Journal of Internet Technology Volume 18 No.7
DOI: 10.6138/JIT.2017.18.7.20141221

Preprint Paper

airiti

A Secure Spontaneous Mobile Ad Hoc Cloud Computing Network

Sandra Sendra[1], Raquel Lacuesta[2], Jaime Lloret[1], Elsa Macias-López[3]
[1]Universidad Politécnica de Valencia. Camino Vera s/n, 46022, Valencia, Spain
[2]Universidad de Zaragoza, Pedro Cerbuna, 12, 50018. Zaragoza, Spain
[3]Departamento de Ingeniería Telemática, Universidad de Las Palmas de Gran Canaria. Edificio de Electrónica y Telecomunicación, 35017. Las Palmas de Gran Canaria, Spain
sansenco@posgrado.upv.es, raquellacuesta@gmail.com, jlloret@dcom.upv.es, elsa.macias@ulpgc.es

Abstract

Spontaneous ad hoc cloud computing networks let us perform complex tasks in a distributed manner by sharing computing resources. This kind of infrastructure is based on mobile devices with limited processing and storage capacity. Nodes with more processing capacity and energy in a spontaneous network store data or perform computing tasks in order to increase the whole computing and storage capacity. However, these networks can also present some problems of security and data vulnerability. In this paper, we present a secure spontaneous mobile ad hoc cloud computing network to make estimations using several information sources. The application is able to create users and manage encryption methods to protect the data sent through the network. The proposal has been simulated in several scenarios. The results show that the network performance depends mainly on the network size and nodes mobility.

Keywords: Spontaneous networks, Secure protocol, Mobile cloud computing.

Preprint

1 Introduction

The rapid development of the processing and storage technologies and the appearance of many Internet services have brought us cheaper, more powerful and more accessible computing resources. These enhances in technology have enabled the development of a new computing model called cloud computing. Cloud compunting provides several features that make it attractive to business owners [1], such as no up-front investment, decreasing the operating cost as well as reducing business risks and maintenance expenses while the networks are becoming in scalable architectures with easy access. These factors potentiate the development of many types of cloud computing systems [2]. The major service models of cloud computing are known as software as a service (SaaS), platform as a service (PaaS), infrastructure as a service (IaaS), and security as a service (SECaaS).

The evolution of smart mobile devices has facilitated to go a step beyond the simple cloud computing network. A mobile ad hoc cloud computing network allows mobile users to share computing resources and applications. The mobile cloud is built in the ad hoc network and is able to work without having access to outside servers [3]. Moreover, mobile devices connected to the cloud computing and belonging to the spontaneous ad hoc network let other mobile device to access the mobile cloud computing services. This type of temporary network is created spontaneously during a period of time to perform a very specific task or function [4][5]. The main features of spontaneous networks are the following ones [6]:

- Network boundaries are poorly defined.
- The network is not planned.
- Hosts are not preconfigured.
- There are not any central servers.
- Users are not experts.

The management of a spontaneous mobile cloud computing network should take into account the mobility of the nodes, the dynamic feature of the network and the knowledge of the user. It is important to control the formation processes and the network communication to let it work autonomously. Each node must have the necessary technology to connect and access to the network services with minimal interaction. The system design, the data management and the secure transactions are performed by the network nodes. In addition, the frequent changes in the network topology, because of the nodes' mobility, which change the connections between nodes, any safety solution with static configuration is not suitable. Security mechanisms must be adapted to these changes [7]. Moreover, the security is a major issue in cloud-basded dynamic systems [8].

Considering all of these factors, we propose a secure mobile application as a service to make estimations using several information sources. The software works under a spontaneous mobile ad-hoc cloud computing network which performs higher processing capacity than a simple ad hoc network.

airiti

The estimation is done considering different online information sources such as related news, previous results, and current status, among others. Our proposal allows users to be registered as cloud members. This temporary network allows the user to have higher processing capacity in order to compute the desired information. All data sent and received is encrypted to ensure the network anonymity due to the use of network session keys. The proposal has been simulated in different scenarios considering various parameters such as network size and node mobility.

The rest of the paper is structured as follows. Section 2 shows previous works related to the creation of spontaneous ad hoc networks. The proposed spontaneous mobile ad hoc cloud computing network and its security are presented in Section 3. Section 4 explains the test bench carried out to simulate our system and check the correct operation. Finally, conclusion and future workare presented in Section 5.

2 Related Work

Integration between mobile devices and cloud computing is shown in several published works, but there are no proposals where mobile devices build a spontaneous network focused on cloud computing.

There are lots of papers where authors address topics related to mobile cloud systems. For example, J. H. Christensen [9] presented the general requirements and key technologies to achieve the vision of mobile cloud computing. He also analyzed the features of the smart phones, context awareness, cloud and restful based web services. He also explained how these components can interact to create the best experience for mobile phone users. R. Buyya et al. [10] presented several cloud platforms and their characteristics, covering some parts of the state-of-the-art of this topic. In addition, they presented an architecture for market-oriented resources allocation within clouds. It is a global cloud which allows the exchange of services. Their proposal was based on a meta-negotiation infrastructure to establish global cloud exchanges and markets, and they illustrate a case study of harnessing ‘Storage Clouds’ for high performance content delivery. G. Huerta-Canepa and D. Lee [11] presented the guidelines to create virtual ad hoc cloud computing providers. Authors proposed to take profit of the pervasiveness of the mobile devices by creating a cloud among the devices and allowing them to execute tasks. The proposal was able to detect nearby nodes that were in a stable status or following the same movement pattern. In [12], the authors proposed a new cloud computing model for VANET called VANETCloud, which is based on two sub-models: permanent and temporary clouds.

Some security solutions have been proposed for cloud computing. D. Zissis and D. Lekkas [13] proposed a Trusted Third Party system focused on enssuring specific security characteristics within a cloud environment. The proposed solution used cryptography, specifically Public Key Infrastructure operating with SSO and LDAP, to ensure the authentication, integrity and confidentiality of involved data and communication process. In [14], authors attempted to design and simulate a mechanism to calculate trustworthiness of service providers based on their compliance to promised Service Level Agreement parameters. The approach required the installation of monitoring services in the user end devices in order to monitor QoS parameters. In [15], authors proposed the TimePRE scheme to achieve fine-grained access control and scalable user revocation in a cloud environment. The main problem with this scheme was that it required the effective time periods to be the same for all attributes associated with a user. N. Tirthani and R. Ganesan [16] analyzed the security issues faced by user’s private data in the cloud system and the need to find a solution to the problem. They proposed an architecture to ensure the users’ authenticity, which can be implemented in cloud environment. It takes the advantages of linear cryptography for establishing a secure connection and exponential cryptography for encrypting the data. The two algorithms used are Diffie Hellman Key Exchange and Elliptical Curve Cryptography respectively.

Preprint

Finally, some authors of this paper proposed two secure spontaneous wireless ad-hoc network protocols for wireless mesh clients based on the computational costs in [17]. The proposed protocols were based on the trust of the humans that are using the devices through a distributed key management scheme. Both protocols provide node’s authentication, intermediate node authentication, integrity checking, random checking, verification distribution and removement of packets with errors. Authors also presented a secure protocol for spontaneous wireless ad hoc networks which used a hybrid symmetric/asymmetric scheme to exchange the initial data and to exchange the secret keys used to encrypt the data [18]. Finally, authors proposed the use of this kind of secure spontaneous ad-hoc network to grant a quick, easy and secure access to the users to surf the Web [6].

airiti

3 Secure Architecture of Mobile Ad Hoc Cloud

This section describes the proposed architecture and its design, the flow diagram when receiving and sending data, the pseudocode, and the application.

When a user requires the use of that cloud network to compute the data available on Internet such as past results or published news, the system can process these data to estimate a specific issue. In order to make more comprehensive our proposal, we include the example of sport bets. Considering the current state of betting on the sport and other parameter related to this bet, the system could generate a number of listings of risks and even the quantities that a user could win at that time. The process to calculate the optimal solution of a single bet can be relatively simple. When several sports and kind of bets are considered, the estimation of the results could be very complicated. Simultaneous estimation of football matches and horse rice bets is an example.

The development of our secure application enables us to maintain the network anonymity and the delivered information. We are not developing the application to process the data. The paper presents the secure access system and data encryption for requesting the cloud information or service.

