CONTEXT-AWARE SECURITY MECHANISM FOR MOBILE CLOUD COMPUTING

Cláudio Aroucha[1] and Higo Felipe Pires[2]

[1]Computer Engineering Coordination,
Federal University of Maranhão, São Luís, Maranhão, Brazil
[2]Postgraduate Program in Electrical Engineering,
Federal University of Maranhão, São Luís, Maranhão, Brazil

ABSTRACT

The use of mobile devices is common among people and something essential these days. These devices have limited resources which makes it critical to provide security without compromising user ergonomics, given the large number of cyberattacks that occur. This work proposes a context-aware security mechanism for Mobile Cloud Computing providing a security level of device data privacy from the analysis of the attributes of the network that is connected, available level of RAM, CPU, and battery at the time of communication. of data with the cloud. In addition, Transport Layer Security (TLS) technology is used to create a secure channel for sending data between the client and the server and implement the analysis of the mobile device context using Fuzzy logic. The impact of the proposed mechanism on mobile device performance was measured through stress tests. The proposed mechanism had a superior performance of 38% in the number of executions, 10% of memory, and 0.6% of CPU about the use of a single type of pre-defined symmetric algorithm for private network environment.

KEYWORDS

Cloud Computing, Context-aware Computing, Network Security, Mobile Computing.

1. INTRODUCTION

The emergence of mobile devices, as well as their great acceptance and demand by the population, have awakened a need to develop applications that satisfy the needs of their users. Methods of data processing, low RAM (random access memory), limited battery, and scarce storage limit [1].

To circumvent this limitation, we used Cloud Computing [2] [3], which provides computing, software, and storage services that do not require the knowledge from the final user about the physical location and configuration of the system, providing a reduction in computational costs from the mobile device.

The application on the mobile device uses the server inside of the Cloud Computing environment to resolve the storage limit it has. To ensure that data traffic between an application and the server is done securely, use a sequence and an authentication. Encryption is used to encrypt messages sent from the mobile device to the Cloud server, ensuring confidentiality and authentication in your communication.

The data's security of mobile and conventional devices, through a single encryption algorithm. In this way, use a mechanism with different algorithms that provide different levels of security and adaptation to different contexts and devices. Adaptive mechanisms must constantly evolve to the changes in contexts that occur in the mobile device, information that is identified from the Fuzzy logic would increase the accuracy of its current state. Since the complexity of each encryption algorithm is tied to its resource consumption, a mechanism that provides savings is needed. In this context, a better analysis of the state of the device can allow a dynamic adaptation of its level of security and a way of increasing performance.

This paper proposes a context-aware security mechanism for mobile cloud computing. The proposed mechanism can provide an efficient optimization of computational resources for mobile devices, as well as to adapt the level of security in-context information for communication with a storage server running in the Cloud. The remainder of this paper is organized below. Section 2 information on work related to the mechanism. Section 3 explains context-aware computing and its types. In section 4 mobile cloud computing is defined. Section 5 describes our schema, a context-aware security mechanism for mobile cloud computing (MSCC). The evaluation of the mechanism is proposed in section 6. Finally, the conclusion is given in section 7.

2. Related Works

Many context security schemes work like the scheme proposed by [4], focuses to maximize the performance of mobile devices in terms of computing resources without degrading their level of security, based on sensitivity to the context. Provides the user with high convenience and resource-saving features. This scheme does not deal with adaptation to various cryptographic algorithms and security automation in the evaluation phase being proposed for future work. The work proposed by Lee [5], is a security model for social networking services for smartphones. This article suggests a context-aware authentication and access control system via scenarios that the smartphone finds. The Fuzzy algorithm is used to analyze the smartphone information acquired by user authentication. The information collected is ambiguous so that it can be defined quantitatively, only be estimated. If Fuzzy logic were not used, several rules could be generated to define whether performance is good or not, making it difficult to apply the level of security that should be used. In [6], CASE is a context-aware attribute learning scheme, which aims to keep private the user data in a unified intelligent environment called Society 5.0 using CP-ABE. The CP-ABE security system obtains data manually, informed by the user, and is intended to automate it, which will provide privacy and context accuracy, since the user may incorrectly inform some context-sensitive data. After the performance analysis, it was found that among the attribute learning schemes based on the SVM, decision tree (DT), and Naive Bayes (NB) models, the one that found the best prediction of time, clock, and precision was the DT.

