, defined as the number of attestation challenges received divided by the number of measurements sent by the same sensor. In the following, we refer to AF as the attestation frequency averaged across all nodes. For instance, if only one sensor is attested each time it sends a measurement, then $AF = 1/N$. Assuming all sensors are equally important, they should all be attested with the same frequency. In such case, the reliability of attestation is high when $AF \approx 1$, i.e. when an attestation response is sent together with each measurement report. However, with such choice, the communication overhead is three times as high on average (because of the challenge and response messages). A noticeable computational overhead is also present for the calculation of the attestation response, which in this scenario becomes the main task of the sensor nodes processors. In conclusion, using $AF \approx 1$ reduces the network lifetime, requiring a high maintenance cost either to replace nodes battery or to insert new nodes.

Low-Power Software Attestation. The cost of software-based attestation can be reduced by decreasing AF . This can be done by either reducing the time between two attestations or the number of attested sensors. However, the reliability of attestation deteriorates when the time between two attestations, i.e. $\frac{T}{AF}$, is close to the time needed to swap the genuine software with a malicious one, where T is the time between two measurement transmissions. A possible way to reduce the power of software-based attestation, is then to reduce the number of attested sensor nodes. This can be done by selecting a random subset of nodes to attest which is small compared to the total N nodes. Similarly to the detection step in measurements inspection, when attestation fails for at least one sensor, a more thorough analysis can be triggered, i.e. attesting all other nodes. Such "attestation-based detection" is reliable only when the number of attested nodes is comparable to $N - C$, i.e. the number of genuine sensors. Therefore, unless almost all sensor nodes are compromised, the power overhead cannot be reduced significantly without a significant loss in reliability.

Low-Power Measurements Inspection. Measurements inspection makes a reliable measurements integrity check during the detection step: when the false information introduced by malicious sensors is far from reality and there are genuine sensors whose correlation with malicious sensors is disrupted, detection unveils the presence of malicious data. This is generally the case when there are at least G genuine sensors, where G depends on the WSN deployment, on the monitored physical phenomenon, and on the kind of malicious measurements. Measurements inspection is also able to identify malicious sensor nodes in the characterisation step. However, for this step to be as successful as detection, a higher value for G is required. To keep power consumption at a minimum, measurements inspection needs to accept the presence of potentially malicious sensors and identify only the most likely compromised. Thereafter,

if the remaining malicious nodes keep injecting malicious data, the attack becomes less efficient, and they need either to make the false measurements closer to reality or to become more detectable.

Highly Reliable Measurements Inspection. To increase the number of discovered malicious nodes, measurements inspection needs to rely more on detection and less on characterisation. For instance, characterisation can produce just a set of mutually exclusive hypotheses for the detected anomaly with the correspondent malicious sensors. Further investigations, conducted in a reliable way, would then reveal which hypothesis is correct and which sensors are malicious. This can be done by checking sensor nodes in the field, but on the other hand it is against the goal of reducing the maintenance cost and measuring the WSN health from within itself.

In conclusion, attestation can ascertain the measurements' integrity, provided that it is run close to their transmission time. Running attestation for all measurements is expensive, but running it for an arbitrary subset decreases reliability. Measurements inspection can detect the presence of malicious measurements and identify suspicious sensors, but the final decision about a sensor's maliciousness should be taken with a more reliable approach. The two techniques complement each other as measurements inspection can trigger attestation in anomalous scenarios only, reducing the attestation frequency and keeping reliability high.

5 COMBINATION STRATEGIES

The objective of the combination is to achieve a trade-off with much higher security confidence than measurements inspection, and a decrease in power overhead compared with software-based attestation. However, their combination gives rise to a *full spectrum of solutions in the trade-off between power efficiency and security, depending on how and when measurements inspection hands over to attestation*. Predicting the resulting performance for a given choice is complex, as it depends on the interdependence of the techniques. Moreover, we wish to preliminary study the potential of the combination independently from the performance of each single approach, to extract information that holds in general, and subsequently consider specific state-of-the-art techniques. For this reason, we present three different combination approaches and express their performance as a function of the variables involved.

5.1 Detect and Attest

The architecture of our first proposed combination, denoted with *Detect and Attest* (D&A), is shown in Figure 3. This scheme is designed to exploit only the most reliable step in measurements inspection, which is detection, and relay the characterisation task to attestation.

Figure 3: D&A Scheme.