Two parts must be secured in a system: the authentication [19] and the communication [20]. Our system is composed by two levels. The lowest level is the network level and it is in charge of providing security to the network. The upper level is the cloud computing level which provides the SaaS accessed by users using a light client via web browser. We have added the security is both, (1) when a point-to-point connection is created between the node that wants to join to the network and an existing node through which the new node will be authenticated, and (2) when sending the messages, which are protected using asymmetric cryptography and the establishment of network session keys. Figure 1 shows a spontaneous ad hoc network that is generated to provide the service to a client which requests the data processing or information service. The connection between the new mobile device and a device belonging to the network is secured using a pre-shared key authenticantion. Data transfers between nodes (dashed red lines) are secured using asymmetric cryptography.

We do not discuss in this paper existing systems that can be installed in the devices or accessed remotely by using our proposal (e.g. sports bets applications), since there are many provided under SaaS in regular cloud computing networks.

3.1 System Proposal and Its Operation

When the node application is running, the user should choose whether to create a new user or to validate an existing account (which has been previously created). The task of creating a user must be done the first time the application is used. This task allows generating user data, name, email, IP and listening port, public and private key and secure certificate. All data will be stored to be used later. Several users can be defined to participate in different networks. Figure 2 shows the window to create a new user. The procedure to create a new user involves three steps: (1) Data entry, (2) Creation of public and prívate keys, and (3) Certificate Creation. When the process is successfully performed, the system will store the data provided and the main menu will be displayed to select the network. After data entry, the user is created. The user must click "Next" to perform the creation of the public and private keys. These keys will be used by the user for sending next messages.

Figure 3 shows the application window that informs the user that a certificate has been generated. The application will generate the certificate necessary for distributing the public key. At the end of this process, the user will have his/her certificate created and digitally signed with his/her private key. This certificate will be exchanged during the authentication process. Thanks to it, the user will be able to send and receive the encrypted data. After creating the user and the certificate, this user will be ready to use the application.

Figure 4 shows the process followed by the nodes to join the secure spontaneous mobile ad hoc cloud computing network. In restrictive networks, the authentication process could be used to request information to the new nodes about their available resources or their installed software. In this case, the cloud network decides if this node is allowed to participate or not in the network (based on some predefined values). This procedure is explained in Figure 5. The acceptance of a new node depends on the node authentication and the data verification through a secure certificate and the data available in the network (see [18] for more details). When a node is accepted, it informs to the mobile ad hoc cloud network about its resources and services including processing capacity and battery level. Computing and storage resources are shared in the network, so any node can request resources to other nodes acording to its needs. Nodes can send update messages in order to know the current status of the network resources. The control packet verifies the signature of the node that sends the information, allowing its authenticity. Updated information will travel signed and encrypted with the network session key.

Preprint

airiti preprint

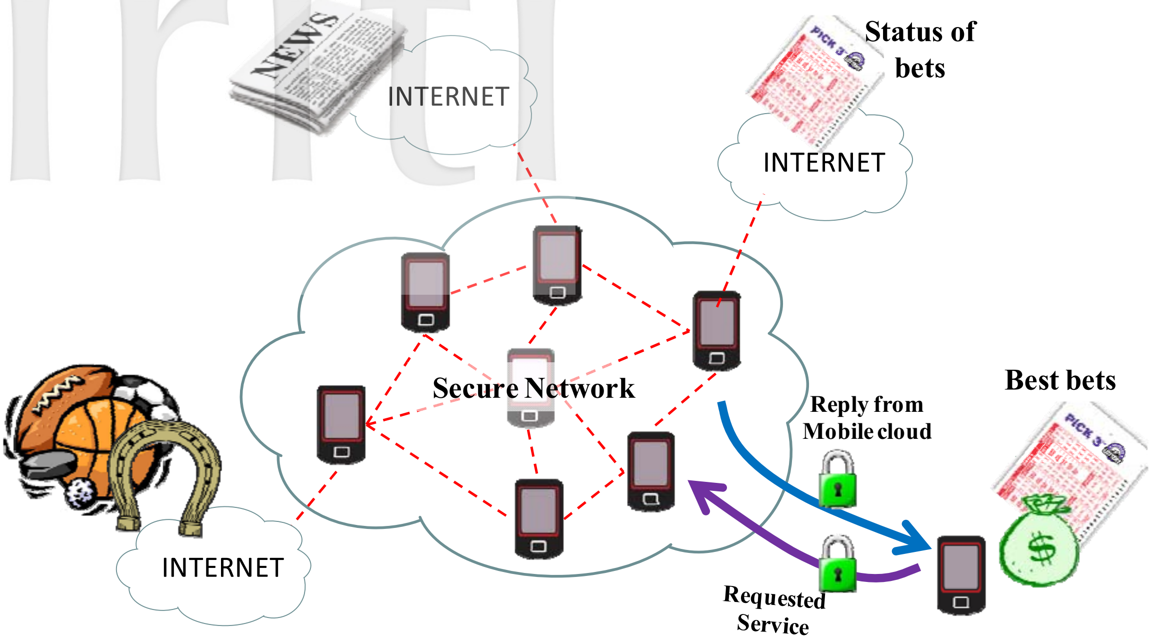
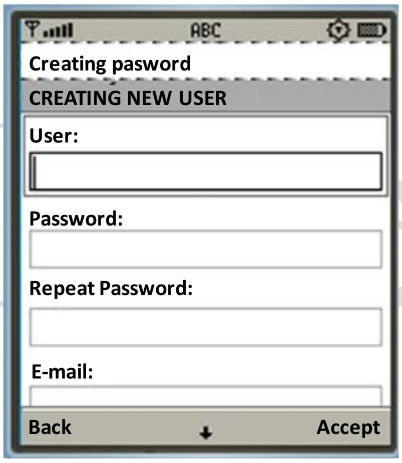


Figure 1 Secure Spontaneous Mobile Ad Hoc Cloud Computing Network Example



ABC

Creating pasword

CREATING NEW USER

User:

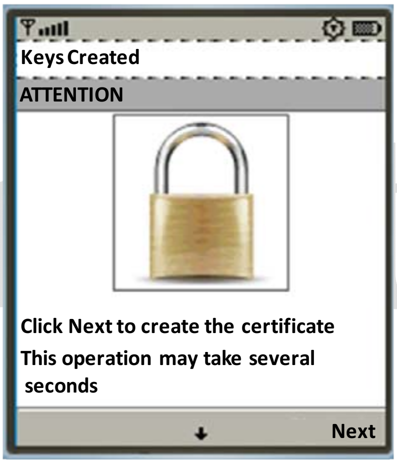
Password:

Repeat Password:

E-mail:

Back Accept

Figure 2 Initial Window for Creating a New User



Keys Created

ATTENTION

Click Next to create the certificate

This operation may take several seconds

Next

Figure 3 Application Window to Generate a Certificate

Then, services will be announced through the network following the excheme published in [17]. In an application scenario, when a node starts a sport bets software, the node is accessing to the service shared by the remote node. Many different sport bets services can coexist in the same cloud network. Any request, computation or consultation is performed by using the aforementioned secure system and shown in Figures 6 and 7.

Data can be sent by a node to a particular node or broadcast to all nodes of the network. In both cases data can be sent in plain text or encrypted with the network session key, guaranteeing the anonymity. If it is a unicast packet, it there is a third option which to encrypt the data with the public key of the receiver. Encrypting the text, data are protected against corruption. The procedure is the same followed in the first case. Figure 6 shows the flow diagram when sending data.