In [7] a scheme called Symmetric Key Management Scheme (SKMS) is proposed, which encrypts data by a symmetric key, focusing on the protection of Wireless Sensor Networks (WSNs), since the sensors have low computational power and are more vulnerable to attacks, choose this type of encryption where it uses the hash and XOR function to protect them. As a result, it obtained safety and low energy cost on the side of the sensors.

2.1. Comparative Analysis

As described earlier, there are numerous deployment techniques and results to provide context-aware security for mobile devices or sensor, based on the client/server architecture. Table 1 presents the characteristics that the context-aware security mechanism for mobile cloud computing (MSCC) has concerning the other adaptive security mechanisms mentioned at the beginning of section 2.

The characteristics of the MSCC are dynamic on-demand adaptation, and security level based on the types of network environments (public, private, and domestic) that the mobile device is connected to, collecting context information at the time the sending or receiving request is processed of a file. The context analysis is based in model DT represented on Fuzzy logic to present the state of the mobile device at the time of the request, and performance evaluation (CPU, memory, and battery) of the cryptographic algorithms in use in the real device to be used as a criterion in the choice of algorithm.

The contributions of each author influenced the development of this work (context-aware security techniques proposed for devices when sending data to the Cloud Computing environment) since in all the works there is clear importance in saving the resources of the mobile devices when providing safety.

Table 1. List of works related to the context-aware security mechanism for mobile cloud computing

P.A	**D.A**	**N.S**	**C.I**	**Fuzzy**	**E.C.A**	**TLS**	**C.C**	**R.D**
G. An [4]	X	-	X	-	-	-	-	X
Lee [5]	X	X	X	X	-	-	-	X
Ghosh [6]	X	X	X	-	-	-	X	X
Al-taha [7]	-	X	X	-	X	-	-	X
CASMCC	X	X	X	X	X	X	X	X

- P.A - Proposed Approaches D.A - Dynamic Adaptation N.S - Security Level.
- C.I - Fuzzy Information Collection - Fuzzy-Based Information Analysis E.C.A - Evaluation of Cryptographic Algorithms.
- TLS - Transport Layer Security - CC - Cloud Computing - R.M.D - Real Devices.
- CASMCC – Context-Aware Security Mechanism for Mobile Cloud Computing.

3. CONTEXT-AWARE COMPUTING

According to [8], the context is first any information that can characterize an entity, which this entity can be a person, place, physical or computational object. Context-aware computing or context-sensitive computing is the use of context information in relevant user tasks. The "five W's" define context-aware computing:

- Who: Current systems identify the iterations of a particular user, rarely incorporating other information from the environment.
- What: Understanding and interpreting human activity is a difficult problem.
- Where: This context component has been used a lot with "When". Some tour guide systems learn from the history of the movements (tourist spots already visited by the tourist).
- When: Time is often used to index records or determine how long a person stayed in a place, most context-oriented applications are not aware of the passage of time
- Why: More challenging than knowing "What" a person is doing is understanding the "Why".

4. Context-Aware Security Mechanism for Mobile Cloud Computing

This section will introduce the mobile cloud context-aware security mechanism and a cryptographic algorithm evaluator for a mobile device to provide secure and adaptive communication to the cloud server.

The purpose of the context-aware security mechanism for mobile cloud computing proposed here is to identify the context of the mobile device and adapt the security degree at the time of sending data to the server in the cloud to provide a saving of the resources of the mobile device. This ensures the safety and ergonomics of the mobile device. For this we have that:

- Evaluate the performance of cryptographic algorithms for the mobile device in use;
- Represent the context of the mobile device using Fuzzy logic;
- Make decisions automatically for the adaptation of employee safety levels.

4.1. Server and Client Engine Architecture

The server engine was designed using the JAVA SECURE SOCKET EXTENSION (JSSE). It runs on the EUCALYPTUS infrastructure environment. Figure 1 shows the Server architecture divided into:

- Application: component related to the server application interface and routines to run the other components;
- InitSSL: component is responsible for initiating the server-side TLS configuration parameters, such as loading the materials of the Keymanager and Trustmanager classes, performing the handshake (connection establishment), and sending the files through a secure channel;
- ConfigClient: negotiates which type of symmetric encryption was chosen by the client so that it can initiate decryption or data encryption.