5.1.1 Performance. Since D&A is the series of the detection step from measurements inspection and the result of attestation, the output TPR is the product of the TPRs of both:

$$TPR_{D\&A} = TPR_{AD}TPR_{AT} \quad (8)$$

A false positive occurs when measurements inspection detects an anomaly, regardless of the presence of malicious data, and attestation fails on a genuine node. Thus, the FPR of D&A is:

$$FPR_{D\&A} = \left(P(C=0)FPR_{AD} + P(C>0)TPR_{AD} \right) FPR_{AT} \quad (9)$$

Where the term in brackets is the probability that anomaly detection triggers. This coincides with the expected attestation frequency, since all sensor nodes are attested when anomaly detection triggers:

$$AF_{D\&A} = P(C=0)FPR_{AD} + P(C>0)TPR_{AD} \quad (10)$$

5.2 Group Subset Attestation

In our second proposed combination, *Group Subset Attestation* (GSA), measurements inspection is used in the detection step and produces the groups of possible malicious nodes, while attestation acts as a judge to take the final decision. Its architecture is depicted in Figure 4. We observe that GSA uses measurements inspection for detection of anomalies and the grouping part of characterisation. Then it iteratively selects a random member for each group and attests it. When the ratio of genuine sensors in a group outweighs the ratio of malicious sensors or vice versa, the attestation for that group stops and the result for the majority of sensors is applied to the whole group. The guard condition that determines whether the number of attestations is high enough for the group is:

$$\frac{||S \in g : S_G| - |S \in g : S_M||}{|g|} > \delta_{GSA} \quad (11)$$

This condition makes sure that the ratio of genuine sensors in a group outweighs the ratio of malicious sensors by δ_{GSA} or vice versa. GSA keeps attesting all nodes in a group until the condition is met. When $\delta_{GSA} = 1$, the condition is never met and GSA coincides with D&A, but with lower values, the number of attestation is lower compared to D&A and GSA may be more convenient.

To maximise energy efficiency, GSA may attest only one node per group. However, this choice is only accurate if the nodes in a group are all genuine or all malicious, which in general cannot be guaranteed. Indeed, measurements inspection is highly accurate in the grouping from the measurements perspective, i.e. the measurements in one group are either genuine or malicious. However, integrity violations at the node layer cannot be identified if the measurements are genuine. When used as a standalone technique, the identification of such malicious sensors is just delayed to the moment when the measurements become malicious. With GSA, if a node runs a malicious software but reports genuine data at the time it is attested, we would infer that all nodes in the same group are malicious. This, in turn, causes a wrong detection of non-compromised nodes whose measurements correlate with that of the malicious node.

The optimal value of δ_{GSA} makes an exhaustive attestation of groups that are mixtures of genuine and malicious nodes, and only one for groups where nodes are either all genuine or all malicious. The value used in the simulation is 0.25, which gives a percentage of attested nodes around 25%.

Figure 4: GSA Scheme.

5.2.1 *Performance.* Note that for each group only a subset of sensors is tested, hence the TPR is:

$$TPR_{GSA} = \delta_{GSA} TPR_{AD} TPR_{AT} + (1 - \delta_{GSA}) \left(P(S'_M | g(S) = g(S') S_M) P(A_{TF}' | S'_M) + P(S'_G | g(S) = g(S') S_M) P(A_{TF}' | S'_G) \right) TPR_{AD} \quad (12)$$

Where $P(A_{TF}' | S'_M) \approx TPR_{AT}$ and $P(A_{TF}' | S'_G) \approx FPR_{AT}$.

We denoted with TPR_{AD} the TPR of the anomaly detection part of measurements inspection, and with $P(S'_M | g(S) = g(S') S_M)$ the probability that the sensor chosen for attestation S' is malicious given that the considered sensor S is malicious and belongs to the same group as S' . Namely, if the malicious sensor node is an attested node (they are δ_{GSA} of the total on average), then detection is correct if anomaly detection triggers and the node fails attestation. Otherwise, the characterisation is correct only if the correct group is selected. Thus, the FPR becomes:

$$FPR_{GSA} = \left(P(C = 0) FPR_{AD} + P(C > 0) TPR_{AD} \right) \left(FPR_{AT} \delta_{GSA} + (1 - \delta_{GSA}) \left(P(S'_M | g(S) = g(S') S_G) TPR_{AT} + P(S'_G | g(S) = g(S') S_G) FPR_{AT} \right) \right) \quad (13)$$

The attestation frequency here is equal to:

$$AF_{GSA} = \left(P(C = 0) FPR_{AD} + P(C > 0) TPR_{AD} \right) \delta_{GSA} \quad (14)$$

Compared with D&A, the attestation frequency is always lower, while the TPR and FPR are as good or better if the anomaly-based grouping is correct, and worse otherwise.

