Intrusion Detection System for AES Encripted Low-Power IoT Networks

Eduardo de Oliveira Burger Monteiro Luiz
Universidade Federal Fluminense
Rio das Ostras, Rio de Janeiro, Brasil
eburger@id.uff.br

Luciano Bertini
Universidade Federal Fluminense
Rio das Ostras, Rio de janeiro, Brasil
lbertini@id.uff.br

Alessandro Copetti
Universidade Federal Fluminense
Rio das Ostras, Rio de janeiro, Brasil
alessandro_copetti@id.uff.br

Juliano Fontoura Kazienko
Universidade Federal de Santa Maria
Santa Maria, Rio Grande do Sul, Brasil
kazienko@redes.ufsm.br

ABSTRACT

The introduction of the IPv6 protocol solved the problem of providing addresses to network devices. With the emergence of the Internet of Things (IoT), there was also the need to develop a protocol that would assist in connecting low-power devices. The 6LoWPAN protocols were created for this purpose. However, such protocols inherited the vulnerabilities and threats related to Denial of Service (DoS) attacks from the IPv4 and IPv6 protocols. In this paper, we prepare a network environment for low-power IoT devices using COOJA simulator and Contiki operating system to analyze the energy consumption of devices. Besides, we propose an Intrusion Detection System (IDS) associated with the AES symmetric encryption algorithm for the detection of reflection DoS attacks. The symmetric encryption has proven to be an appropriate method due to low implementation overhead, not incurring in large power consumption, and keeping a high level of system security. The main contributions of this paper are: (i) implementation of a reflection attack algorithm for IoT devices; (ii) implementation of an intrusion detection system using AES encryption; (iii) comparison of the power consumption in three distinct scenarios: normal message exchange, the occurrence of a reflection attack, and running IDS algorithm. Finally, the results presented show that the IDS with symmetric cryptography meets the security requirements and respects the energy limits of low-power sensors.

KEYWORDS

IoT, 6LoWPAN, Intrusion Detection System, AES, Cybersecurity

1 INTRODUCTION

In the last decade laptops and servers have been the biggest targets of hacker attacks. However, lately there has been a growing increase in attacks targeted at subsystems of connected homes, such as lighting, thermostats, intelligent locks, etc. The Internet of Things (IoT) has been responsible for a 60 % increase in the number of attacks [1]. The increase was not only in volume but also in frequency and complexity. One of the common attacks is the reflection attack. Especially in low-power IoT systems, such attacks are more difficult to deal with due to scarce resources such as computing power and low bandwidth. The challenge of preventing an attack by reflection is due to its ability to create very large attacks [2].

The reflection-based attack creates an amplification of traffic, in which the source IP address is replaced by the IP address of the attacked host and these requests are sent to servers or other devices that can be used to reflect network traffic, making responses to these requests sent to the destination node. The victim will end up receiving a large volume of response packets that it never had requested. The traffic reflection mechanism increases complexity to identify the real source of the attack [3].

The problem is how to ensure the security of IoT devices, how to keep the network informed that security has not been violated. This work presents a C language-based Intrusion Detection System (IDS), developed with the symmetric AES encryption algorithm, designed to inhibit reflection attacks, to enable a secure environment for IoT devices on a low-power network.

The AES algorithm is quite convenient for structuring an IDS due to its excellent performance compared to other encryption algorithms [4]. Furthermore, there is an excellent cost-benefit ratio when using AES with lower power consumption to applying encryption on low-power devices [5].

Several IoT devices can be highly secure. However, if only one device connected to this network has some security vulnerability, it could be exploited by an unauthorized attacker and the entire network would be compromised. In this context, new protocols are being developed to make connecting IoT devices more secure. Moreover, this fact can be verified in the adaptation of IPv6 with the 802.15.4 (WPAN - Wireless Personal Area Network) protocol that was deployed in the 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Network) [6], which in addition to having low-power consumption in wireless networks, can also be implemented in embedded systems, enabling greater security for IoT devices.

In this work, an environment of IoT devices that communicate on a 6LoWPAN network was simulated using the COOJA simulator. The COOJA allows you to create a realistic scenario, that is, the same scenario in which physical devices working on a network during a real operation [7]. Devices are emulated using the concept of virtual machines, and emulation lets you specify that nodes are real machines running the real protocol, where each node has the Contiki operating system running internally as if it were physically installed [8]. Therefore, all code developed for the COOJA simulator can be immediately deployed in a real environment. The results of the simulations in terms of power consumption were compared between three distinct scenarios: while performing normal operations, during the attack itself, and while the IDS is running.

Briefly, the contributions of this paper are three:

- Address reflection attacks on low-power IoT devices using the Contiki operating system;
- Present an IDS developed in C language based on symmetric AES encryption designed to inhibit reflection attacks, tested on a simulator based on real emulated IoT devices, COOJA;
- Comparison of energy consumption results obtained during simulations and collected during execution in three distinct scenarios: normal operation, during an attack, and running IDS.

In the Section 2 we will present the review of the recent literature, the theoretical background and the adopted model of attacks in the Section 3, the experimental environment used with results in the Section 4, and finally the conclusions are in the Section 5.

2 LITERATURE REVIEW

Recent works in areas of IoT and Wireless Sensor Networks (WSN) demonstrate the need to design specific IDS systems. In Kasinathan et al. [9], the scenario proposed was the manufacturing environment where the DoS attack is directed at devices that control the temperature and pressure of machines during production. The attack is made through Jamming where it is directed to a 6LoWPAN network, causing the company security officer to receive a notification from the IDS system. The proposed DoS detection architecture provides immunity to similar attacks protected by the IDS architecture, centralized processing running on Linux, overcoming the constraints on IoT devices, and reducing false-positives that are managed by IDS. However, the work adopts familiar IDS strategies that were not designed for WSNs and are IP-based. Because DoS attacks often render the wireless network channel unusable, the intrusion detection systems studied fail to perform the most basic operations they should perform.