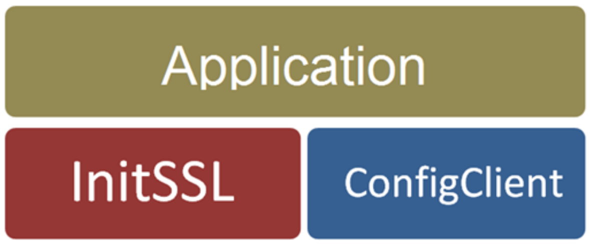


Figure 1. Proposed architecture for the server mechanism.

The client mechanism was created for the Android platform [9], it has the functions of evaluating the cryptographic algorithms regarding the use of mobile device resources, analyzing the context information representing them with Fuzzy logic, and making the decision to use the TLS encryption algorithm that best suited for the situation the mobile device is in Figure 2 shows the components of the client engine architecture, which are:

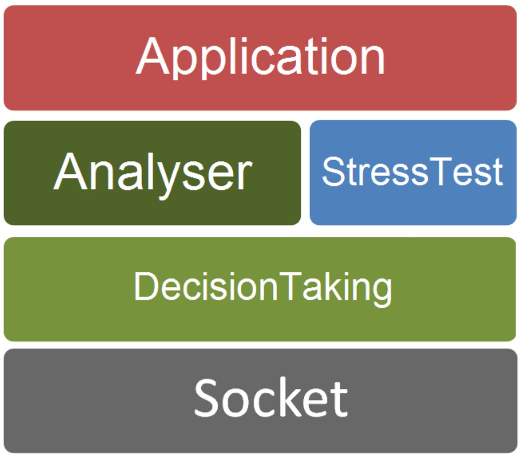
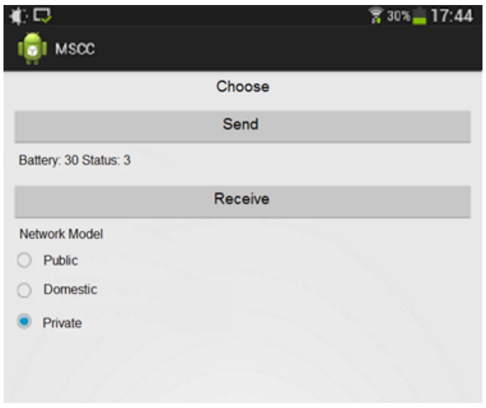


Figure 2. Proposed architecture for the client mechanism

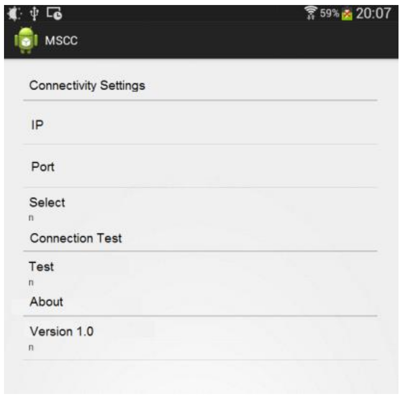
- Application: related to the client application interface. Figure 3 is the main screen of the client application where you have the send or receive buttons previously selected in the configuration, in addition to the network environments for the user to inform which the mobile device is connected. Figure 4 shows the settings to be completed before sending or receiving files such as IP, port, the file to be selected, either send or receive, and the test button to verify that the server is active.



Battery: 30 Status: 3

Network Model
Public
Domestic
[✓ Private]

Figure 3. Client application settings screen



Version 1.0

Figure 4. Client application main screen

- Analyser: The Analyser acts on the context information of the mobile device and represents it in Fuzzy logic. The Analyzer process is presented in figure 5. The context information is inserted to be transformed into linguistic terms from the established intervals, it performs the implication of the linguistic terms in the defined Fuzzy rules, to identify which rule it has the highest degree of relevance.