When a node receives a data packet, the node processes it. If the received packet has not been encrypted, it is shown to the user. However, if content has been using some encryption method, it will be previously analyzed and decrypted. Figure 7 shows the flow diagram followed during the process of receiving data.

airiti Preprint

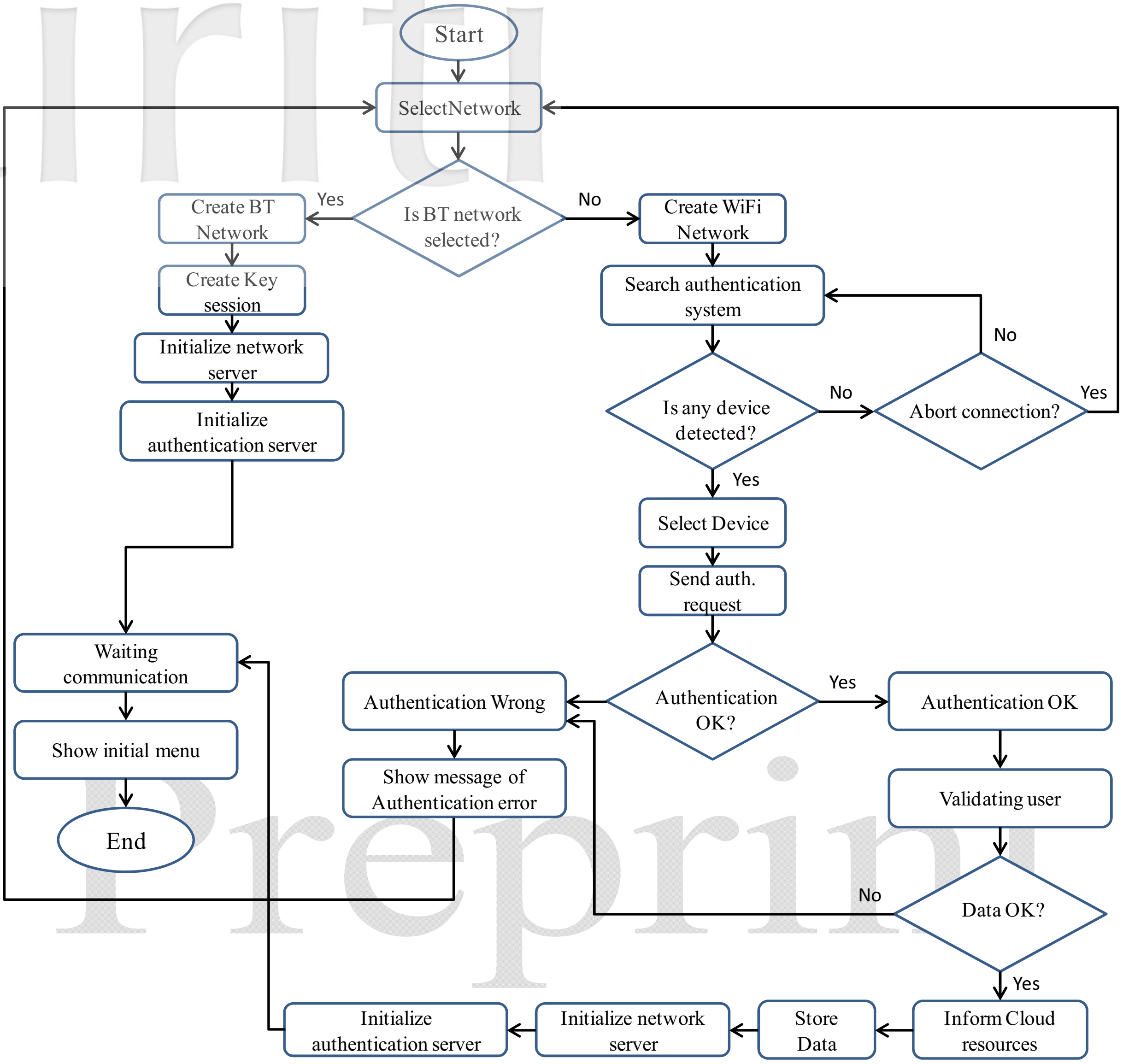


Figure 4 Network Selection

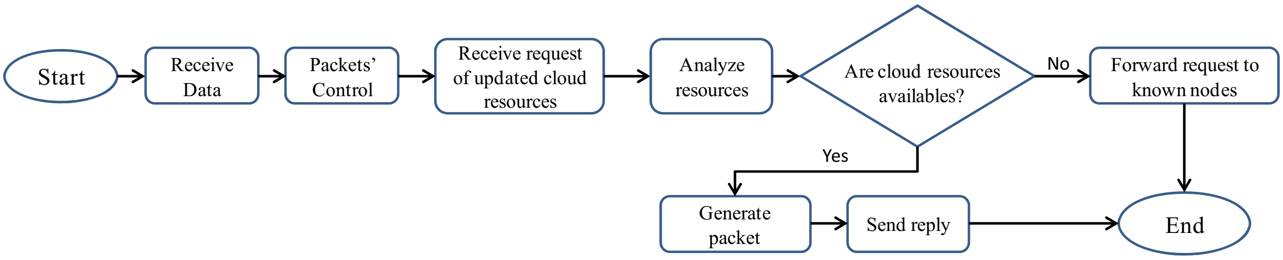

Start
Receive Data
Packets' Control
Receive request of updated cloud resources
Analyze resources
Are cloud resources availables?
No
Forward request to known nodes
Yes
Generate packet
Send reply
End

Figure 5 Reply from the Cloud to Data Request

airiti

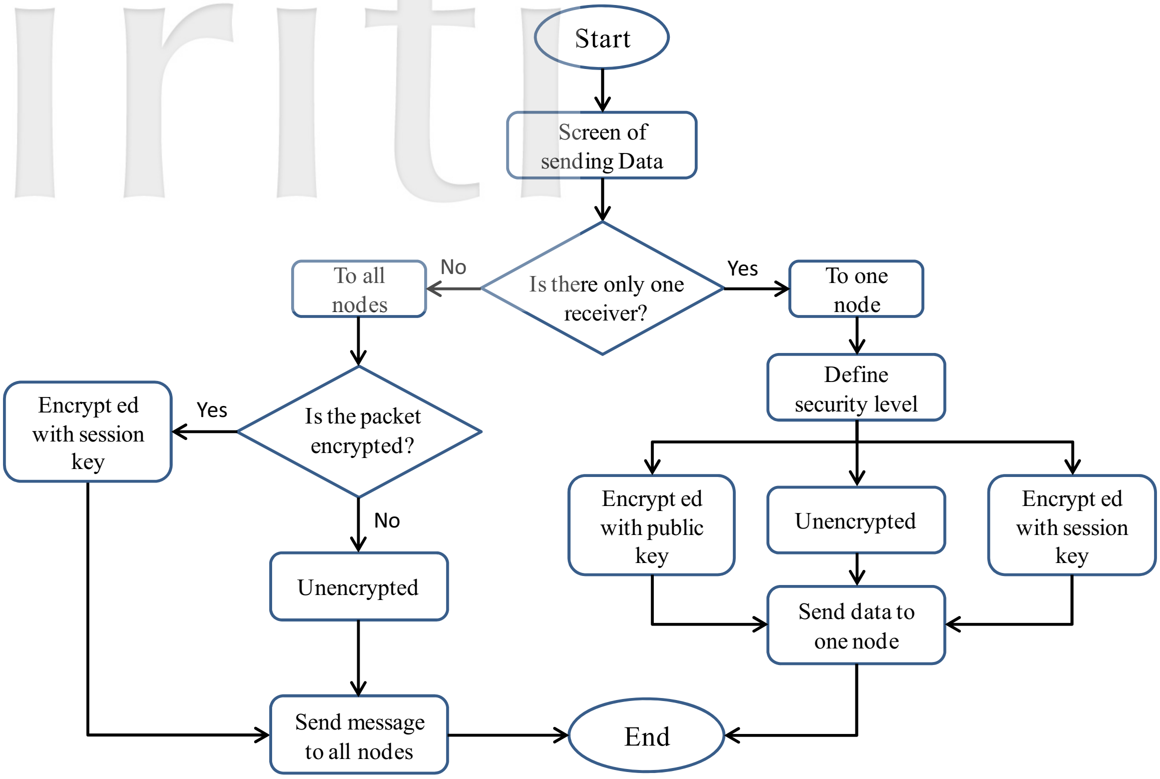


Figure 6 Flow Diagram for Sending Data

Preprint

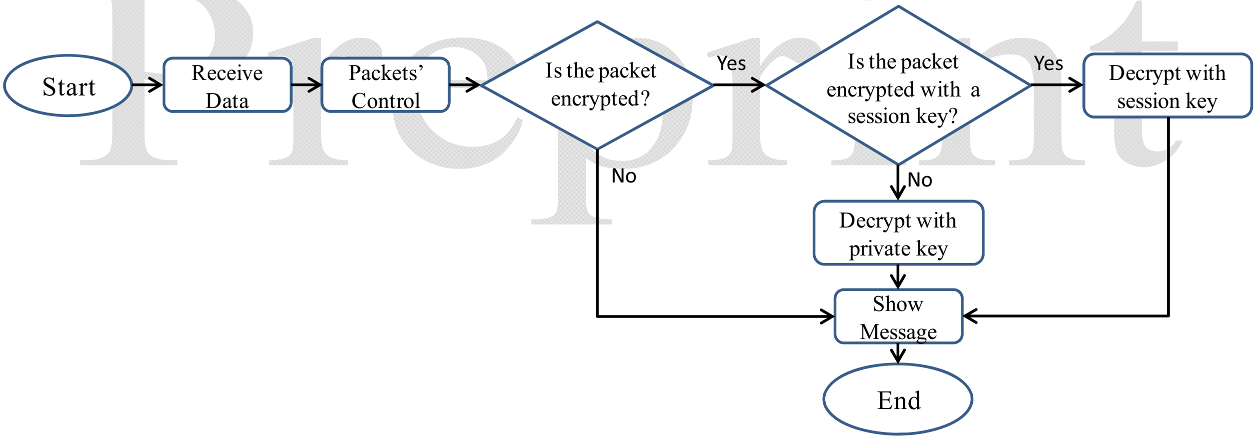


Figure 7 Flow Diagram for Receiving Data

3.2 Design of Classes for Encrypting/ Decripting the Information

The classes responsible for encrypting/decrypting data, the key generation, summary and certificate validation tasks are shown in Figure 8. The tasks performed by each class are:

- **AesCode:** The AesCode class is responsible of creating the session key and its encryption and decryption.
- **AsymetCode:** The AsymetCode class is in charge of creating public and private keys and its encryption and decryption.
- **Hash:** The hash class is responsible of performing the summary messages.
- **HexCodec:** The HexCodec class is used by the hash class to perform the summary messages.

Figure 9 shows the pseudocode of our secure application.

airiti

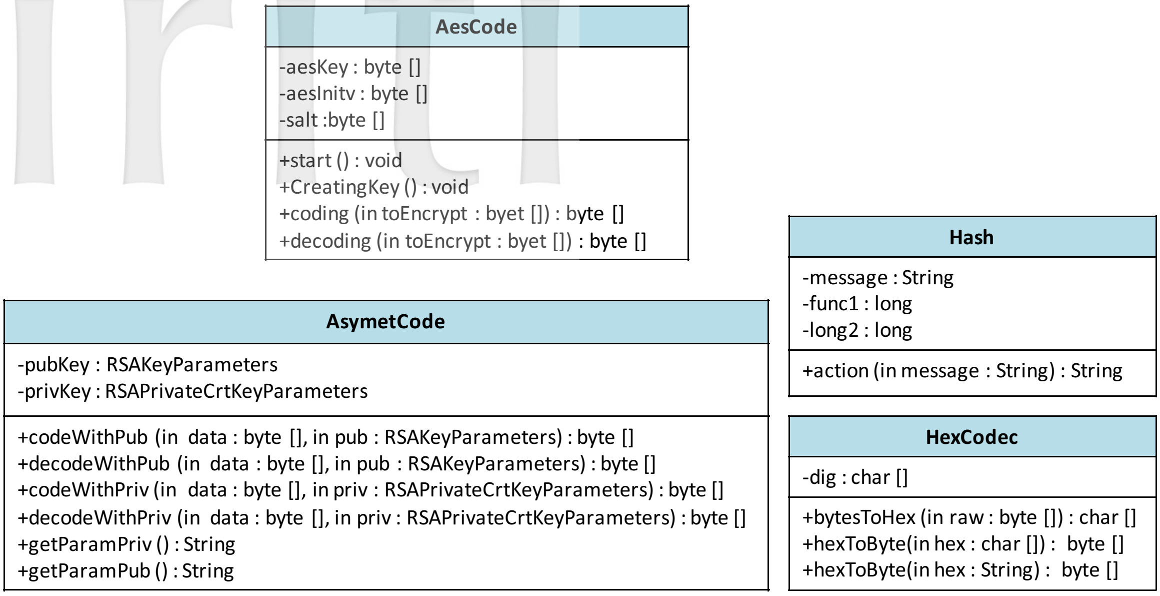

AesCode
-aesKey : byte []
-aesInitv : byte []
-salt :byte []
+start () : void
+CreatingKey () : void
+coding (in toEncrypt : byet []) : byte []
+decoding (in toEncrypt : byet []) : byte []
Hash
-message : String
-func1 : long
-long2 : long
+action (in message : String) : String
AsymetCode
-pubKey : RSAKeyParameters
-privKey : RSAPrivateCrtKeyParameters
+codeWithPub (in data : byte [], in pub : RSAKeyParameters) : byte []
+decodeWithPub (in data : byte [], in pub : RSAKeyParameters) : byte []
+codeWithPriv (in data : byte [], in priv : RSAPrivateCrtKeyParameters) : byte []
+decodeWithPriv (in data : byte [], in priv : RSAPrivateCrtKeyParameters) : byte []
+getParamPriv () : String
+getParamPub () : String
HexCodec
-dig : char []
+bytesToHex (in raw : byte []) : char []
+hexToByte(in hex : char []) : byte []
+hexToByte(in hex : String) : byte []

Figure 8 Classes Related in Encrypted/Decrypted Data Process

```
Package application;
public class AsymetCode {
        RSAKeyParameters pubKey;
        RSAPrivateCrtKeyParameters privKey;
        public AsymetCode() { }
        public void generateKeys() {
             //METHOD TO GENERATE THE PAIR OF KEYS
        privKey = (RSAPrivateCrtKeyParameters) keyPair.getPrivate(); // PRIVATE CLASS
        pubKey = (RSAKeyParameters) keyPair.getPublic();// PUBLIC CLASS }
        public byte[] encrypPub(byte[] data, RSAKeyParameters pub) {
             // METHOD TO ENCRIPT WITH PUBLIC KEY
                  Show message ("message encrypted"); }
        public byte[] encrypPriv(byte[] data, RSAPrivateCrtKeyParameters priv) {
             // METHOD TO ENCRIPT WITH PRIVATE KEY }
        public byte[] decrypPriv(byte[] data, RSAPrivateCrtKeyParameters priv) {
             // METHOD TO DECRIPT WITH PRIVATE KEY }
                  Show message ("message decrypted"); }
       public byte[] decrypPub(byte[] data, RSAKeyParameters pub) {
             // METHOD TO DECRIPT WITH PUBLIC KEY }
        public RSAKeyParameters getPublic() {
              // METHOD TO OBTAIN PUBLIC KEY  }
        public RSAPrivateCrtKeyParameters getPrivate() {
              // METHOD TO OBTAIN PRIVATE KEY }
        public String getParamPriv() {
             // METHOD TO OBTAIN PRIVATE PARAMETERS }
         public String getParamPub() {
             // METHOD TO OBTAIN PUBLIC PARAMETERS }
}
```

Preprint

Figure 9 Pseudocode Used to Generate the Keys in Our Secure Application

4 Test Bench

To ensure that the route is not obtained fraudulently, the system prevents to reply any intermediate node with the cached routes. The requested action is always replied by the destination node. Before a reply packet is sent, the destination node signs the hash of the information sent with its private key.

The source node knows the received data and the information about the route between the source and the destination. The system does not allow intermediate nodes storing packets to destination nodes whose links are broken. These tasks prevent malicious nodes to redirect packets over a given node in order to saturate or provoke packet losses. It also prevents other similar attacks.

In order to test the impact of these parameters over the network, a set of simulations with different degrees of node mobility were made using OPNET simulator. Simulations are performed in four different scenarios. Table 1 shows the parameters used in each scenario. The parameters that affect the analyzed aspects are both "Packet salvaging" and "Route replies using cached routes". They have been enabled and disabled in each scenario to compare the protocol performance in different situations. For each scenario, "simulation 1" shows the results when parameters are enabled and "simulation 2" shows the results when they are disabled.

4.1 Scenario 1

Figures 10(a), 10(b) and 10(c) show the results of scenario 1. The main aspect to highlight is that in both simulations no data packet has been lost. However, the routing packets associated with on demand requests influence highly the network performance. In the initialization process of route discovery requests, the source node sends a request packet that is forwarded by the nodes till it reaches the destination node or an intermediate node which knows a route to that node. If after some time, the source node has not received any reply, it starts a new process for discovering the route. When the route is found, the packets are sent with the route information in its header, which reduces the network load. Data submission is completed when acknowledgment packets are received. In this scenario, the traffic received is slightly higher when the parameters are disabled. Hovewer, the traffic sent is higher when parameters are enabled. When "Route Replies using cached routes" is disabled, the network is not overloaded with route reply messages. However, there is an increase of overload in the received traffic requests due to the excessive propagation of the path information. In this scenario, it is faster to find the destination node than in the rest of scenarios because all nodes are available to everyone. The route discovery time is greater in the simulation where only the destination nodes provide replies. In conclusion, we see that it does not matter if we enable or disable these parameters, because in a scenario where all nodes are reacheable and the mobility level is zero, no data are lost. Tthe absence of mobility facilitates the maintenance of the routes learned by the nodes, reducing the excessive use of route requests to restore those broken links.