Bouyeddou et al. [10], mentioned that ICMP (Internet Control Message Protocol) flood attacks are still one of the most challenging threats on IPv4 and IPv6 networks. It is noteworthy, therefore, that the smurf attack uses this method widely. The authors proposed an approach based on the Kullback-Leibler (KLD) to detect DoS flood attacks based on ICMP and DDoS. This is motivated by the KLD high capacity to discriminate quantitatively between two distributions. The analysis was supported by the six-sigma rule that was applied to KLD distances for anomaly detection by assessing the effectiveness of data sets using the 1999 DARPA's IDS [11].

In [12] an approach based on 6LowPAN neighbor discovery protocol is proposed to mitigate DoS attacks initiated from the Internet, without adding additional overhead on the 6LoWPAN sensor devices. To combat hacking attacks, the authors used ICMPv6 messages with Address Registration Option (ARO) [1], which is used in Neighbor Discovery Protocol (NDP) and modified in a 6LoWPAN (6LoWPAN-ND) network. Duplicate Address Request (DAR) and Duplicate Address Confirmation (DAC) messages were used, which are two types of ICMPv6 messages defined to perform optional duplicate address detection. In the 6LoWPAN-ND network, nodes in a 6LoWPAN network use NDP to perform automatic address configuration, network layer address resolution, neighbor unreachability detection, and to locate default routers. However, the authors saw deficiencies and limitations, as most mechanisms against security attacks require high computational resources, making them unsuitable for use in smart object networks. Deficiencies arise for two reasons: it is very easy to consume resources on low-power networks, as sensor energy can be consumed quickly, making them unavailable until the attack is over and the battery is recharged.

In Cervantes et al. the authors proposed the Thatachi intrusion detection system, which is a system that seeks to detect and isolate from the network attacking devices that exhibit any abnormal behavior. Using the COOJA simulator, a low false-positive rate was noticed for low-power and low resource devices. Thatachi has been effective in mitigating specific attacks on the IoT network routing service. The author compares Thatachi's false-positive metrics with INTI, the latter being designed to mitigate sinkhole attacks. Overall the metrics used showed a good Thatachi score in relation to INTI within the sinkhole attack scenario. The limitation of Thatachi IDS is to achieve a good result only in the type of sinkhole attack and selective forwarding [13].

An implementation of a UDP flood attack was described in [14]. They used the Contiki operating system, and also implemented and demonstrated the model in the COOJA simulator. The goal was to propose a bandwidth throttling mechanism that should be incorporated into the Contiki operating system to mitigate UDP flood attacks. The proposed scheme reduced the victim's energy consumption by 9% and saved the victim's total transmission power by 55%. The authors also provided an overview of distributed DoS type attacks trends targeting IoT protocols, such as IPv6, 6LoWPAN, and RPL. They explain that DoS aims to consume resources from a host or remote network by denying or degrading services to legitimate users [14].

One of the searchable symmetric encryption (SSE) used in IoT devices was proposed by DAO et al. [15] who presented a compact and low energy consumption AES core, using a small S-box and an improved key in an expansion block. The authors presented two optimizations: the first was the optimization of the S-box; the second was the optimization of the Rcon block. The S-box is transformed from the GF(2^8) [2] architecture to the GF(2^8) / GF(2^4) / GF(2^2) architecture and is then inverted. The Rcon block, therefore, is optimized by the simple Boolean optimization method. Finally, the authors concluded by stating that AES encryption is highly potential for use in power restricted wireless network applications, such as wireless sensor networks and IoT systems for monitoring the environment that requires low-power consumption.

The construction of microcontroller architecture based on the AES architecture presented by the authors ZHAO, et al. [5] had a result which was the lower energy consumption by cryptography. They studied variations in the architectures of the AES algorithm and certify that the preferred choice for IoT devices is AES. In the article, the authors still reinforce that it will be the choice adopted as a reference for new explorations in the development of smart devices [5].

[1] The Address Registration Option (ARO) is a type of ICMPv6 message option used by the new neighbor discovery protocol (NDP or ND) [12].

[2] A finite field GF(2^m) is an algebraic set structure of 2^m elements upon which various arithmetic operations uses, including cryptography. Every GF(2^m) finite field contains 2^m elements. Polynomials over GF(2) indicate that the coefficients have elements between 0 and 1. As form the polynomial basis with m elements, every finite field contains at least one irreducible polynomial over GF(2) associated [16].

3 BACKGROUND AND THREAT MODEL

Embedded devices and smart devices are the elements that make the Internet of Things a reality. Some examples of embedded devices that can be cited are: health monitoring devices such as pacemakers, home automation devices, industrial automation devices, intelligent monitoring, and environment monitoring systems [17]. Technologies such as wireless sensors, remote sensing, embedded sensors in agricultural machines, and traceability tools are currently facing the biggest research challenges regarding safety [18].

Industry 4.0 refers to a recent concept that describes a productive environment supported by industrial automation, including the Internet of Things, big data, cloud computing, blockchain, and machine learning algorithms. [19–22]. The emergence of Industry 4.0 has brought many technological and operational advantages resulting in competitive opportunities, but with it, cybersecurity challenges have made it one of the priority themes in this new scenario [20].