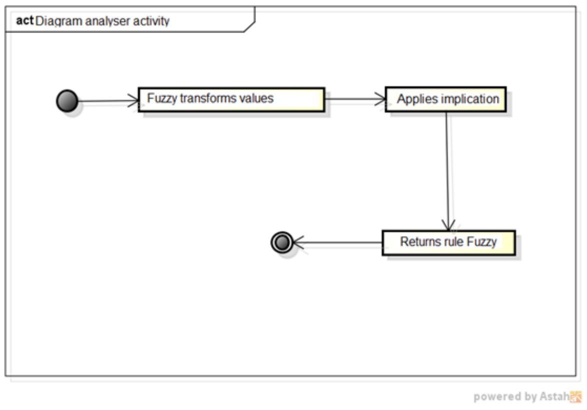


Figure 5. Analyzer activity diagram

This degree can be identified in figure 6, which has the membership variation or degree of pertinence, that the closer the value of x is to 1, the more pertinent to that state the variable is. For example, in Figure 6, it is found that the battery is the only context information that has a closed range for use at 100%. Inferences of CPU and memory resources are performed in the same way as the battery.

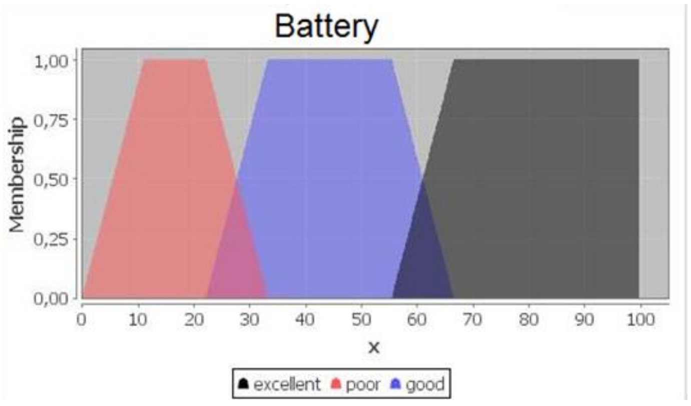


Figure 6. Battery levels intervals

The Fuzzy rules are defined according to figure 7. The characteristics that determine the rules are the battery (defined by the terms Poor, Good and Excellent), CPU (Critical, Medium, and Comfort), and memory (Low, Half, and Full), which represent a combination between them, forming the 27 implemented rules, in the figure are shown rules of the worst and best case. These rules listed in 27 describe the possible states that the mobile device may come up with.

RULE 1: IF BATTERY IS POOR AND CPU IS CRITICAL AND MEMORY IS LOW THEN status IS DOWN;
RULE 27: IF BATTERY IS EXCELLENT AND CPU IS CONFORT AND memory IS FULL THEN status IS UP;

Figure 7. Rules Fuzzy

- Stress Test: responsible for generating the profile of the cryptographic algorithms for the mobile device. Figure 8 shows the process of stressing the cryptographic algorithms when selecting the file to be sent to the server. During the sending, context information such as the number of rounds (rounds or tests), memory usage, and free CPU. This information will be saved in the profile of that algorithm until the device's battery level decreases by 5%.

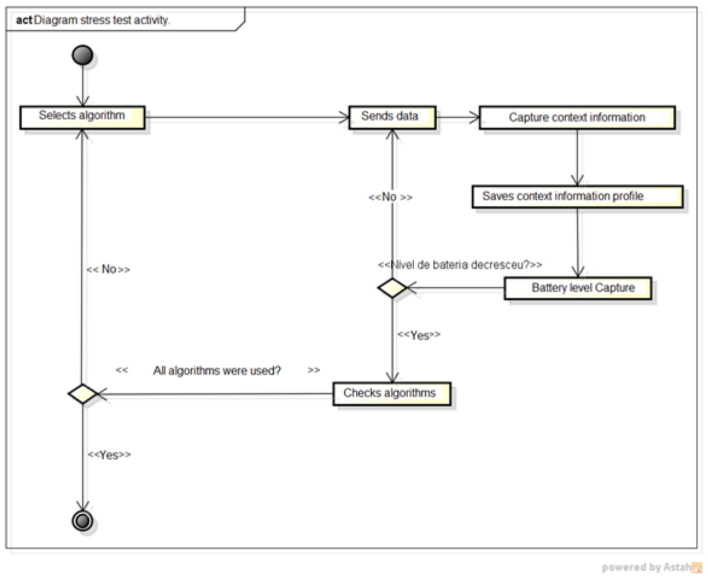

act Diagram stress test activity.
Selects algorithm
Sends data
Capture context information
Saves context information profile
<<No >>
<< No>>
<<Nivel de bateria decresceu?>>
Battery level Capture
<<Yes>>
<< All algorithms were used? >>
Checks algorithms
<<Yes>>
powered by Astah