5.3 Cascade

If the strict time constraints in software-based attestation are always guaranteed by genuine sensor nodes, a failed challenge response is an undisputed proof of a sensor's maliciousness. A malicious node found with anomaly detection, instead, may be the consequence of an unforeseen or unprecedented scenario that caused a wrong estimate of the measurements probability. Hence, software-based attestation can be used to confirm a sensor node's maliciousness

after it has been identified as such by the measurements inspection, as shown in Figure 5. We refer to this approach as *Cascade*.

Figure 5: Cascade Scheme.

5.3.1 *Performance.* The TPR and FPR of the scheme are:

$$TPR_{Cascade} = TPR_{MI} TPR_{AT} \quad (15)$$

$$FPR_{Cascade} = FPR_{MI} FPR_{AT} \quad (16)$$

The Cascade scheme power overhead is lower than attestation since the attestation frequency is:

$$AF_{Cascade} = E[C | C > 0] P(C > 0) TPR_{MI} + P(C = 0) FPR_{MI} \quad (17)$$

This is also lower than the attestation frequency of GSA, when the measurements inspection FPR and the number of malicious nodes are low. Instead, with higher FPR, the Cascade scheme makes many attestations, while the number of attestations for the GSA scheme is upper bounded by the number of groups given in output by characterisation.

Note that if the software attestation time constraints are reliable then $TPR_{AT} = 1$ and $FPR_{AT} = 0$, so, compared with attestation, the Cascade solution trades a reduced power overhead for a lower TPR, which coincides with that of measurements inspection.

6 ANALYTICAL EVALUATION

In this section we evaluate the techniques in a WSN of 200 sensor nodes, of which a varying number C are compromised. Sensors collect a measurement about every 4 minutes. The probability of attack is as high as 10^{-2} , implying that, on average, there is an attack about every 7 hours. This value is in the same order of FPR_{AD} , so the analytical results remain substantially unchanged also for

lower attack probabilities. When an attack occurs, the probability that a sensor is malicious has a uniform prior distribution across all sensors (i.e. each sensor is malicious with probability $1/C$).

The evaluation is done by calculating: 1) The *Receiver Operating Characteristic* (ROC) curves, i.e. the relationship between TPR and FPR in the identification of malicious nodes. 2) The attestation frequency, i.e. the time between two attestations divided by measurements transmission period, and averaged across all nodes.

The ROC curves of the combination schemes are obtained through the results of Section 5. The ROC curves for measurements inspection are obtained by interpolating the experimental curves in [12]. The ROC curves for attestation are obtained by modelling the degradation in the reliability of attestation, when the time between two attestations is close to the time needed to swap the genuine software with a malicious one, and vice versa. Thus, the probability that an attacker manages to compromise a measurement and pass attestation behaves like an exponential probability distribution:

$$1 - e^{-\lambda(\frac{1}{AF} - 1)} \quad (18)$$

The parameter λ models the time needed to replace the genuine software and restore it. In particular, we show the performance of attestation when $\lambda = 0.5$, which is the case where the probability that a malicious node passes attestation is close to 0 when $AF = 1$, and quickly increases to about 0.4 and 0.6 for $AF = 1/2$ and $AF = 1/3$, respectively.

6.1 ROC curves comparison

Figure 6 shows the ROC curves for both individual and combined techniques. In particular, the TPR of software-based attestation is shown for $AF = \{1, 1/2, 1/3\}$, corresponding to one attestation run as soon as a new measurement is collected, or once every two or three measurements are collected. The TPR decreases with the attestation frequency since a quick attacker may substitute the original software with a malicious one, inject malicious data, and replace the original software before attestation is run again.