3.1 IPv4 vs IPv6

IPv6 protocols allow 128-bit addresses to be made available, while IPv4 only distributes 32-bit addresses. Thus, IPv4 supports about 4.29 billion IP addresses, and therefore this would explain the need for the evolution to IPv6, as it is no longer possible to generate addresses using the protocol technology currently provided by IPv4. With the advent of IoT, the demand for new specific IP addresses for each embedded device has increased significantly over the last few years.

Focusing on the IPv6 protocol and the security issue the IP Security Protocol (IPSec) is used, natively in the IPv6 protocol, encrypting the packets and operating at the OSI network layer level, while SSL/TLS act at the middle layer, that is, between the application and transport layer of the TCP/IP model. IPSec uses a variety of encryption capabilities. It uses asymmetric keys to ensure the authenticity and integrity of the parties involved, symmetric keys for data confidentiality, and hash functions for data integrity. The implementation of IPSec does not require any changes to applications and operating systems and can be used in its standard configuration [23].

An advantage of IPSec for IoT is the use of the authentication header (AH), as it does not care about confidentiality, but guarantees the authenticity and integrity of the data. AH ensures the authenticity and integrity of both the IP header and the data. One of the advantages of using AH is the ability to use specific parameters that make DoS or DDoS attacks weakly because they discard these packet types that do not meet the requirements that are set by specific parameters.

3.2 6LoWPAN

IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) is an IETF working group responsible for creating and maintaining specifications that allow you to use IPv6 in IEEE 802.15.4 networks. To this end, packet compression and fragmentation are considered in data transmission, making information exchange between low-power devices efficient. By manufacturing specification, such devices are susceptible to packet loss, which would make it impossible to use the IPv6 protocol. Therefore, the 6LoWPAN protocol is a specification for transmitting data over the IPv6 standard in wireless networks by low-power devices and limited processing power. For this reason, 6LoWPAN is currently considered one of the best specifications for IoT implementation. In addition to 6LowPAN, two other examples of protocols that use the IEEE 802.15.4 physical protocol stack are ZigBee and the Thread protocol. A comparison between ZigBee and Thread is shown in Figure 1 [24]. IEEE 802.15.4 standardization enables interoperability between these protocols. The IEEE 802.15.4 standard uses 5 MHz channels and can range from 2.405 GHz to 2.480 GHz. The maximum data rate is around 250 kbps, ranging from a few meters to hundreds of meters of reach.

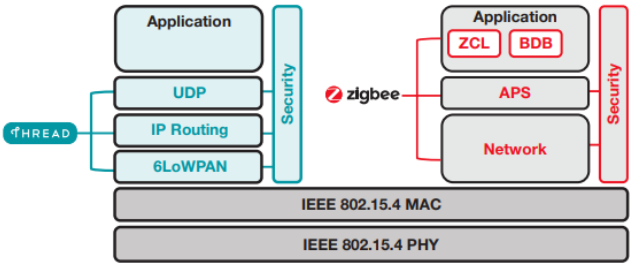

Figure 1: Network architecture for low-power devices [24]

3.3 ContikiOS and COOJA

ContikiOS is an operating system used on a wide range of IoT devices and is open source. ContikiOS supports the IPv6, IPv4, and 6LoWPAN family protocols. Applications can be developed in C and through ContikiOS it is possible to use COOJA, a real environment network simulator of interconnected devices, which reproduces the development complexity of each instruction allowing each device to have an active operating system instance with the same software that would be used on the physical device [25].

The COOJA is a flexible Java-based simulator designed to simulate sensor networks on a platform supported by the ContikiOS operating system [26]. COOJA can simulate networks with varying sensor nodes and, as stated by Osterlind et al, [8], a node simulated by COOJA has three basic properties: a data memory, the node type, and hardware peripherals. COOJA executes native code through Java Native Interface (JNI) calls from the virtual machine to the compiled ContikiOS system.

The COOJA simulator has a data collection tool called Collect View. It is based on the Java language and aims to capture information from nodes as well as to send commands to each node. This tool assists in obtaining the metrics and data generated regarding energy consumption, packet delivery time, among other possible readings by the sensors.

3.4 Threat Model

It is well known that IoT technology has inherited conventional attacks targeting computers on the Internet, and with the emergence of new protocols, specifically for IoT, the number of possible attacks has considerably increased. One such attack that has been a major concern in conventional Internet security and also possible in IoT is the Distributed Denial of Service attack (DDoS). The DDoS attack

aims to consume the resources of a host remotely or a network in a distributed manner, denying or degrading services made available to legitimate users [14]. The DDoS attack can be of two types: direct or by reflection.

The reflection attack is the threat model adopted in this paper. It consists of three steps or components: 1) a server capable of IP address spoofing, which consists of masking IP packets using MAC addresses of false senders; 2) a protocol vulnerable to reflection, that is, a poorly designed UDP-based request protocol; 3) a list of reflectors, which are servers that support the protocol's vulnerability with the victim's address. In resume, the victim will end up receiving a large volume of response packets that it never had requested. With a large enough attack, the victim may end up with a totally congested network. The responses delivered to the victim can be larger than the spoofed requests when used with amplification by reflection technique. A carefully mounted attack may amplify the hacker's traffic.

The idea is based on sending fake UDP requests, spoofing the source IP address, and placing the victim's IP address in the source address field, each packet destined for a random reflector server. The victim will eventually receive a large volume of response packets that they never requested. With a sufficiently large attack, the victim can end up with a congested and unstable network.

Cryptography as a security solution, although it is a technology prior to the Internet as we know it, and, briefly, it can be described as a mechanism for data encryption and decryption is a fundamental ally in supporting and maintaining information security. In this context, the AES algorithm has a fixed 128-bit block and a key that can vary between 128, 192, or 256 bits. AES operates on a two-dimensional array of bytes with a 4x4 position that is called the state, consisting of four stages, which makes it very difficult due to its complexity, to be broken encrypted content.