Figure 8. Activity diagram of Stress Test

- Decision Taking: responsible for scoring the cryptographic algorithms taking into account the profile generated by the Stress Test component and the context analysis, to choose the most appropriate. The decision-making process is presented in Figure 9, which illustrates the numerical profile of the tests of each algorithm, transforming them into linguistic terms, saving in a file called the linguistic profile. Context information is captured to be used as input parameters for the Analyzer component, which, when performing its process, returns the rule with the highest degree of pertinence. From this rule the context information present in it is punctuated, for each profile of the algorithm.

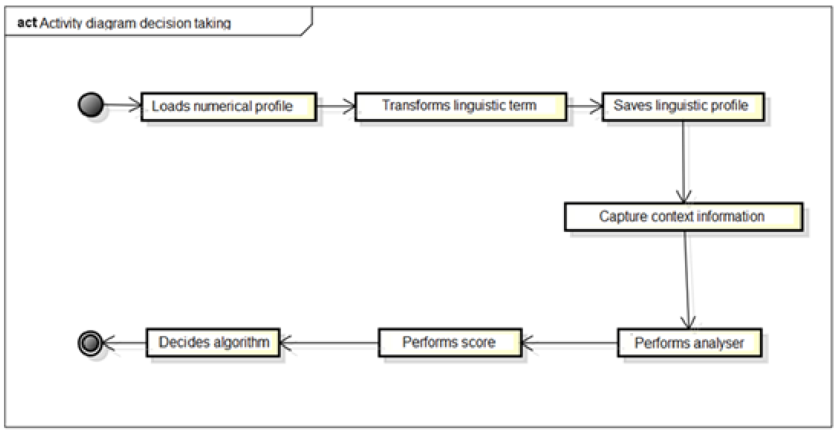


Figure 9. Activity diagram Decision Taking

For example, in the case of the CPU, if the context profile analyzed for the CPU is "CONFORT" the cryptographic algorithm that has the profile corresponding to the context points 1, if "MEDIUM" scores 0.5 and if it is "CRITICAL" it also scores 0.5. If the profile of the analyzed context for CPU is "MEDIUM" the cryptographic algorithm that has the "CONFORT" profile scores 0 because it is above the analyzed context and this algorithm consumes more than the current device's capacity, if it is " MEDIUM "score 1 and if it is" CRITICAL "score 0.5. The last state of context that the CPU could take would be "CRITICAL" if the profile of the cryptographic algorithm is "CONFORT" it does not score, "MEDIUM" also does not score because it is above the analyzed context and what has "CRITICAL".

- Socket: is responsible for initializing the TLS configuration to establish the secure channel between the client and server with the information of which type of encryption will be used.

5. RESULTS

To evaluate the context-aware mechanism concerning resource economics, battery stress in 80% segmented in 8 parts as said before, running in a controlled environment, in which only the Android operating system and its primitives would be working, besides the developed client application.

For the evaluation of the behavior of the proposed mechanism, it remained the same used by [4], which uses the CPU occupation, memory, and battery level spent time information of the device that will be used to mount the profile of cryptographic algorithms as well as to demonstrate the use of context-aware security mechanism and static security configuration.

5.1. Tests to Assemble the Profile of each Symmetric Algorithm

Figure 10 shows the graph with the number of 2MB file send rounds (iterations) over a TLS secure channel up to the 5% consumption of the battery level for each of the three symmetric algorithms, AES_256, AES_128, and RC4, being executed on the actual mobile device for the creation of the algorithmic profile.

It is verified that the symmetric algorithm RC4 is the most economical to perform more rounds than the others at the cost of 5%. The connection establishment is done once for each algorithm, with the connection continuing until the consumption of 5% of the battery, making a new connection establishment for the remaining cryptographic algorithms.

