Intrusions Detection in Intelligent Agent-Based Non-traditional Grids

Razvan Bogdan, Mircea Vladutiu
“Politehnica” University of Timisoara
Bd. V. Parvan nr. 2
300223, Timisoara
Romania
{rbogdan, mvlad}@cs.utt.ro

Abstract—The architecture of intelligent grid offers a solution to the existing problems in ambient intelligence, based on intelligent agents. State-of-the-art concerns regarding intelligent agents based systems are showing the necessity of developing techniques for addressing the problem of security and fault tolerance. This paper proposes dedicated detection schemes for different types of attacks, being a rational continuity of the previous work in the fields of fault tolerance and intelligent grid.

Keywords—ambient intelligence, intelligent grid, errors, fault tolerance, intrusions detection

I. INTRODUCTION

The most recently progressive ideas on computer innovation states that by 2040 or 2050, machine intelligence will surpass human intelligence, an event called by Raymond Kurzweil as “singularity” [34]. The problem which is raised in this context is to develop security techniques in the context of artificial intelligence solutions. It is admitted also that substantial efforts are to be employed in order to accurately identify abusive behavior.

It should also be stressed out that the classical buffer overflows have been replaced by malicious scripts as the most common approach for penetrating systems. The goal of overcoming such behaviors is actually to obtain what is called a reflective system [34], namely a system that can reference and modify their own behavior in the context of a system malfunctioning.

Ambient intelligence [7] is but one example in which intelligent agents’ engaging has been the method [2] used in order to surpass different shortcomings such as power consumption, portability, scalability, configurability and reliability [7, 10]. As described in the literature [7, 10], ambient intelligence is an environment composed of sensors, actuators, multimedia devices. More than this, another participant might be taken into consideration, namely the PC [2]. This element of the network can provide computational power by processing certain tasks, therefore reducing overall power consumption, it can move (spatially), it can disappear or appear at will.

This paper addresses the problem of detecting different types of attacks in the context of the architecture proposed in [2], based on intelligent agents and called intelligent grid, as a response to the drawbacks of the classical model of ambient intelligence. The process of communication between agents is based on messages. Whenever there is a message exchange between agents, a malicious attack can occur and the message can be corrupted. The idea presented in this paper is to identify peculiar attacks, each attack being identified by certain signature. The method implied for this task offers a tailored detection scheme for each particular attack.

This paper is organized as follows: Section 2 briefly presents the architecture of intelligent grid and the digital signatures based on hash functions solution for attaining fault tolerance, as previous work for the problem of attaining security and fault tolerance in intelligent grid. Section 3 introduces at the beginning the Generalized Linear Feedback Shift Register as the basis component for creating a targeted detection scheme. Furthermore, dedicated schemes are being introduced for specific attacks, while the experimental part is relying on specific metrics. The last section presents the conclusions and future work.

II. NON-TRADITIONAL PARALLEL MODEL

Ambient intelligence is designed to be used into an environment suitable to be controlled [7]. Into such an environment, besides the sensor, actuators and multimedia processing another participant might appear: the general purpose computing system. This participant is not taken into consideration in the ambient intelligence approach. In the same time, there are some particular problems that are being faced by the ambient intelligence approach, namely: computation, security, power control, reliability etc. In [2] is presented an non-traditional parallel architecture based on intelligent agents, called intelligent grid, which can address the problems of ambient intelligence.

In intelligent grid another participant is introduced, the general purpose computing system. The idea of such an approach is that the general purpose computing system is just another node in the network. It can move (spatially), it can disappear or appear at will, or it can sleep. But most important aspect is that it can provide computational power, therefore the network can use it to process operations. Doing so, other nodes in the network can sleep, therefore reducing overall power consumption.

In [2] were presented the levels of this architecture, as they can be seen in Fig. 1. The first level is called the device level and it contains different equipments, such as sensors, actuators, controllers, DSPs and PCs. The connection level is composed by the links existing between the components of the device level. The collection level is formed by the networks, functionalities and the streaming data.
In this point it should be mentioned that such a complex architecture like intelligent grid is expected to be a reliable system [2, 5]. Therefore reliability is an important attribute of this system due to the fact that unreliability is inherent to the disappearing electronics concept. This is primary caused by the fact that nodes may emerge unexpectedly, may move, may fail and may finish their energy reserves (temporary or not). All this problems are forced into focus by the cost, power and size constraints. The solutions used in ambient intelligence in order to achieve reliability are redundancy [2] and security [35].

In [35] was presented a method of correcting the messages compromised by an attack. This paper proposes tailored detection schemes in order to intercept different attacks. Each attack is identified by a specific signature.