3.5 Intrusion Detection Systems

An Instruction Detection System (IDS) can be defined as an automated security and defense system that enables the detection of suspicious and often malicious activity on a network or computer device. In addition, IDS attempts to prevent such malicious activities by reporting a different pattern of behavior to the network administrator.

The intrusion detection process is specifically designed to retentively respond to suspicious activity that may interfere with the principles of integrity, reliability, and availability [27]. There are three types of IDS: based in anomaly; based in signature; based in the host.

3.5.1 Anomaly Detection. This type of IDS seeks to identify unusual behaviors within a network, considering these behaviors as anomalies. This type of IDS assumes that attacks are themselves different from normal activity and can thus be detected after they have identified these differences [28].

3.5.2 Signature Detection. Analyze network activity by looking for events with predefined behavior patterns, previously considering in a list these behaviors as an attack [27].

3.5.3 Host Based. The IDS resides on the host or IoT device itself that will be alerted to attacks against the device itself. This type of IDS is very effective as it provides security for types of attacks where the firewall and a network-based IDS do not detect [29].

3.6 Proposed IDS

The IDS proposed in this work is of the hybrid type, by anomaly and host-based. The encryption feature allows identifying behaviors that are outside the standard considered normal by the security policy. The proposed IDS workflow is shown in Figure 2.

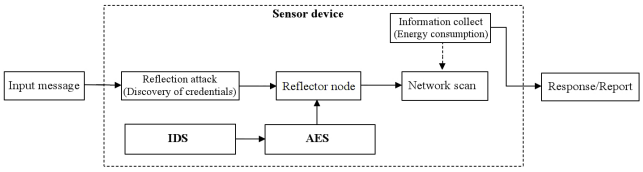

Figure 2: IDS workflow

The evolution of the reflection attack and the moment when the IDS is executed are summarized in the six steps as follows:

- **Step 1:** a reflection attack will be carried out, to discover the credentials, which are usually from the factory, or to discover the common credentials, which are widely used by inexperienced users and, therefore, have low complexity;
- **Step 2:** after the attack is successful, the attack code will be sent to the IoT device chosen at random. These devices are known as reflector nodes;
- **Step 3:** through the botmaster, who is usually the sink, he will continue to scan the network in an attempt to identify other victims, always sending reports about his current state and the other vulnerable devices;
- **Step 4:** the IDS is activated and together its AES algorithm to improve its security. An encrypted message is sent to alert that device was compromised;
- **Step 5:** collect the attack information to obtain the energy consumption obtained over the operation;
- **Step 6:** compare the information collected to obtain metrics of evolution and efficiency in the energy consumption.

4 EXPERIMENTAL SETUP AND RESULTS

The communication operations between the IoT devices were simulated in the COOJA environment on a Tmote Sky type network. In this network, each IoT device was represented by a node. Each node represented a type of sensor with memory and a core of interfaces. This representation facilitated the work of implementing the intrusion detection algorithm, allowing the analysis and study of energy consumption. The analysis and study were based on customized scenarios. The low-power network was created in the Contiki operating system, which also offers the possibility of using the RPL routing protocol.

The first simulation scenario has demonstrated the normal functioning with a limited number of devices to not overload the operation as shown in Figure 3. Due to the limited processing capacity of the physical machine, it was observed that less than thirty nodes would adequately attend the study, without overloading the tests unnecessarily. During the simulation, raw data were collected in

isolated operations, in which each was simulated for 37 minutes in COOJA. The second simulation addressed the attack by reflection. Finally, the third one allowed the observation of the intrusion detection system in operation during the normal simulation. The IDS with a normal operation was set to verify the power consumption. The energy consumption information of the IDS was collected in a normal operation to prove its efficiency in energy consumption. The data collected about the energy consumption of the sensors using the IDS demonstrated that the encryption used didn't overload the devices. Therefore, they demonstrated that energy consumption was slightly higher than when using devices without IDS. The three scenarios proposed were simulated using all devices configured as *TMote Sky* during the entire operation. All information was extracted using a tool called Powertrace, which is a system call of the ContikiOS operating system [30].

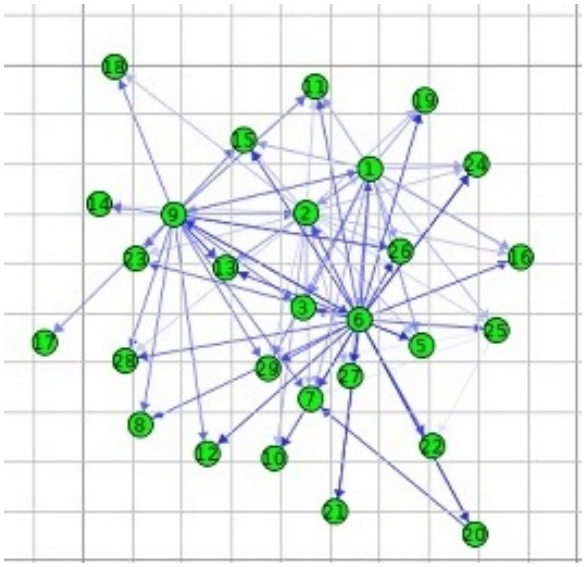

Figure 3: COOJA Nodes Simulation

4.1 Power consumption of sensors

Based on the analysis of the application layer protocols of IoT devices, the average energy consumption depends on four levels: CPU, LPM, TX, and RX [31]. The highest energy consumption during the simulation was undertaken at the RX level, while the lowest consumption was observed at the LPM level. Using the Powertrace, it was possible to monitor the energy flow used in IoT devices and this tool identified the four levels that contributed to the energy consumption of the sensor.