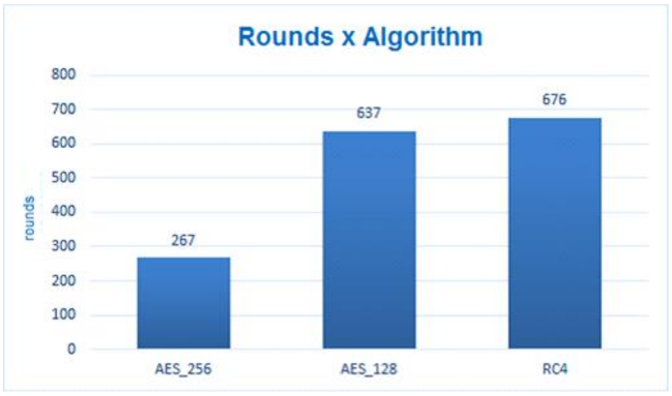


Figure 10. Graphic Rounds x Algorithm.

In Figure 11, the average memory occupation to send the data through the secure channel used by each algorithm is demonstrated. This graphic shows AES_256 as the one that takes up the most memory to accomplish this task. This type of information is important to assist in the decision-making of the context-aware security mechanism.

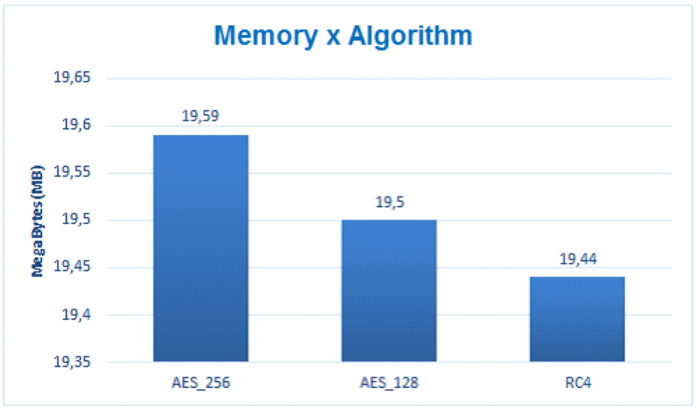


Figure 11. Graphic Memory x Algorithm

In Figure 12, AES_256 has the longest CPU allocation time to perform the task of sending data through a secure channel, since it works with a 256-bit key that demands a greater effort for calculations than 128.

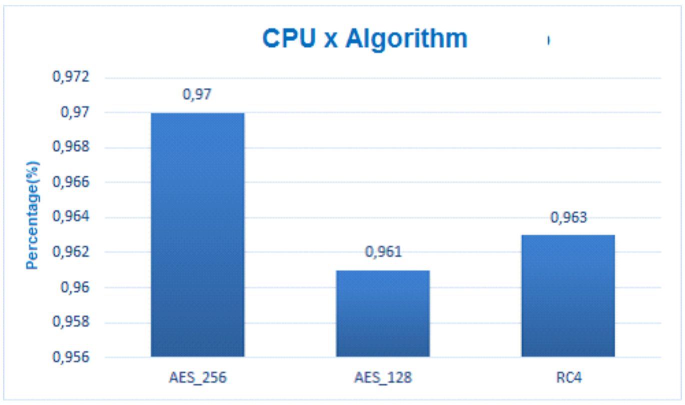


Figure 12. Graphic CPU x Algorithm

The objective of these tests is to create a profile of the symmetric algorithms for the mobile device in the first execution of the application, thus identifying which algorithms consume or allocate more resources for the task of sending data through a secure channel. This profile will only be re-created if it is deleted by the user.

5.2. Context - Aware Security Mechanism Tests X Static Security Configuration

The context-aware security mechanism tests the network environment (public, private, and home), and the static configuration of security, using a single symmetric algorithm, is explained in Figure 13. This graph demonstrates the number of rounds (iterations) for the task of sending data from the mobile client to the server, similar to the test to set up the algorithmic profile, the difference being that to have a greater precision of the results there was the stress of 80% of the battery level of the mobile device charged to 100 %. This is to make sure that the decision-making will go through all the states of the battery and will not assume the critical state that the Android operating system firm in less than 20% in most mobile devices, a fact verified by tests.