Comparing Figures 6a, 6b, and 6c, the measurements inspection TPR curves appear to saturate at a value around 0.5 that decreases as C increases. This is an effect of the conservative group-wise characterisation [12] which, on the other hand, is the tool that keeps the FPR close to 0 where the TPR is around 0.4. The highest TPR of measurements inspection, achieved at the rightmost point of the ROC curve, is an upper bound to the TPR of Cascade. Nevertheless, while measurements inspection achieves such TPR with FPR close to 1, Cascade achieves it with FPR close to 0. The performance of D&A and GSA is not limited by measurements inspection since they mainly exploit the detection step, whose performance is comparable to attestation's. Indeed, their ROC curves nearly overlap with that of attestation.

6.2 Attestation Frequency Comparison

To make a fair comparison, we also consider the attestation frequency of each scheme, since with $AF \approx 1$ there is no advantage in using a combination scheme in place of simple attestation.

Figure 7 shows that the points where Cascade performs at its best are expensive, as the attestation frequency is close to 1. Instead, when the attestation frequency of D&A is as low as 0.02, the TPR in Figure 6 is close to attestation. Finally, GSA has a small attestation

frequency which saturates and holds even for the highest values of TPR. In conclusion, Cascade is generally not convenient, since it covers a point in the reliability-cost space where reliability is close to measurements inspection and cost is close to attestation. For D&A and GSA, the reliability is almost as high as attestation. The cost for GSA is always comparable to measurements inspection. For D&A, the cost can be kept low for a low decrease in reliability.

7 NUMERICAL SIMULATIONS

In the previous section, we abstracted from the computations and network protocols that enable the application of each combination scheme, so we address them below. Preliminarily, we address the process of tailoring software-based attestation for measurements reliability, which has not been analysed in literature yet.

Moreover, we validate the analytical ROC curves and complete them with the time needed to achieve a certain TPR, which corresponds to the latency in the reaction to malicious data. Finally, we make an accurate estimation of the spent energy, which can be obtained only after network protocols are defined. This allows us to calculate the impact of each approach on nodes' battery life.

To investigate accurately these parameters, we have set up simulations of a realistic application scenario in the open-source *Castalia* [16] simulator. We recur to simulations as existing datasets do not contain sophisticated injection attacks such as those we consider. Moreover, simulations allow running both the individual and combined schemes under exactly the same scenario, giving accurate comparisons that highlight the gains in performance and energy consumption, and allow evaluation with more devices than it would be practical with real nodes.

7.1 Simulation Settings

The simulations consider 200 nodes in a star-topology network, where the base station is located at the centre performing the role of network coordinator, measurements sink, and attestation verifier. Sensors measure the temperature and send the observed value to the sink about every 4 minutes. Nodes are equipped with the CC2420 RF transceiver [13], which is common for its low-power transmissions. Namely, this device spends 57.42 mW while transmitting (at 0dBm), 62 mW while receiving and 1.4 mW while in sleep mode [13].

For C out of 200 sensor nodes, the temperatures sent to the sink are replaced with higher values to trigger the detection of a wildfire. This is done with the data injection technique described in [12], which aims to overcome anomaly detection by minimising the maximum expected correlation between genuine and compromised sensors. Indeed, if a malicious measurement is expected to be highly correlated with a genuine measurement, significant changes in the former would introduce obvious anomalies. The simulations are run for all individual and combination schemes, with three different values of C : 50, 100, and 150. After 5 hours the simulation is stopped and we calculate the TPR and FPR for the detection of malicious nodes, and the energy consumption.

We use the 802.15.4 MAC protocol [10] because it provides us low energy consumption and a hierarchical architecture. The protocol makes use of a coordinator node that dictates how and when

¹A perturbation of ± 0.02 was introduced in GSA and D&A to better distinguish the curves.

Figure 6: ROC curves comparison varying the number of malicious sensors C^1 . Figure 7: AF curves comparison¹.

nodes can communicate through the use of a superframe structure, which constitutes an active and an inactive period, where nodes are allowed to transmit or switch off their communication devices to save energy, respectively. Furthermore, at the application level we define a transmission schedule so that nodes transmit their measurements one after the other, thus avoiding collisions. Once the sink receives all measurements it can trigger the measurements inspection, attestation, or a combined scheme. Note that when just measurements inspection or no scheme is in use, the nodes can go to sleep right after they transmit their data. Whereas, if a scheme with attestation is in place, the nodes must keep awake as they do not know in advance whether they will be attested. Since the communication pattern is not fixed we cannot use the canonical 802.15.4 guaranteed time slots. Also to avoid collisions, so that the network round-trip time can be reliably estimated, attestation is performed one node at a time. When the sink is done attesting all nodes for a collection round, it sends a message signalling to nodes that they can go immediately to sleep until the next active period.