4.2 Scenario 2

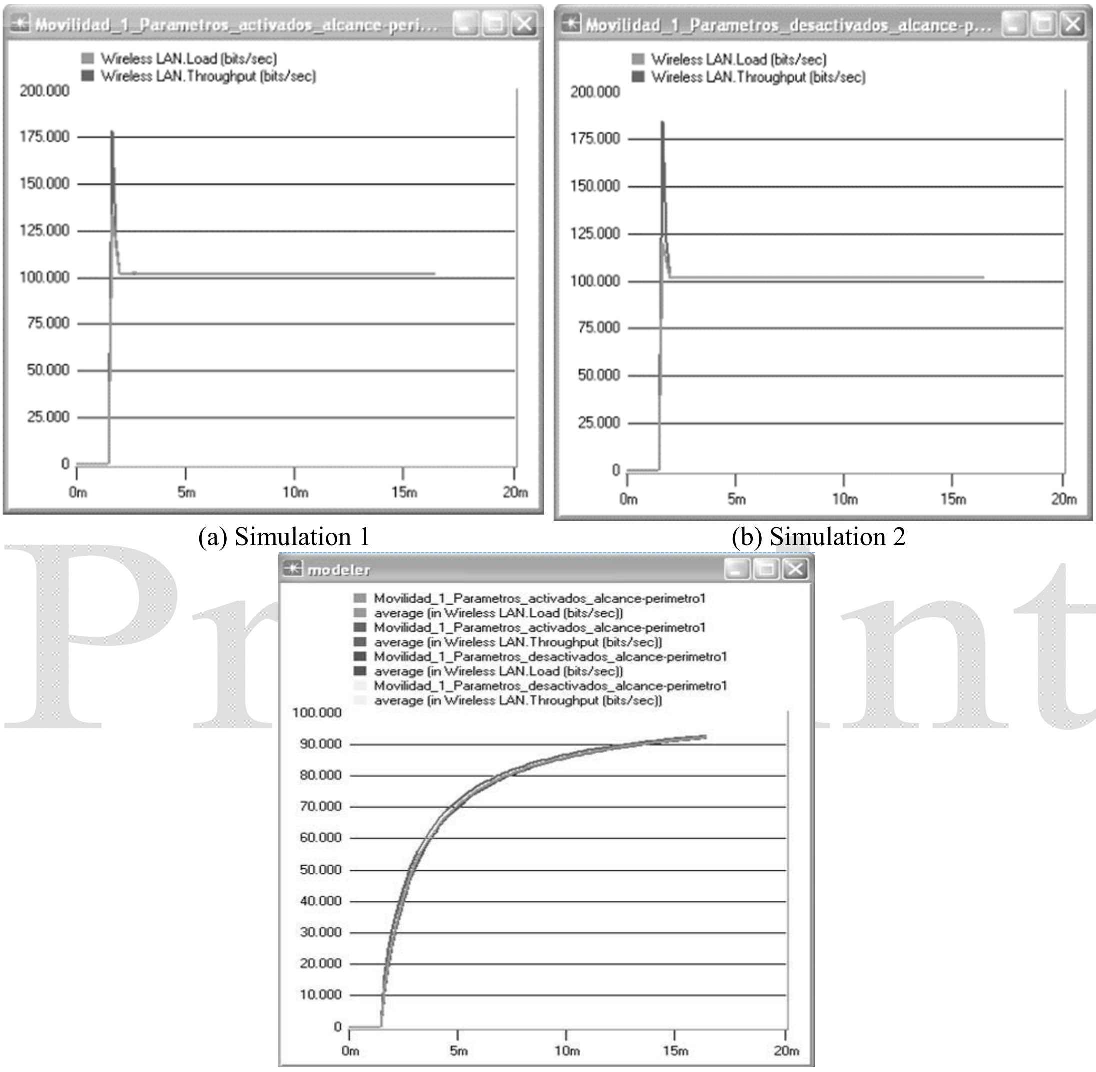
Scenario 2 explores the influence of mobility on the overall network performance. Figures 11(a), 11(b) and 11(c) show the load and the throughput supported in both simulations. As we can see, the routing traffic increasesas a result of the increase of the number of links broken. In scenario 1, it was almost zero. The mobility of the nodes, in a small size scenario implies greater number of errors in the packets of the intermediate nodes. Although the levels of load and throughput are quite similar, we can see that in simulation 1, the network load exceeds the throughput value. In both simulations, the MANET traffic sent is the same, but the received traffic is slightly higher when the parameters are disabled because there are less losses. The increased routing traffic generated by routing packets, when intermediate nodes do not reply to route requests, is offset by the increase of errors in data packets, acknowledgement packets and sent data packets generated when parameters are enabled.

Table 1 Characteristics of the Scenarios Used in Our Simulations

Scenario	Characteristics	Kind of scenario
1	Num. of Nodes: 20. Max. range: 250 m. Maximum cached routes: 26. Route Timeout: 300 s. Max. size of send buffer: 50 packets Packet Timeout: 30 s. Transmission Power: 5mW Reception threshold: -95 dBm Technology: IEEE 802.11g Data Rate (bps): 54 Mbps Size of packets sent: 1500 bytes Total number of packets sent: 7200 packets Num. of packets sent per node: 360 packets/node Packets sent to the network per second: 7.2 packets/s Simulation Time: 1000 s.	Office Size (L x W): 240 m. x 240 m.; Mobility of all nodes: 0 m/s
2		Office Size (L x W): 350 m. x 320 m.; Mobility of all nodes: 5 m/s
3		Office Size (L x W): 350 m. x 320 m.;

airiti Print

	Constant flux of packets.	Mobility of all nodes: 30 m/s
4	Number of Nodes: 67 nodes Total number of packets sent: 1075 packets Num. of packets sent per node: 16 packets/node Packets sent to the network per second: 6.7 packets/s Simulation Time: 160 seconds	Office Size (L x W): 760 m. x 840 m.; Mobility of all nodes: 15 m/s