The main equation used to obtain the energy consumption is showed in Equation 1 and was used while the information was collected by the power system in the ContikiOS operating system. The initial cycle of sensor energy consumption is represented by (D) while (y) indicates the type of operation of the IoT device and finally (Ny) represents the number of times the device is in a certain mode of operation. The watt (W) is the unit of power equivalent to one joule (J) per second.

$$D_y = \frac{\sum_{i=1}^{N_y} L_y * V_y * \Delta t_{ty}}{E_{total}} \quad (1)$$

After having the energy consumption value used for each variable at Equation 1, another formula was used to calculate the total energy consumption (in Joule) as shown in Equation 2, as also used in [32]. This equation considers the current voltage product and time interval to calculate the energies of each stage: TX, RX, CPU, and LPM.

4.2 First scenario - Normal

In this scenario, a normal operation was performed between network nodes through message exchange using the UDP protocol. After the execution, a constant behavior was observed proving normality during the operation concerning the power used in miliwatts (mW). The average energy consumption by sensor throughout the experiment is shown in Figure 4.

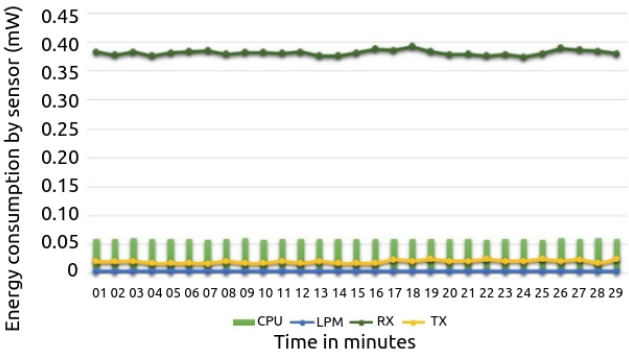

Figure 4: Power energy consumption - normal operation

In the standard error graph, an average standard deviation in the range of 0.0345 in power consumption (mW) was observed during the execution of the normal operation without being under attack and without using the IDS. The observation of behavior occurred by all nodes. The result obtained can be seen in the graph (Standard Error - Normal Consumption) available through Figure 5.

The peak power shown at the beginning of the graph, as showed in Figure 5, represents the initial state of the sensors when they initiate the message exchange in the COOJA simulator. After all the sensors exchange information on the network, it is noticed that there is a decrease in the use of energy until they reach a normal state. The standard error graph shows this disparity in the high exchange of messages between sensors during the start of the simulation.

4.3 Second scenario - under attack

The energy consumption in this simulation was exceedingly high, which indicates the disturbance suffered by the sensors during the attack. Figure 6 shows the average energy consumption by sensor until collapse during the attack and, therefore, there is a high cost in terms of energy during this type of simulation.

$$E_{total}(mJ) = \sum_{\forall i \in D_{LPM}}^{N_{LPM}} I_{LPM} * V_{LPM} * \Delta t_{LPM_i} + \sum_{\forall j \in D_{Tx}}^{N_{Tx}} I_{Tx} * V_{Tx} * \Delta t_{Tx_j} + \sum_{\forall k \in D_{Rx}}^{N_{Rx}} I_{Rx} * V_{Rx} * \Delta t_{Rx_k} + \sum_{\forall z \in D_{CPU}}^{N_{CPU}} I_{CPU} * V_{CPU} * \Delta t_{CPU_z} \quad (2)$$

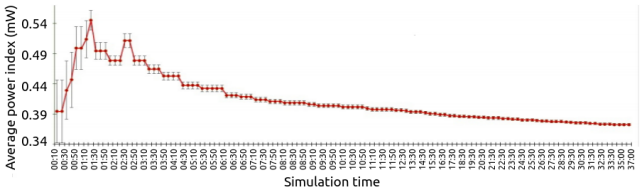

Figure 5: Standard Error - Normal Consumption

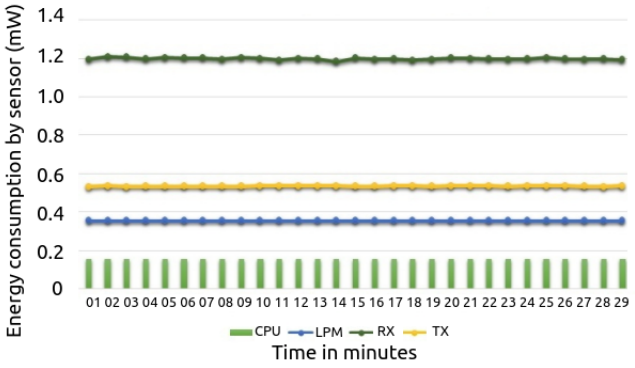

Figure 6: Power energy consumption - under attack

In the standard error graph for the scenario under attack, an average standard deviation in the range of 0.07461 in power consumption (mW) was observed, without using the IDS. The result obtained can be seen in Figure 7. It was observed that the disturbance of the sensors at 11:50, the time of the COOJA simulator, compromised the devices, causing, in turn, the complete shutdown of the sensors. It was noticed, therefore, that after the attack suffered during this time a complete scenario of unavailability was created.