7.2 True Positives / False Positives Results

We calculate the TPR, defined as the number of malicious nodes that are detected at least once during the simulated attack. We do not assume the presence of *intrusion reaction systems*, therefore a detected node keeps carrying out the attack. This constitutes an upper bound on the system's performance since both the energy consumption and the detection of further nodes improves with the correct characterisation of malicious nodes. Indeed, a malicious node that is detected does not need to be evaluated again by neither

attestation nor measurements inspection. The performance of the latter is also likely to increase, since less malicious nodes have less benefits from collusion.

Figure 8 shows the TPR and FPR for each scheme with different numbers of malicious sensors. Rather than their final values, we show the cumulative TPR and FPR time series. Indeed, the time needed to achieve a certain TPR is decisive to minimise the attack's damage. Analogous considerations hold for the FPR. The combination technique with the best performance is certainly D&A, whose curves are the closest to those of attestation. Since D&A uses the detection step of measurements inspection, the TPR curves, in Figures 8a, 8b, and 8c, jump from 0 to 0.99 after 28, 28, and 10 collection rounds respectively, which are the points where anomalies are detected and attestation is triggered.

The curves of the Cascade scheme are close to measurements inspection, but an improvement is brought in the FPR curve in Figure 8d, which is decreased from 0.15 to 0. Higher TPR can be achieved by increasing the measurements inspection FPR and letting attestation take care of the false positives by attesting them. However, this choice increases the attestation frequency, as discussed in Section 6. The simulations also confirmed that D&A can achieve an FPR close to 0 and a TPR around 0.99. In the case of GSA the TPR is around 0.96 and the FPR is close to 0, especially when 50 and 100 sensor nodes out of 200 are malicious. When 150 sensors are malicious the FPR is evidently higher than D&A, but the TPR that eventually gets close to 1 makes GSA still a valid choice. Indeed, when the system detects that 150 out of 200 sensor nodes are malicious, it means that a severe attack has taken place. In such scenario, generating false positives is not the main concern since the system needs a thorough recovery and reconfiguration process anyway.

7.3 Energy Consumption Comparison

In Table 2, the energy consumption during simulation time, averaged across all nodes, is reported for each scheme. First, we note that attestation has an energy consumption which is between 33 and 58% higher compared with measurements inspection, and that the latter does not introduce a perceptible increase with respect to the case where no security scheme is applied (NONE in Table 2).

Compared with measurements inspection, GSA and Cascade generally demand less than 1% extra energy. The increase for D&A, instead, is between 3 and 10%. To understand practical implications

Figure 8: TPR/FPR curves comparison with different numbers of malicious sensors C .

Table 2: Average Energy Consumption (Joules) after 5 hours.

C \ Scheme	NONE	MI	ATT	GSA	D&A	CASC
50	79.54	82.86	109.84	83.65	91.14	83.59
100	81.51	80.96	116.65	80.91	83.73	80.04
150	82.01	79.67	125.81	80.45	83.35	80.07

Table 3: Days To Battery Depletion.

C \ Scheme	NONE	MI	ATT	GSA	D&A	CASC
50	49.03	47.06	35.51	46.62	42.79	46.66
100	47.85	48.17	33.43	48.20	46.58	48.73
150	47.55	48.95	31.00	48.48	46.79	48.71

of such differences in energy consumption, we used the results in Table 2 to retrieve the expected number of days until the batteries would drain. As a reference, we assumed a typical power source of two alkaline long-life AA batteries, which store an energy of 18720 Joules [1]. As reported in Table 3, we see that the average duration of the batteries without measurements inspection nor attestation is about 48 days. When running measurements inspection, there is no significant change in battery life. With attestation instead, the batteries last 10 to 15 fewer days. GSA and Cascade generally cause the batteries to last 1 fewer day, while with D&A battery life diminishes by 2 to 4 days.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we show that combining measurements inspection with attestation achieves high accuracy in identifying malicious nodes whilst significantly reducing power consumption. We proposed three combination schemes: *Detect and Attest*, *Group Subset*

Attestation, and *Cascade*. The first gives most relevance to the attestation step, the third stresses the measurements inspection steps, while the second is at a point in between. In this way, the spectrum of combinations is well covered.