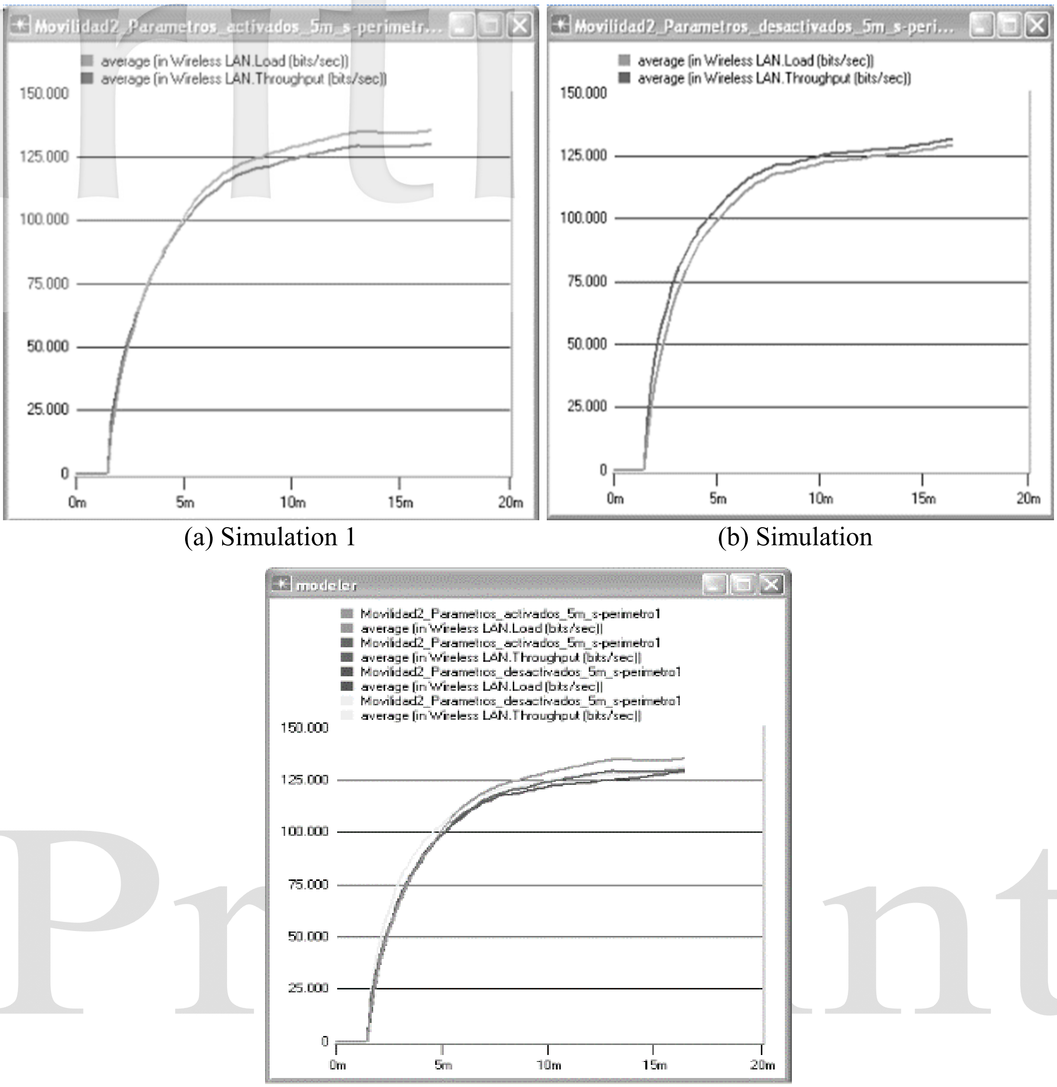
(a) Simulation 1

(b) Simulation 2

(c) Comparison

Figure 10 Load and Throughput Results for Scenario 1

airiti Preprint



(a) Simulation 1

(b) Simulation

(c) Comparison

Figure 11 Load and Throughput Results for Scenario 2

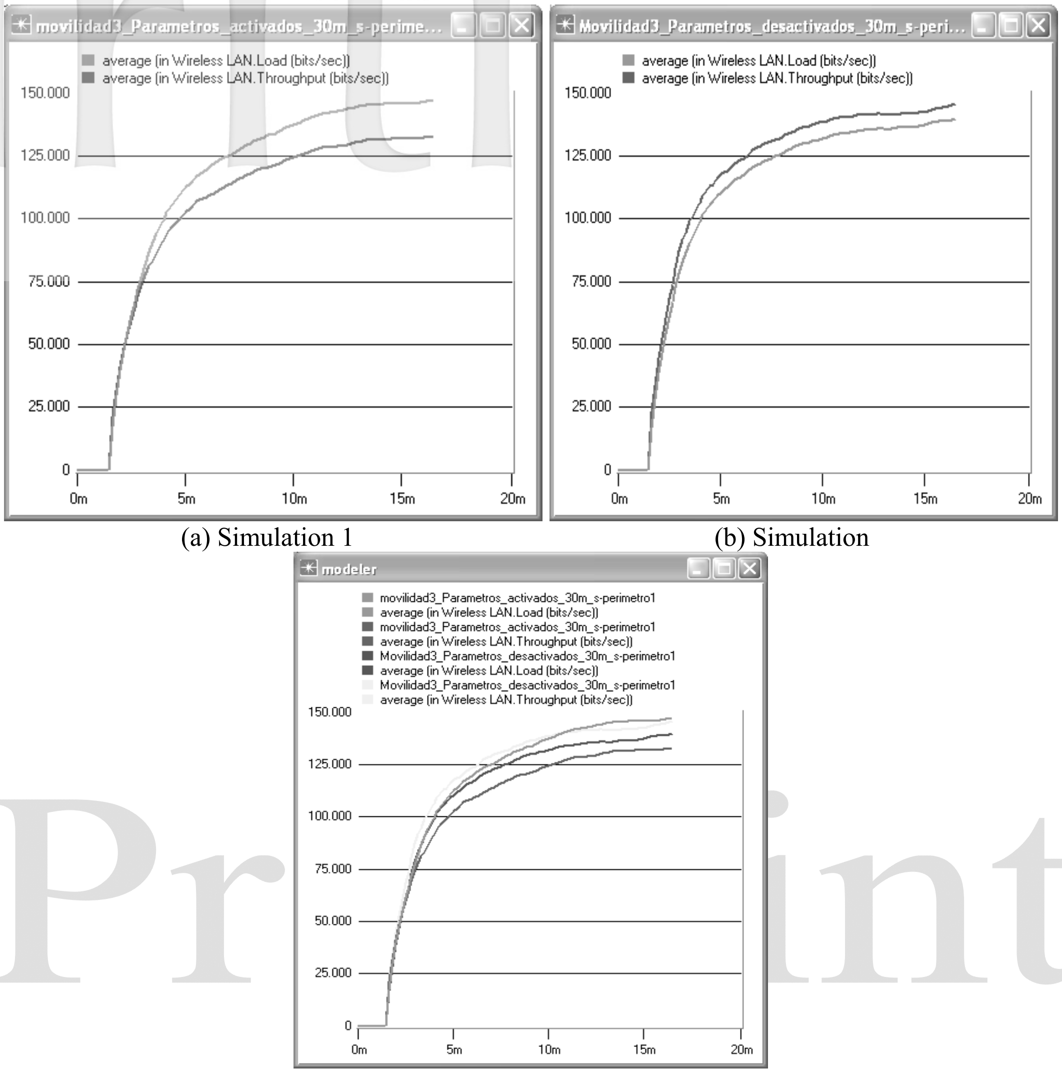
4.3 Scenario 3

Figure 12(a), Figure 12(b) and Figure 12(c) show the results of the load and the throughput supported in both simulations and their comparison. In this simulation, the number of lost packets is quite high. Moreover, there is a significant increase in traffic routing because there is higher number of broken links. The traffic received by the MANET reflects the constant loss of data packets. The loss is substantially higher than the results of simulations shown in Figure 11. This is due to the increasing number of routes learned from broken links.

In simulation 1 (Figure 12(a)), the load is higher than the throughput. This is because the number of lost packets is quite high. The existence of nodes with mobility in a small scenario implies sending greater number of route reply messages by intermediate nodes that do not find a way back to the source node. It implies more time in the route discovery process. So, if parameters are enabled, there will be higher end to end delay due to the route searching time, and higher delay in the access medium to send more routing packets (by intermediate nodes).

The number of routing packets is higher and also the network congestion. In both simulations, the MANET traffic, sent and received, is very similar.