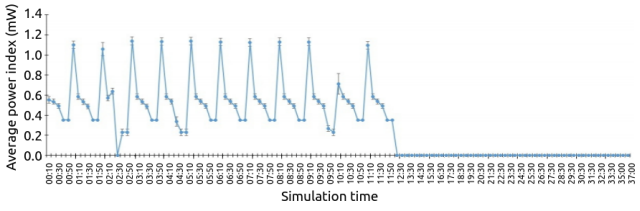

Figure 7: Standard Error - Attack Consumption

4.4 Third scenario - IDS

The developed IDS uses 128-bit AES encryption to create an authentication between sensors. The use of this IDS during operation on the network was also simulated for comparison with the two previous scenarios. The energy consumption obtained during the execution of a normal scenario using the IDS had a very small increase as demonstrated in Figure 8. This small increase demonstrated the efficiency in using the IDS developed using symmetric key cryptography.

The complexity of the proposed IDS algorithm goes back practically only to the Network Scan stage since the sending of the AES message occurs only when an attack appears. The complexity can be observed in the experiments through the resulting energy overhead. From Figure 4 to Figure 8, which depict the normal operation and the operation with IDS, it is noted that the CPU power consumption moved from 0.05 to 0.1 mW, that is, it doubled. Even so, this is not critical, as the overhead consumption of the IDS represented only 25% of the main consumption, that of the antenna in RX.

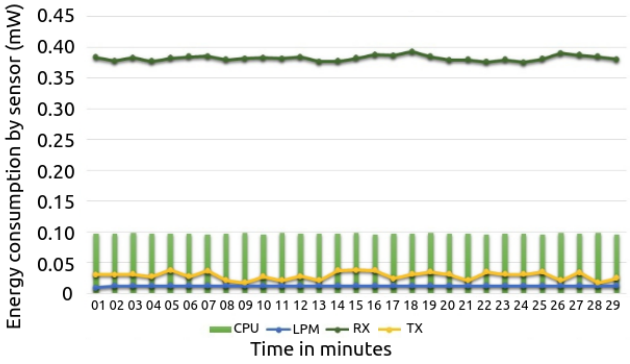

Figure 8: Power energy consumption - IDS

The standard error graph showed an average standard deviation in the range of 0.03846 in power consumption in megawatt during the execution of the IDS in a normal operation. The result obtained can be seen in Figure 9.

The attack algorithms were created aimed at low-power sensors and the creation of an efficient IDS with the ability to detect the attack by reflection in a low-power IoT network. To establish trust between the nodes and prevent the attack, a value calculation was implemented in the IDS to establish a trust relationship between the nodes in the network. The calculation was performed using the security algorithm using AES encryption. The purpose of the calculation was to establish and validate trust relationships between nodes.

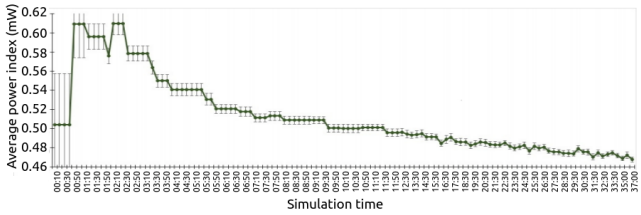

Figure 9: Standard Error - IDS Consumption

Figure 10 shows the average power of the devices (blue line), with peaks the represent the high power consumption in each attack suffered. This variation was demonstrated in peaks that represent the high power consumption in each attack suffered. In turn, the devices not being able to resist the attack stopped responding. Consequently, the power consumption has been reduced to zero for the devices due to their spontaneous drop caused by the malfunction. The sensors were turned off until they were reset. Upon returning to work, they again suffer an attack that consequently turns them off. This process enters a repetitive cycle that inevitably causes all devices on the network to stop working altogether.

In the remaining two lines, represented by the colors orange and green, their normality is observed during the operation. The orange line shows the operation of exchanging messages between IoT devices in a normal scenario, without attacks and also without any protection. However, on the green line, it can be noted that it was slightly above the standard, even in the normal scenario. This increase was due to the presence of the IDS, causing energy consumption to suffer a small increase. This addition is also due to the use by the IDS of the AES encryption algorithm. The purpose of IDS is to notify the network administrator by sending messages at the moment when the reflection attack is verified. Therefore, IDS will make it possible to intervene in the availability of devices. However, human interaction will be necessary to block the attack or to reestablish services after being notified by IDS. After detecting the attack, the IDS has a delay of less than a second in its operation to notify the intrusion.

In the graph of the confidence interval shown in Figure 11, an average standard error of 0.413 was observed for normal operation, of 0.301 during the attack and, finally, of 0.512 when using the intrusion detection system. The simulation in each scenario was performed eight times, although the proximity of the numbers was noted, generating an average from the third simulation. Interpreting the attack indicator line, it is possible to be 95% sure that the variation between the lower and upper limits was between the range of 0.24mW to 0.36mW. This information characterizes the power loss that the devices had during the execution of the simulation.

5 CONCLUSIONS

The emergence of IoT has enabled a considerable increase in attacks on telecommunications and information services. With this increase, information security procedures have emerged through policies, rules of use, and prevention techniques. These techniques can be seen as the use of encryption and the use of intrusion detection systems.

The purpose of using cryptography was to make it possible to preserve one of the three pillars of information security known in the literature as availability. To preserve this pillar of information security, the IDS based on anomaly and host was developed. Due to its detection character, through unwanted behaviors and installed in each sensor, an IDS was implemented using the AES algorithm. To obtain information on energy consumption and the feasibility of using cryptography as an IDS solution, internal and external COOJA resources were used with Operational System Contiki support.

Therefore, it was possible to notice that the traffic and energy consumption presented in the face of a normal operation was compatible with reality. The relation to the security protection provided by the IDS, the consumption was compatible with a low-power network. As consequence, the target about the total energy consumption that a sensor can consume in a low-power network was achieved.