III. SECURITY MATTERS

A. Specific Attacks

In [11] the attacks on such networks have been classified and a threat model resulted. There are specific attacks that make sensor containing networks vulnerable. Such attacks are interception, node hijacking and router attacks – link state.

- Interception

In [3] it is stated that this is one of the easiest means of attack in a sensor network because the anatomy of the attack consists of eavesdropping or information gathering in a passive manner. The data streams passing between different sensor nodes and the base station can be intercepted by a laptop type attacker. Such an attacker has a passive nature therefore the information about the attack and the identity of the attacker can be hided. According to [5], an adversary is more malicious if it can change the contents of the information in order to cause confusion.

- Node Hijacking

First of all, a node can be compromised and the resulting action is to extract secret information from that node [11]. The solution for this case is to exclude the specific node. Secondly, a new node can be introduced into the network. This node can introduce malicious information, but also can consume a lot of network bandwidth.

- Router Attacks – Link State

If a router is compromised than it can add a false link, delete an already existing link (proactive attack) or the router ignores a change in link state of its neighbors (inactive attack). There is one solution based on a centralized attack analyzer module which detects attacks based on some possible alarm events sequences. This technique is called intrusion detection and it has been proved [3] that is not a scalable solution when connecting to the internet. The second solution is called protocol driven and the detection capability is embedded in the link state protocol itself. In this case a router won’t believe an update unless it receives a confirmation link state update from the node which is sending the questionable link.

A possible solution to such an attack is adopting a broker. Based on the proposed architecture of intelligent grid [2], a broker is a distinctive agent which role is to guard the data traffic on the network. The way to create a broker [11] is through a given protocol between the owner and consumer
based on a list of brokers received from the directory facilitator. The role of broker is to detect the attack’s pattern within a message, based on a targeted detection scheme. In other words, a certain attack’s signature would be obtained by means of a dedicated scheme, as will be presented in the next section.

B. Solution Based on GLFSR

The Generalized Linear Feedback Shift Register is a pattern generator with \( n = (\delta \times m) \) outputs, being presented for the first time in [12]. Such a structure is designed over \( GF(2^\delta) \) and all its elements are part of \( GF(2^\delta) \). Even if the necessary components used for building such a GLFSR are adders, multipliers and storage elements, these components are not regular Galois field multipliers. More than this, it should be noticed that these components multiply the \( \delta \) bit feedback input with a constant \( \Phi_i \). In Fig. X is presented the structure of a Generalized LFSR. As a particular component, the adder is a set of \( \delta \) XOR gates, in the same time the multiplier uses also XOR gates. The general form of the Generalized LFSR can be represented as

\[
\Phi(x) = x^m + \Phi_{m-1}x^{m-1} + \ldots + \Phi_1x + \Phi_0 \tag{1}
\]

What is interesting to notice in this case is that the Generalized LFSR has a total of \( m \) stages, namely \( D_0, D_1, \ldots, D_{m-1} \), with \( \delta \) storage cells. At every shift, \( \delta \) bits are being shifted from one stage to the next, while the feedback from the last stage, \( D_{m-1} \), is sent to all the stages. The coefficients of the polynomial from (1) define the feedback connections specific to a GLFSR and are elements over \( GF(2^\delta) \). The multiplied feedback input realized by an element \( \Phi_k \) can be realized using XOR gates.

![Figure 3. Structure of a GLFSR [12]](image)

It has been demonstrated in [12] that a GLFSR with a primitive feedback polynomial (termed MLFSR), with \( \delta > 1 \) and \( m > 1 \), represents a structure which is a very effective pseudo-random pattern generator. GLFSR is a general structure [12] and all the other known structures like LFSR, MISR etc., are being special cases of GLFSR. When used to generate patterns, for a structure with \( n \) inputs, there can be \( m \) stages (like the one in relation (1)). The condition is that each element has to belong to \( GF(2^\delta) \), where \( (m \times \delta) \) is at least or equal to \( n \). In order to use a GLFSR as a pattern generator, it is required that a non-zero seed be loaded into the GLFSR and then clocked to produce the patterns.

For the sake of simplicity, the GLFSR used in this paper has \( \delta = 1 \), therefore is a LFSR.

C. Construction of Detection Schemes

In [32] is presented a method (for the sake of simplicity, let’s call it Chatterjee-Pradhan method) of developing pattern generators which can detect hard-to-detect faults, based on a relatively modest number of patterns. This method was developed in order to test specific faults from a circuit under test.