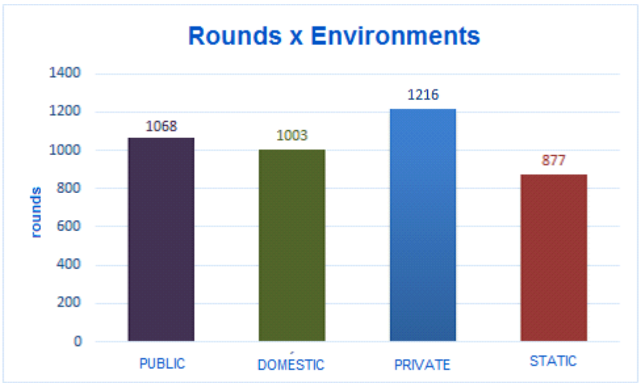


Figure 13. Graphic Rounds x Environments

The test was performed for each network environment and these crusaders with the static to demonstrate how much economy the mechanism could provide in relation to the use of a fixed (static) symmetric algorithm. In this graph, we verified that using a fixed symmetric algorithm, the AES_256, has a number of rounds lower than the public, home and private network environments of the context-aware security mechanism, stating that the mechanism works more at the same battery cost.

Figure 14 shows the behavior of the security mechanism in the private network environment with the lowest average memory allocation to perform the data sending task. It is expected that the AES_128 and RC4 algorithms, as previously seen, occupy less memory for your tasks.

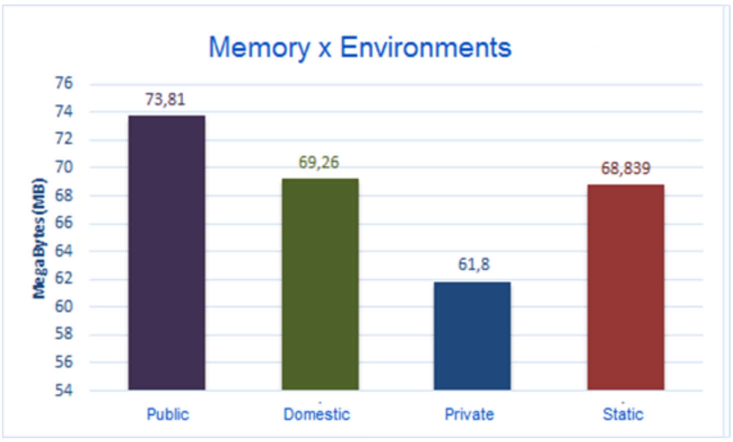


Figure 14. Graphic Memory x Environments

In Figure 15, the average CPU time allocated to the mechanism in the public and static network environment are the highest, since among all the network environments the public is the most vulnerable, it is verified that AES_256 allocates more time CPU reflecting on its average for the mechanism in this environment. The others are very low realizing that the changing mechanism of network environments may have a significant improvement in CPU allocation.

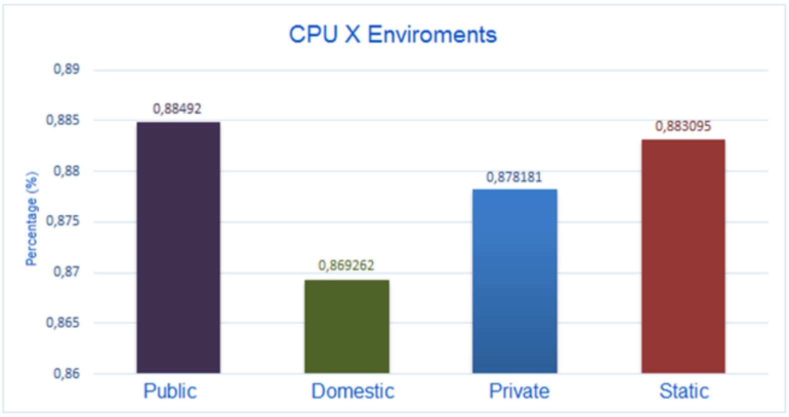


Figure 15. Graphic CPU x Environments

The table 2 shows the values of the resources consumed based on the graphs previously displayed.

Table 2. Table with the values of each resource consumed for static and dynamic configuration.

Measurements	**Config. Static Seg.**	**Context-aware Security Mechanism**		
		private	**domestic**	**public**
Rounds (nº)	877	1216	1003	1068
CPU (%)	0,883	0,8781	0,8692	0,8849
Memory (MB)	68,83	61,8	69,26	73,81

Table 3 summarizes the savings and expenses that the context-aware security mechanism has concerning the static security configuration.

Table 3. Savings table or resource spend by using the context-aware security mechanism for mobile cloud computing versus static configuration.