Finally, COOJA is a great simulator that allows cost reduction. It enables the creation of a real scenario, conducive to the research environment and the study of information security, related to low-power devices. By elaborating a set of resources for the sensors and applying them to the COOJA simulator, the solutions can be implemented in a physical and real environment after being exhaustively tested through the simulator on the logical devices.

REFERENCES

[1] Arbor. Internet das coisas permitiu aumento de 60% no tamanho de ataques, aponta relatório anual da arbor. IoT Industrial agita sector de Petróleo e Gás Africano, 2018. URL http://amediaagency.com/iot-industrial-agita-sector-de-petroleo-gas-africano.

[2] Akamai. Memcached udp reflection attacks. By Akamai SIRT Alerts, 2018. URL https://blogs.akamai.com/2018/02/memcached-udp-reflection-attacks.html.

[3] Y. A. Bekeneva and A. V. Shorov. Simulation of drdos-attacks and protection systems against them. In *2017 XX IEEE International Conference on Soft Computing and Measurements (SCM)*, pages 165–167, May 2017. doi: 10.1109/SCM.2017.7970527.

[4] Tarun Kumar Goyal, Vineet Sahula, and Deepak Kumawat. Energy efficient lightweight cryptography algorithms for iot devices. *IETE Journal of Research*, 0 (0):1–14, 2019. doi: 10.1080/03772063.2019.1670103. URL https://doi.org/10.1080/03772063.2019.1670103.

[5] W. Zhao, Y. Ha, and M. Alioto. AES architectures for minimum-energy operation and silicon demonstration in 65nm with lowest energy per encryption. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2349–2352, May 2015. doi: 10.1109/ISCAS.2015.7169155.

[6] Eunsook Kim, Dominik Kaspar, and JP Vasseur. Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). RFC 6568, April 2012. URL https://rfc-editor.org/rfc/rfc6568.txt.

[7] P. Kugler, P. Nordhus, and B. Eskofier. Shimmer, cooja and contiki: A new toolset for the simulation of on-node signal processing algorithms. In *2013 IEEE International Conference on Body Sensor Networks*, pages 1–6, 2013. doi: 10.1109/BSN.2013.6575497.

[8] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with cooja. In *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, pages 641–648, Nov 2006. doi: 10.1109/LCN.2006.322172.

[9] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits. Denial-of-service detection in 6lowpan based internet of things. In *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 600–607, Oct 2013. doi: 10.1109/WiMOB.2013.6673419.

[10] B. Bouyeddou, F. Harrou, Y. Sun, and B. Kadri. Detection of smurf flooding attacks using kullback-leibler-based scheme. In *2018 4th International Conference on Computer and Technology Applications (ICCTA)*, pages 11–15, May 2018. doi: 10.1109/CATA.2018.8398647.

[11] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, New York, NY, USA, 2006. ISBN 0471241954.

[12] Luís M. L. Oliveira, Joel J. P. C. Rodrigues, Amaro F. de Sousa, and Jaime Lloret. Denial of service mitigation approach for ipv6-enabled smart object networks. *Concurrency and Computation: Practice and Experience*, 25(1):129–142, 2013. URL http://dblp.uni-trier.de/db/journals/concurrency/concurrency25.html#

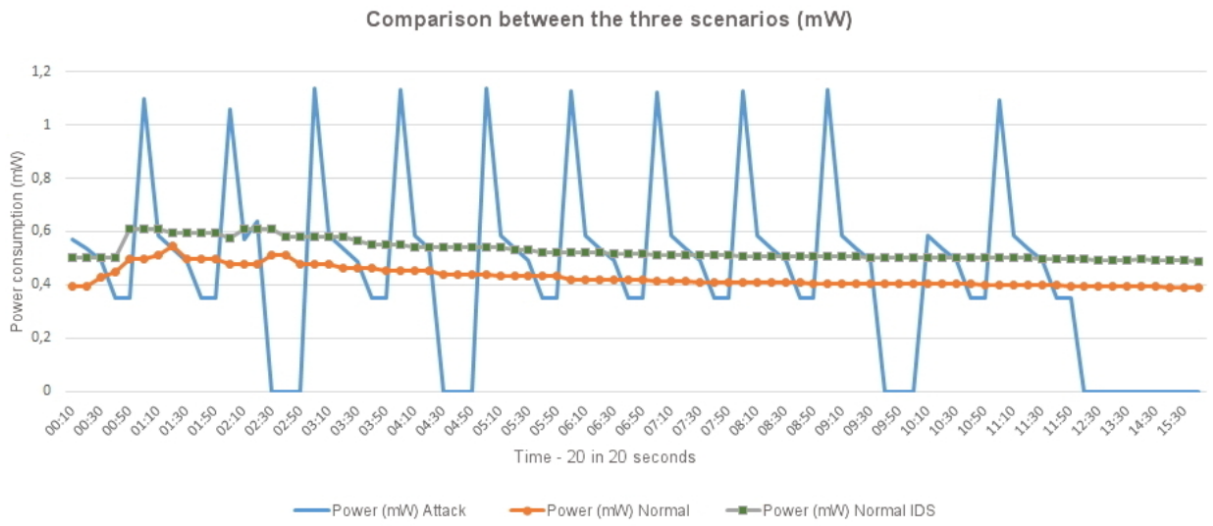

Figure 10: Average power of all sensors in the network

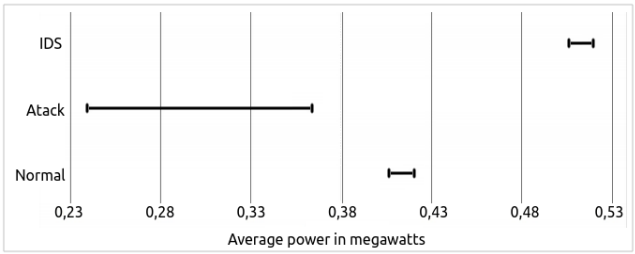

Figure 11: Confidence Interval Chart

OliveiraRSL13.