![Figure 4. The process of detecting an attack’s signature](image)

### TABLE I. Dictionary of Attacks Signatures

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interception</td>
<td>FD01</td>
</tr>
<tr>
<td>Router Attack-Link State</td>
<td>C0AF</td>
</tr>
<tr>
<td>Node Hijacking</td>
<td>2F2E</td>
</tr>
</tbody>
</table>

Starting from the Chatterjee-Pradhan approach, a specific method for intelligent grid can be disclosed, as depicted in Fig. 4, but in the case of intelligent grid architecture, there is a necessity to verify that each message exchanged in the communication process between the nodes is not malicious in some way. To be more specific, it is required to identify if a message is infected by a certain attack. This can be achieved by a detection process of the attack’s signature.

In Fig. 4 it is presented how the outputs of the GLFSR are being transformed by the mapping logic into test vectors (attack’s signatures) for the message under test (MUT). The GLFSR is loaded initially with a seed from which the patterns for the mapping logic will be generated. In [32] it is stated that using GLFSRs as pseudorandom pattern generator is a way to provide better coverage of majority of faults. As stated before, in the case of intelligent grid, the faults are aimed to be taken from a digital signatures [33] dictionary of certain attacks, as depicted in Table I.

Given a MUT, the Chatterjee-Pradhan method can be rephrased for detecting attacks in intelligent grid by the following design procedure:

1. Select a GLFSR functioning as a PRPG engine for the composite pattern generator
2. Build a signature attack dictionary
3. Determine a minimal set of target patterns which test the determined fault list.
4. For the signatures of each attack, design an efficient combinational circuit with the PRPG so that the combined pattern generator generates all the signatures for a certain attack.

The last step from the design procedure can be stated as the following problem: being specified an initial seed to the PRPG and a constrained test pattern length, design an efficient combinational logic area which aims at providing the expected target patterns within the specified length.

The result of the design procedure is the detection of a dedicated signature, namely a particular attack to the intelligent grid architecture.

D. Practical Results

The idea that has been envisioned was to offer a targeted detection scheme for a certain attack. The first part of this process was to test which feedback polynomial would be most appropriate to be used in order to obtain the lowest test application time. In other words, time detection of the attacks to be the most convenient.
The method presented previously was tested to the broker node with different feedback polynomials. For example, let’s consider \( G(x) = x^{16} + x^{14} + x^{13} + x^{11} + 1 \). \( G(x) \) was used as a PRPG engine for obtaining all the combinations used as inputs for the mapping logic. In order to facilitate the process of obtaining all the combinations, a testbench was used, namely LFSRTestbench [36].

The next step was to build a dedicated mapping logic which would provide the attack’s signature. The outputs of GLFSR are being transformed by the logic part into attack’s signatures. For each attack type from Table I, it was build a separate detection scheme, depicted in Fig. 5. In other words, the broker was capable of detecting three different types of attacks on the intelligent grid network, the detectability rate of an attack being maximum.

The next step was to try different feedback polynomials in order to see which one would offer the lowest test application time. In Table III the experimental results have been centralized according to specific metrics for this type of solution [5, 32], such as the number of iterations used to obtain the input combinations to the mapping logic, test application time and fault coverage. It can be seen that the number of iterations used in order to obtain all the combinations used for achieving the attack’s signature, exponentially increases with the test application time, as it can analyzed in Fig.6. It can also be noticed that different initial seeds were utilized because a particular initial seed cannot be employed for each feedback polynomial.

The experimental results show that \( G_2(x) \) is the most optimum feedback polynomial taken into consideration the values of number of iteration and the test application time.

IV. CONCLUSIONS

This article addresses the problem of security in a non-traditional parallel architecture called intelligent grid, by presenting a method for detecting different types of attacks using their signatures. The obtained results are specific detection schemes, targeted to a spectrum of attacks.

The experimental part revealed that a certain feedback polynomial is the most appropriate to be used in order to detect the attacks from the developed attacks dictionary.

As future work, we intend to develop a method based on which a large spectrum of feedback polynomials to be tested in order to seek the optimum feedback polynomial.

REFERENCES


<table>
<thead>
<tr>
<th>Tap</th>
<th>Initial seed</th>
<th>Number of iterations to obtain the combinations</th>
<th>Test Application Time [ms]</th>
<th>Fault coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>( G(x) = x^{16} + x^{14} + x^{13} + x^{11} + 1 )</td>
<td>FE1C</td>
<td>65</td>
<td>8.671</td>
<td>3</td>
</tr>
<tr>
<td>( G_2(x) = x^{16} + x^{15} + x^2 + 1 )</td>
<td>FFDC</td>
<td>15</td>
<td>2.297</td>
<td>3</td>
</tr>
<tr>
<td>( G_3(x) = x^{16} + x^{12} + x^5 + 1 )</td>
<td>639C</td>
<td>50</td>
<td>7.063</td>
<td>3</td>
</tr>
</tbody>
</table>

Figure 6. Variation of time with number of iterations