We evaluated all schemes both analytically and by simulations. The Cascade scheme has shown to be limited by the measurements inspection's maximum TPR, which is achieved only with frequent attestations. Instead, both D&A and GSA offer a considerable gain in performance, which resulted very close to attestation's, but for significant less energy. This is confirmed by the energy results from the simulations and is due to the dramatic reduction in the number of attestations, which is observable in the analytical evaluation. A good trade-off between energy and performance is achieved by both D&A and GSA schemes. The former should be preferred when a couple of fewer days in battery life do not have a considerable impact on costs. The latter, instead, is the best solution when the cost for the maintenance and materials involved in the battery replacement process is critical. While the individual techniques forced to choose between accuracy close to 100% with power overhead of 33-58%, or accuracy close to 50% with power overhead close to 0, the combination schemes allow to choose the accuracy in the range 96-99%, with a power overhead in the range 1-10%.

In the future, we plan to expand the threat model to consider more attacks. In particular, we wish to consider an adversary that physically tampered with the sensor nodes hardware to increase their clock speed, or that tampers with almost all sensor nodes. Attestation cannot be applied in the first scenario, while measurements inspection has poor detection accuracy in the second. In such extreme cases there is the need to give to the two techniques the possibility of raising alarms on their own.

ACKNOWLEDGMENTS

This work was supported by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant No. 206707/2014-2 awarded to Rodrigo Vieira Steiner, and by the UK *Engineering and Physical Sciences Research Council* (EPSRC) under Grant No. 1391446 awarded to Vittorio Paolo Illiano.