[13] Christian Cervantes, Michele Nogueira, and Aldri Santos. Mitigação de ataques no roteamento em iot densa e móvel baseada em agrupamento e confiabilidade dos dispositivos. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, may 2018. URL http://ojs.sbc.org.br/index.php/sbrc/article/view/2453.

[14] Kamaldeep, M. Malik, and M. Dutta. Contiki-based mitigation of udp flooding attacks in the internet of things. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 1296–1300, May 2017. doi: 10.1109/CCAA.2017.8229997.

[15] V. Dao, V. Hoang, A. Nguyen, and Q. Le. A compact, low power aes core on 180nm cmos process. In *2016 International Conference on IC Design and Technology (ICICDT)*, pages 1–5, June 2016. doi: 10.1109/ICICDT.2016.7542040.

[16] Siva Ramakrishna Pillutla and Lakshmi Boppana. An area-efficient bit-serial sequential polynomial basis finite field gf(2m) multiplier. *AEU - International Journal of Electronics and Communications*, 114:153017, 2020. ISSN 1434-8411. doi: https://doi.org/10.1016/j.aeue.2019.153017. URL http://www.sciencedirect.com/science/article/pii/S1434841119318485.

[17] Z. Shelby and C. Bormann. *6LoWPAN: The Wireless Embedded Internet*. Wiley, 2009.

[18] J. R. P. de Carvalho, P. M. da Silveira, and S.R. Vieira. Geoestatística na determinação da variabilidade espacial de características químicas do solo sob diferentes preparos. *Pesquisa Agropecuária Brasileira,PAB*, 37(8):1151–1159, September 2002. URL http://ainfo.cnptia.embrapa.br/digital/bitstream/AI-SEDE/23448/1/pab1083.pdf.

[19] D. Wang. Building value in a world of technological change: Data analytics and industry 4.0. *IEEE Engineering Management Review*, 46(1):32–33, Firstquarter 2018. ISSN 1937-4178. doi: 10.1109/EMR.2018.2809915.

[20] M. Sony and S. S. Naik. Ten lessons for managers while implementing industry 4.0. *IEEE Engineering Management Review*, 47(2):45–52, Secondquarter 2019. ISSN 1937-4178. doi: 10.1109/EMR.2019.2913930.

[21] A. Ancarani and C. Di Mauro. Reshoring and industry 4.0: How often do they go together? *IEEE Engineering Management Review*, 46(2):87–96, Secondquarter 2018. ISSN 1937-4178. doi: 10.1109/EMR.2018.2833475.

[22] Chao Lin, Debiao He, Xinyi Huang, Kim-Kwang Raymond Choo, and Athanasios V. Vasilakos. Bsein: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. *Journal of Network and Computer Applications*, 116:42 – 52, 2018. ISSN 1084-8045. doi: https://doi.org/10.1016/j.jnca.2018.05.005. URL http://www.sciencedirect.com/science/article/pii/S1084804518301619.

[23] S. S. Kolahi, Yuqing Cao, and Hong Chen. Evaluation of ipv6 with ipsec in ieee 802.11n wireless lan using fedora 15 operating system. In *2013 IEEE Symposium on Computers and Communications (ISCC)*, pages 000203–000206, July 2013. doi: 10.1109/ISCC.2013.6754946.

[24] Roberto Sandre. Thread and zigbee for home and building automation. Texas Instruments, 2018. URL https://www.ti.com/lit/wp/sway012/sway012.pdf.

[25] Contiki. Contiki: The open source os for the internet of things. Contiki, 2018. URL http://www.contiki-os.org/index.html.

[26] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *29th Annual IEEE International Conference on Local Computer Networks*, pages 455–462, Nov 2004. doi: 10.1109/LCN.2004.38.

[27] Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16 – 24, 2013. ISSN 1084-8045. doi: https://doi.org/10.1016/j.jnca.2012.09.004. URL http://www.sciencedirect.com/science/article/pii/S1084804512001944.

[28] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Fonte:*, 28(1):18 – 28, 2009. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2008.08.003. URL http://www.sciencedirect.com/science/article/pii/S0167404808000692.

[29] Dit-Yan Yeung and Yuxin Ding. Host-based intrusion detection using dynamic and static behavioral models. *Pattern Recognition*, 36(1):229 – 243, 2003. ISSN 0031-3203. doi: https://doi.org/10.1016/S0031-3203(02)00026-2. URL http://www.sciencedirect.com/science/article/pii/S0031320302000262.

[30] Adam Dunkels, Joakim Eriksson, Niclas Finne, and Nicolas Tsiftes. Powertrace: Network-level power profiling for low-power wireless networks. Technical Report 2011:05, SICS, 2011.

[31] Georgi Stojanov and Andrea Kulakov. *ICT Innovations 2016: Cognitive Functions and Next Generation ICT Systems*. 01 2018. ISBN 978-3-319-68854-1. doi: 10.1007/978-3-319-68855-8.

[32] Ashutosh Bandekar, Akshay Kotian, and Ahmad Y. Javaid. Comparative analysis of simulation and real-world energy consumption for battery-life estimation of low-power iot (internet of things) deployment in varying environmental conditions using zolertia z1 motes. In Michele Magno, Fabien Ferrero, and Vedran Bilas, editors, *Sensor Systems and Software*, pages 137–148, Cham, 2017. Springer International Publishing. ISBN 978-3-319-61563-9.