airiti Preprint

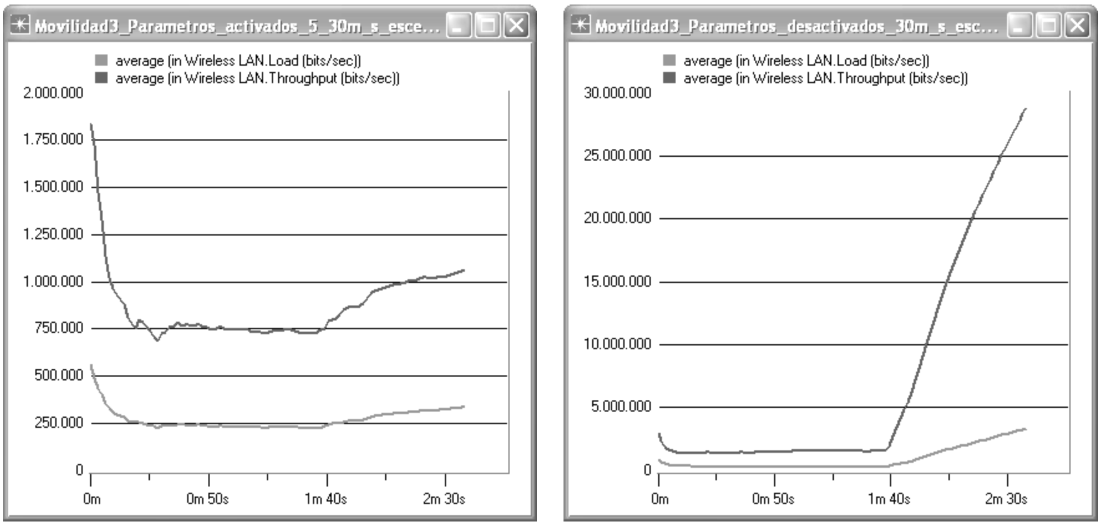


(a) Simulation 1

(b) Simulation

(c) Comparison

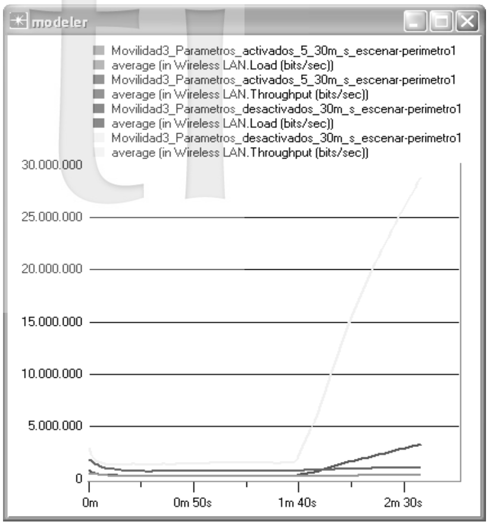
Figure 12 Load and Throughput Results for Scenario 3



(a) Simulation 1

(b) Simulation

airiti



(c) Comparison

Figure 13 Load and Throughput Results for Scenario 4

4.4 Scenario 4

Figure 13(a), Figure 13(b) and figure 13(c) show the results obtained in scenario 4. Remarkable changes are observed when parameters are disabled. The load starts to increase at 1m 40s. This happens because the nodes with mobility and the intermediate nodes (Figure 13(b)) cannot reply to the requests and save the packets with routing information. Then, the number of received packets increases and the network congestion occurs. The number of lost packets is low while the load is not high. This value increases when the load and the network congestion increase, so the time of route searching to the destination node increases. However, Figure 13(a) (with parameters enabled) shows that the protocol controls the load and maintains similar performance throughout when there is mobility despite of the large number of nodes and the high number of packets. The end to end delay is better when the parameters are disabled. The load increases and the performance changes, which becomes optimal in the other simulation. Medium access delay behaves similar. The initially received MANET traffic is higher in simulation 2 (Figure 13(b)). Network congestion becomes higher with enabled parameters. Traffic sent is similar in terms of congestion. They increase significantly at this point in simulation 2. This happens because it is needed to search obsolete routes and the management of lost packets.

In conclusion, the network (in both simulations) achieves high congestion levels when load increases due to the medium-high level of mobility of the nodes. This causes losses when sending data packets.

Preprint

4.5 Other Scenarios

We have also tested other scenarios in order to measure the maximum acceptable levels of "Packet salvaging" and "Route replies using cached routes" when these parameters are disabled. This case was obtained when the network size was 700 x 560 m^2.

Table 2 shows the combination of parameters and the most significant observations in the simulation results. In last two cases, we can see that, when parameters are disabled, the values increase in excess up to the network saturation, which does not happen when parameters are enabled. This generates a bad routing management.

4.5 System Degradation

Furthermore, the use of wireless links make these networks susceptible to attacks that may range from passive listening (eavesdropping) to active interpretations (relay, message distortion, etc.). There are many other factors which can cause the degradation of communications. They range from network security failures to user data entry failures. Figure 14 shows the main degradation sources of communication in spontaneous networks using security mechanisms and without them [21][22].

4.6 Comparison with Existing Systems

Our proposal has been compared with other existing proposals in order to highlight its benefits. The features taken into account for this comparison are the following ones:

- Trusted network based on human factors.

airiti

- Secure sistributed computation and computation issues among all network nodes. It includes cloud conection and messages exchanges.
- Well-balance security overload. Using cryptographic algorithms and checking operations randomly to guarantee security and to avoid overload.

Table 3 shows a comparison between some existing systems and our proposal.

Table 2 Combination of Parameters and the Most Significant Observations in the Simulation Results

Number of Nodes	Network Load (packets/s)	Nodes mobility (m/s)	Observations
20	2	5	Similar behavior in terms of received packets (throughput), packets sent (load) and MANET traffic.
20	2	15	Slightly greater load exists when the parameters are enabled and greater throughput exists when they are disabled. The MANET traffic received is slightly higher in simulation 2.
20	22	15	The load, throughput and MANET traffic received are slightly higher with the parameters enabled.
40	2	5	The load is very similar in both cases. The throughput is higher in simulation 1. The MANET traffic received is slightly higher, but the amount of lost packets is higher.
40	5	5	The load and MANET traffic received in simulation 1 are slightly larger than the results registered in simulation 2. Throughput is higher than the other simulations, but simulation 1shows more lost packets.
40	5	30	Load is greatly increased in simulation 2 (almost twice the load shown in simulation 1). The throughput and MANET traffic received are also slightly higher. More packets are lost when parameters are enabled.

Table 3 Comparison of Parameters and the Most Significant Observations in the Simulation Results

	Asymmetric Cryptograpy	Symmetric Cryptograpy	Hash functions	Necessity of infrastructure security central points to establish security	Route security	Distributed trust network
[11]	Yes	Yes	Yes	Yes, use of a Trusted Third Party	No	No
[12]	No	No	No	No, only trustworthiness of service provider	No	Yes
[13]	Yes	Yes	No	Yes, use of cloud service provider.	No	No
[14]	Yes	Yes	Yes	Yes, use of servers.	No	No
Our Protocol	Yes	Yes	Yes	No	Yes	Yes

5 Conclusion

The creation of spontaneous ad hoc cloud computing networks offers great advantages such as higher processing capacity. We propose a secure mobile application to make estimations using several information sources. It is based on a mobile ad-hoc cloud computing network which presents higher processing capacity than a single mobile device. The proposed system offers the users a secure access to the cloud computing network. Data are encrypted for ensuring the network anonymity and data accuracy. It has been simulated in different scenarios considering the network size and nodes' mobility.

We can conclude that the network congestion grows when parameters such as nodes' mobility, the number of packets to manage the network and the number of nodes increase. This behavior is more notable in situations where intermediate nodes cannot reply to route requests and save packets. This growth is particularly remarkable in the number of received packets. If these parameters are excessively increased, as Scenario 4 shows, the network is saturated and it presents a very low performance. However, under the studied conditions, where spontaneous networks have low number of nodes and low mobility, it is observed that both cases show acceptable network performance, since the received data traffic and load are very similar. Therefore, while the network does not have very high load levels (due to the number of packets sent and high mobility of the nodes) both ways of working are acceptable.

airiti

The proposed protocol allows the management and implementation of a distributed secure routing system that integrates both the key management and user authentication. It also optimizes the number of cryptographic operations performed and allows the participation of all nodes in the network, despite of its possible restrictions for managing this process. Finally, our protocol allows the establishment of different security levels according to the users needs.

As a future work, we will include intermediate nodes in the route searching process in order to allow them to answer route requests.