REFERENCES

- [1] 2011. AllAboutBatteries.com. (12 Jan 2011). <http://www.allaboutbatteries.com/Energy-tables.html>
- [2] Moreno Ambrosin, Mauro Conti, Ahmad Ibrahim, Gregory Neven, Ahmad-Reza Sadeghi, and Matthias Schunter. 2016. SANA: Secure and Scalable Aggregate Network Attestation. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. ACM, New York, NY, USA, 731–742. <https://doi.org/10.1145/2976749.2978335>
- [3] Frederik Armknecht, Ahmad-Reza Sadeghi, Steffen Schulz, and Christian Wachsmann. 2013. A Security Framework for the Analysis and Design of Software Attestation. In *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS '13)*. ACM, New York, NY, USA, 1–12. <https://doi.org/10.1145/2508859.2516650>
- [4] N. Asokan, Ferdinand Brasser, Ahmad Ibrahim, Ahmad-Reza Sadeghi, Matthias Schunter, Gene Tsudik, and Christian Wachsmann. 2015. SEDA: Scalable Embedded Device Attestation. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*. ACM, New York, NY, USA, 964–975. <https://doi.org/10.1145/2810103.2813670>
- [5] Ismail Butun, Salvatore D Morgera, and Ravi Sankar. 2014. A survey of intrusion detection systems in wireless sensor networks. *IEEE Communications Surveys & Tutorials* 16, 1 (2014), 266–282.
- [6] Xavier Carpent, Karim ElDefrawy, Norrathep Rattanavipanon, and Gene Tsudik. 2017. Lightweight Swarm Attestation: A Tale of Two LISA-s. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (ASIA CCS '17)*. ACM, New York, NY, USA, 86–100. <https://doi.org/10.1145/3052973.3053010>
- [7] Claude Castelluccia, Aurélien Francillon, Daniele Perito, and Claudio Soriente. 2009. On the Difficulty of Software-based Attestation of Embedded Devices. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*. ACM, New York, NY, USA, 400–409. <https://doi.org/10.1145/1653662.1653711>
- [8] V. Chatzigiannakis and S. Papavassiliou. 2007. Diagnosing Anomalies and Identifying Faulty Nodes in Sensor Networks. *Sensors Journal, IEEE* 7, 5 (May 2007), 637–645. <https://doi.org/10.1109/JSEN.2007.894147>
- [9] Ing-Ray Chen, Yating Wang, and Ding-Chau Wang. 2010. Reliability of wireless sensors with code attestation for intrusion detection. *Information Processing Letters* 110, 17 (2010), 778–786. <https://doi.org/10.1016/j.ipl.2010.06.007>
- [10] Jose A. Gutierrez, Edgar H. Callaway, and Raymond Barrett. 2003. *IEEE 802.15.4 Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensor Networks*. IEEE Standards Office, New York, NY, USA.
- [11] Vittorio P. Illiano and Emil C. Lupu. 2015. Detecting Malicious Data Injections in Wireless Sensor Networks: A Survey. *ACM Comput. Surv.* 48, 2, Article 24 (Oct. 2015), 33 pages. <https://doi.org/10.1145/2818184>
- [12] V. P. Illiano, L. Munoz-Gonzalez, and E. C. Lupu. 2017. Don't fool Me!: Detection, Characterisation and Diagnosis of Spoofed and Masked Events in Wireless Sensor Networks. *IEEE Transactions on Dependable and Secure Computing* 14, 3 (May 2017), 279–293. <https://doi.org/10.1109/TDSC.2016.2614505>
- [13] Texas Instruments. 2013. Chipconfi!CC2420. *TEXAS INSTRUMENTS*, [Online]. Available: <http://www.ti.com/product/cc2420>. [Accessed 31 1 2015] (2013).
- [14] Yanlin Li, Yueqiang Cheng, Virgil Gligor, and Adrian Perrig. 2015. Establishing Software-Only Root of Trust on Embedded Systems: Facts and Fiction. In *Cambridge International Workshop on Security Protocols*. Springer, 50–68. https://doi.org/10.1007/978-3-319-26096-9_7
- [15] Michael Mitzenmacher and Eli Upfal. 2005. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press.
- [16] NICTA. 2007. Castalia Simulator. (2007). <https://github.com/boulis/Castalia>
- [17] Taejoon Park and Kang G. Shin. 2005. Soft Tamper-Proofing via Program Integrity Verification in Wireless Sensor Networks. *IEEE Trans. Mob. Comput.* 4, 3 (2005), 297–309.
- [18] Mohsen Rezvani, Aleksandar Ignjatovic, Elisa Bertino, and Sanjay Jha. 2013. A robust iterative filtering technique for wireless sensor networks in the presence of malicious attacks. In *SenSys*, Chiara Petrioli, Landon P. Cox, and Kamin Whitehouse (Eds.). ACM, 30.
- [19] Arvind Seshadri, Mark Luk, Adrian Perrig, Leendert van Doorn, and Pradeep K. Khosla. 2006. SCUBA: Secure Code Update By Attestation in sensor networks. In *Workshop on Wireless Security (2008-02-07)*, Radha Poovendran and Ari Juels (Eds.). ACM, 85–94.
- [20] Arvind Seshadri, Mark Luk, Elaine Shi, Adrian Perrig, Leendert van Doorn, and Pradeep Khosla. 2005. Pioneer: Verifying Code Integrity and Enforcing Untampered Code Execution on Legacy Systems. *ACM SIGOPS Operating Systems Review* 39, 5 (2005), 1–16. <https://doi.org/10.1145/1095809.1095812>
- [21] Arvind Seshadri, Adrian Perrig, Leendert van Doorn, and Pradeep K. Khosla. 2004. SWATT: SoftWare-based ATTestation for Embedded Devices. In *IEEE Symposium on Security and Privacy (2004-06-21)*. IEEE Computer Society, 272–282.
- [22] Rodrigo Vieira Steiner and Emil Lupu. 2016. Attestation in Wireless Sensor Networks: A Survey. *ACM Comput. Surv.* 49, 3, Article 51 (Sept. 2016), 31 pages. <https://doi.org/10.1145/2988546>
- [23] Sapon Tanachaiwiwat and Ahmed Helmy. 2005. Correlation Analysis for Alleviating Effects of Inserted Data in Wireless Sensor Networks. In *MobiQuitous (2006-01-25)*. IEEE Computer Society, 97–108.
- [24] Dazhi Zhang and Donggang Liu. 2010. DataGuard: Dynamic data attestation in wireless sensor networks. In *DSN. IEEE*, 261–270.
- [25] Yun Zhou, Yuguang Fang, and Yanchao Zhang. 2008. Securing wireless sensor networks: a survey. *IEEE Communications Surveys and Tutorials* 10, 3 (2008), 6–28. <https://doi.org/10.1109/COMST.2008.4625802>
- [26] Shazana Md Zin, Nor Badrul Anuar, Miss Laiha Mat Kiah, and Al-Sakib Khan Pathan. 2014. Routing protocol design for secure WSN: Review and open research issues. *Journal of Network and Computer Applications* 41 (2014), 517–530.