	Consumption (%)		
	private	**domestic**	**public**
Rounds	-38,65%	-14,37%	-21,78%
CPU	-0,66%	-1,57%	0,21%
Memory	-10,33%	0,61%	7,22%

The negative values are shown in the table identify the economy of the resource by the use of a security mechanism aware of the context concerning the static configuration, positive values represent the expenses that the mechanism had the most concerning the static.

Tests were performed in laboratory-based environments demonstrating the consumption and average allocation of the battery, memory, and CPU resources of the actual mobile device, from the use of the mechanism for each connected network environment. These values were crossed with those of the static security configuration to evidence the economy of resources by the mechanism.

6. CONCLUSION

This work has contributed to the context-aware security mechanism for mobile cloud computing, with TLS-based security associated with Fuzzy context analysis for Cloud computing, to better identify the contexts and guarantee the confidentiality and authentication economically. In addition to a cryptographic algorithm evaluator for mobile devices, determining the average resource spend has also been developed. The mechanism was implemented in layers, allowing greater interaction between the algorithm evaluator and the context analyzer. In addition, it is possible to update the scoring system without interfering with the assembly of the algorithmic profile.

Tests were performed in laboratory-based environments demonstrating the consumption and average allocation of the battery, memory, and CPU resources of the actual mobile device, from the use of the mechanism for each connected network environment. These values were crossed with those of the static security configuration to evidence the economy of resources by the mechanism.

Finally, a functional prototype of the proposed mechanism was obtained. The main contribution is to adapt the degree of security to the context, with the ability to collect data in real time and make the decision of the best symmetric algorithm to provide security without impacting the ergonomics of the mobile device. The results obtained in the tests are considered satisfactory, considering that the performance presented by the mechanism with dynamic adaptation concerning the static one was superior.

The limitation of the mechanism is that it works by downloading and uploading files to the cloud server from a persistent network connection, so if there is a change of network connection during the sending of data, the process must be restarted. In addition to data, encryption is applied to each request for downloading and uploading files, persistently until the end of the transfer, there is a change of context of the mobile device that will not be detected since the decision is only made at the beginning of the transfer and not during.

REFERENCES

[1] M. Satyanarayanan. Pervasive computing: Vision and challenges. Personal Communications, IEEE, 8(4):10–17, 2001.

[2] M. A. AlZain, B. Soh, and E. Pardede. A new model to ensure security in cloud computing services. Journal of Service Science Research, 4(1):49–70, 2012.

[3] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian. Mobile cloud computing: A survey, state of art and future directions. Mobile Networks and Applications, 19(2):133–143, 2014.

[4] G. An, G. Bae, K. Kim, and D. Seo. Context-aware dynamic security configuration for mobile communication device. In New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on, pages 1–5. IEEE, 2009.

[5] H. Lee and M. Chung. Context-aware security model for social network service. In Broadband and Wireless Computing, Communication and Applications (BWCCA), 2011 International Conference on, pages 144–151. IEEE, 2011.

[6] Ghosh, Timam, et al. "CASE: A context-aware security scheme for preserving data privacy in IoT-enabled society 5.0." IEEE Internet of Things Journal (2021).

[7] Al-taha, Mohammed A. "Symmetric Key Management Scheme for Hierarchical Wireless Sensor Networks." International Journal of Network Security & Its Applications (IJNSA) Vol 10 (2018).

[8] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In Handheld and ubiquitous computing, pages 304–307. Springer, 1999.

[9] (2015, January) Welcome to the Android open source project. Google Inc. [Online]. Available: http://source.android.com/

AUTHORS

Cláudio Manoel Pereira Aroucha received the B.S. and M.Sc. degree in Computer Science from the Federal University of Maranhão (Brazil in 2012) and the Federal University of Maranhão (Brazil in 2015) respectively. He is now a professor of Computer Engineering at the Federal University of Maranhão (UFMA) in Brazil. His research interests include distributed systems, Network Security, and Cloud Computing.

Higo Felipe Silva Pires received the B.S. and M.Sc. degree in Computer Science from the Federal University of Maranhão (Brazil in 2014) and the Federal University of Maranhão (Brazil in 2017) respectively. He is now a professor of Information Technology at the Federal Institute of Education, Science, and Technology of Maranhão (IFMA) in Brazil. His research interests include Artificial Intelligence and Network Security.