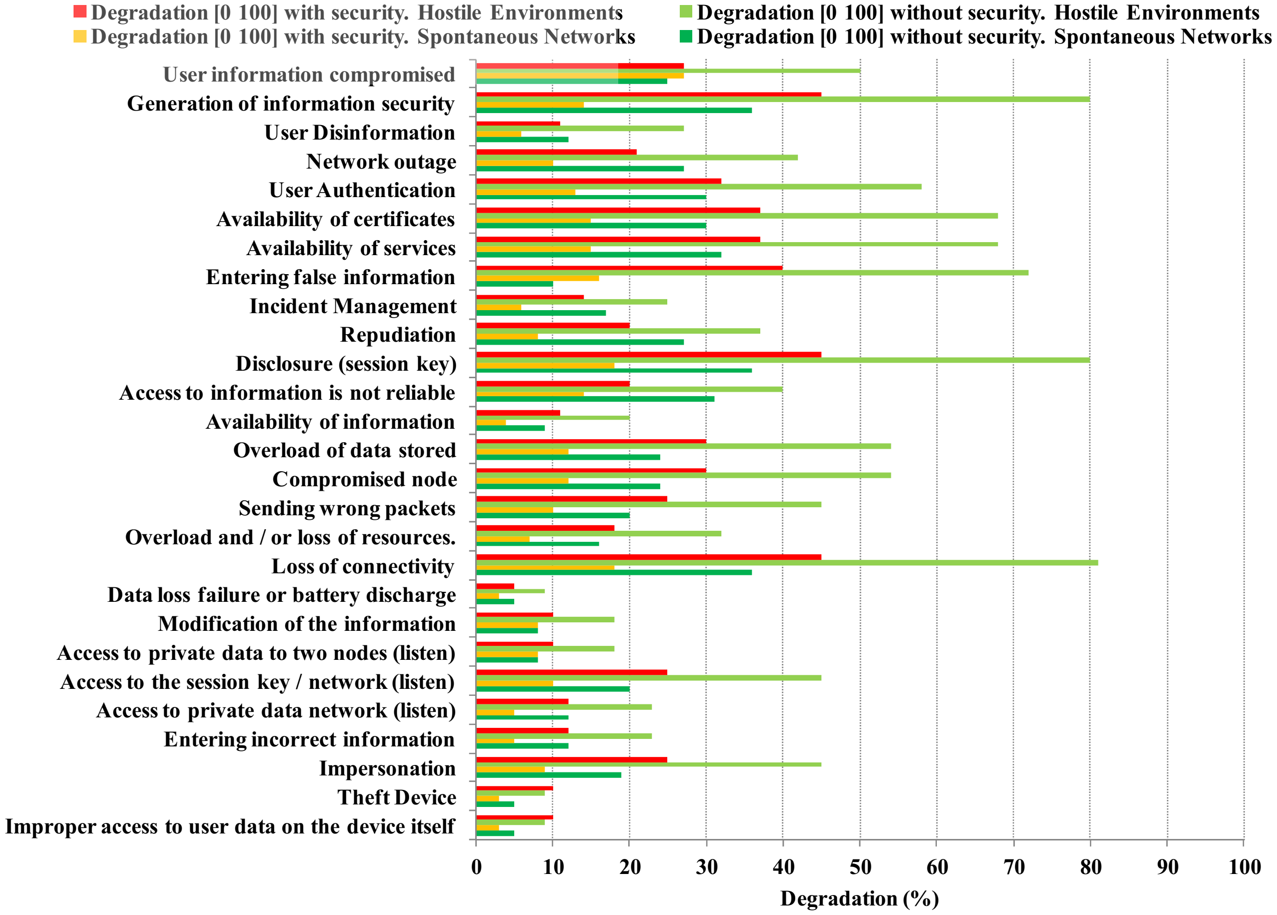


Figure 14 Sources of System Degradation in Spontaneous Networks Using Security Mechanisms and without Them

References

[1] Q. Zhang, L. Cheng, R. Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 2010, Vol. 1, No. 1, pp. 7-18.

[2] N. Zanoon, D. Rawshdeh, STASR A New Task Scheduling Algorithm For Cloud Environment, *Network Protocols and Algorithms*, Vol 7, No 2 (2015). Pp. 81-95

[3] R. Lacuesta, J. Lloret, S. Sendra, L. Peñalver, Spontaneous Ad Hoc Mobile Cloud Computing Network, *The Scientific World Journal*,Vol. 2014 (2014), Article ID 232419, 19 pages.

[4] S. Preuß and C. H. Cap, Overview of Spontaneous Networking - Evolving Concepts and Technologies, *Rostocker Informatik-Berichte*, 2000, Vol. 24, pp. 113-123.

[5] J. Lloret, L. Shu, R. Lacuesta, M. Chen, User-Oriented and Service-Oriented Spontaneous Ad Hoc and Sensor Wireless Networks, Ad Hoc & Sensor Wireless Networks, Vol. 14, Issue 1-2, Pp. 1-8. January 2012.

[6] L. Raquel, J. Lloret, M. Garcia, L. Peñalver. A spontaneous ad hoc network to share WWW access. *EURASIP Journal on Wireless Communications and Networking*, 2010, pp. 1-16.

[7] R. Dutta, Annappa B., Protection of data in unsecured public cloud environment with open, vulnerable networks using threshold-based secret sharing, *Network Protocols and Algorithms*, Vol 6, No 1 (2014).

[8] T. Cho, S.-H. Seo, I. You, A Remote Control System for Cloud-Based Smart Homes Supporting Dynamic User Management, Journal of Internet Technology, Vol. 15 No. 6, PP. 1069-1081, 11 2014

airiti Preprint

[9] J. H. Christensen, Using RESTful web-services and cloud computing to create next generation mobile applications, *24th conference on Object oriented programming systems languages and applications, OOPSLA '09,* New York, New York, USA, 2009, p. 627.

[10] R. Buyya, C. S.Yeo, S. Venugopal, J. Broberg and, I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 2009, Vol. 25, No. 6, pp. 599-616.

[11] G. Huerta-Canepa and D. Lee, A virtual cloud computing provider for mobile devices. *In Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond,* San Francisco, USA. June 15, 2010.

[12] S. Bitam, A, Mellouk, and S. Zeadally, VANET-cloud: a generic cloud computing model for vehicular Ad Hoc networks. *IEEE Wireless Communications*, 2015, Vol. 22, No.1, pp. 96-102.

[13] D. Zissis and D. Lekkas, Addressing cloud computing security issues. *Future Generation computer systems,* 2012, Vol. 28, No.3, pp. 583-592.

[14] J. Sidhu and S. Singh, Compliance based trustworthiness calculation mechanism in cloud environment. *Procedia Computer Science*, 2014, Vol. 37, pp.439-446.

[15] Q. Liu, G. Wang, W. Jie, Time-based proxy re-encryption scheme for secure data sharing in a cloud environment, *Information Sciences*, 2014, Vol. 258, pp. 355-370.

[16] N. Tirthani and R. Ganesan. Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography. *International Association for Cryptologic Research Cryptology ePrint Archive*, Vol. 2014, No.49, pp.1-5.

[17] R. Lacuesta, J. Lloret, M. Garcia, L. Peñalver, Two secure and energy-saving spontaneous ad-hoc protocol for wireless mesh client networks. *Journal of Network and Computer Applications*, 2011, Vol. 34, No. 2, pp. 492-505.

[18] R. Lacuesta, J. Lloret, M. Garcia, L. Peñalver, A Secure Protocol for Spontaneous Wireless Ad Hoc Networks Creation. *IEEE Transactions on Parallel and Distributed Systems*, 2013, Vol. 24, No. 4, pp. 629-641.

[19] S. Almuairfi, P. Veeraraghavan, N. Chilamkurti, "A Comparative Study of Authentication Schemes with Security and Usability of IPAS," Journal of Internet Technology, Vol. 15 No. 4, PP. 615-624, 7 2014

[20] W. Ren, uLeepp: An Ultra-Lightweight Energy-Efficient and Privacy-Protected Scheme for Pervasive and Mobile WBSN-Cloud Communications, Ad Hoc and Sensor Wireless Networks, Vol. 27, Number 3-4 (2015), p. 173-195

[21] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, M. Galloway, “A Survey of Key Management Schemes in Wireless Sensor Networks,” *Computer Communications*, 2007, Vol. 30, Nos. 11/12, pp. 2314-2341.

[22] V. Kumar and M. L. Das, Securing Wireless Sensor Networks with Public Key Techniques, *Ad Hoc and Sensor Wireless Networks*, 2008, Vol. 5, No. 3/4, pp. 189-201.

Biographies

Sandra Sendra is PhD in electronic engineering. She has more than 75 scientific papers in international conferences, journals and books. She is editor-in-chief of “WSEAS Transaction on Communications” and associate editor in several international journals. She has been involved in more than 130 committees of international conferences until 2015.

Raquel Lacuesta obtained her Computer Science Engineering diploma from the University of Valencia in 1999, she earned her PhD in Computer Science Engineering (Dr. Ing.) in 2008 at the same university. Her academic interest and research are security, sensor networks, human-computer interaction issues, education innovation and strategies.

Jaime Lloret is Associate Professor at Polytechnic University of Valencia. He is the head of the research group “Communications and Networks” of the IGIC. He is EiC of “Ad Hoc and Sensor Wireless Networks” and “Networks Protocols and Algorithms”. He has been general chair of 28 International workshops and conferences.

Elsa Macias-López is Associate professor at Las Palmas de Gran Canaria University (Spain). PhD in Telecommunications (2001); Author of about 10 papers in refereed journals, 40 papers in refereed conferences and co-editor of several books. Member of Program & Organizing Committees & Chair sessions for several international and Spanish conferences.

airiti

Preprint