Digital Signature for Mobile Devices: A New Implementation and Evaluation

Vagner Schoaba
Faculdade Câmara Cascudo
Av. Alexandrino de Alencar, 708
Alecrim – Natal, RN - Brazil
+55 21 (84)31981600
vschoaba@gmail.com

Felipe Eduardo Gomes Sikansi,
Daniel Fernando Pigatto,
Kalinka Castelo Branco
University of São Paulo
Av. Trabalhador Saocarlense, 400
São Carlos, SP – Brazil
+55 21 (16) 33738174
felipe.sikansi@gmail.com
pigatto@icmc.usp.br
kalinka@icmc.usp.br

Luiz Castelo Branco
LG Electronics de São Paulo Ltda. South Central America R&D Lab.
Open OS Team – Windows Mobile
Av. Nações Unidas, 14.171 - Marble Tower – 10º andar, Vila Cordeiro - São Paulo, SP – Brazil
+55 21 (11) 22025123
luiz.branco@lge.com

Abstract

This paper presents a new implementation and evaluation of digital signature for mobile devices. This digital signature system takes into account the device limitations and thus generates a functional digital signature. Its efficiency is described and justified along the paper. It also presents a case study that allows the evaluation of the device implication in the key generation that takes part of the digital signature process.

Keywords: *Digital signature; mobile devices; cryptography algorithms; hash function.*

1. Introduction

In the 70s, with the advent of computer networks, they were highly costly and difficult to understand in the point of view of operation. Only the research agencies such as Universities and Military Institutes had access to this technology, so there was great concern for information security [1].

From the 80s, two advances in technology have changed this situation. The first one was the development of microprocessors with greater processing power and lower cost, and the second one was the provision of computers in local networks of high speed called LANs (Local Area Network) [1].

Along with these networks, it was necessary to create rules and standards to ensure a safe limit, as the networks that had been previously used by a few researchers began to be used by a body of people carrying out business transactions, banking, making purchases, among others [1].

With the advent of mobile phones and later the Internet, a revolution took place. Mobility became popular with the introduction of mobile phones. This invention aroused much interest and had, in a few years, an explosion in its use worldwide, transforming the society and causing the greatest moment of mobile communication [2].

The number of mobile phone subscribers worldwide increased from 34 million in 1993 to more than one billion in 2003 and 4.6 billion in 2008 [3].

The many benefits of cell phones are obvious to everyone - anywhere, anytime, unimpeded access to the global telephony equipment via a lightweight and fully portable [4] device. As the use of these devices is still new, there is a methodology for security in the information transaction, which does not give confidence to users.

Without an effective security mechanism, a malicious user is able to capture information from other users by means of transmission, and use them as they see fit, committing fraud, causing damage and inconvenience to the owner of the information.

In financial transactions, electronic commerce, e-mail sending and other ways to send some information, it is necessary to make sure the transmitter and receiver can sign the document or transaction in a digital way, giving greater reliability to the transaction [5].

Electronic payment is becoming increasingly common nowadays, which makes necessary a high degree of trust between the media (host client, server), which is usually accomplished with the help of security protocols developed for the Internet, as SSL / TLS (Security Socket Layer / Transport Layer Security). Those protocols already give a degree of trust on both sides [6].

When considering mobile devices, they do not have an usual and efficient protocol which ensures trustworthiness in transactions and integrity and reliability of the data, as well as online payment and transfer of sensitive information.

With the creation of authentication protocols using digital signature we can provide more reliability to the transaction, allowing users to transact with the highest degree of security because the user would have a trusted host that is in the transaction. Thus, this article presents a new way to provide digital signatures on mobile devices in order to provide security in applications with transaction requiring a high degree of security. This new is called DISIMOD (Digital Signature for Mobile Devices).

2. Related Works

In [7] the implementation of a framework for payment on mobile devices for educational institutions, which enables students to perform banking transactions for the payment of fees or tuition is described. The framework provides an efficient, safe and reliable way to carry out banking transactions and reduce transaction costs for both the educational institution and the student. The framework was designed with a strong authentication and non-repudiation system through the use of Digital Signature. The confidentiality and integrity of messages are also secured through the use of public key certificates. This paper proposes security architecture and algorithms for authentication client/server similar to the one implemented in this project.

In [8] there is the study of the use of public key signatures in low-power computing devices such as cell phones and PDAs. This article examines the conceptual and practical implications of using Server-Aided Signatures for these devices. SAS is a signature method that relies on partially trusted servers to generate (usually with a high cost) public key signatures for regular use.

A major problem of mobile devices, which is their apparent inability to authenticate transactions in hostile environments, is discussed in [9]. This article was considered a framework for the prevention of adulteration of the mobile device without compromising mobility or autonomy of the agent through an approach which implements RSA encryption functions.

In [10] the author examines the risks and methods involved in the use of signatures in digital devices that are non-primary user, i.e., the user does not have total control

over applications running on the device and can render them incapable of ensure the security of private key, which is essential to the validity of the signature. This paper proposes a way of easing the exposure of the transfer of the private key for the mobile device without the need to use an encrypted channel or a secure memory device.

In the above cited works the main problem was the use of digital signature on mobile devices. Thus, their considerations are used to determine a security scheme which may be applicable in an environment with little computational power while maintaining the integrity of information.

In [11] the authors deal with the use of digital signature to ensure non-repudiation of communication. In order to do that, the digital signature and electronic document should remain valid until a specified date according to a policy of non-repudiation. As signature keys may be compromised, validity of digital signatures may be questionable, and therefore security mechanisms to maintain the validity of keys can be added to the digital signature. This article examines the mechanisms for maintaining the validity of digital signatures and provides guidance on the use of these mechanisms in the context of various applications.

The authors in [12] propose a security scheme that protects mobile devices against malicious hosts in e-commerce environment. This system uses a technique of non-detachable and Digital Signature algorithms public key cryptography based on elliptic curves that can guarantee: that the identity of the store and the customer is not forged and that the customer information and transaction cannot be modified. Moreover, the scheme is able to withstand attacks and has a lower computational consumption compared to similar schemes that use RSA encryption. The proposed scheme has features, such as ensuring the identities of hosts that will be used in this project, but using the RSA algorithms.

3. Security

One of the most common ways to implement security in a computer system is known as cryptography. In short, encryption can be explained as a set of methods and techniques to encrypt data by using an encryption algorithm parameterized by a key, converting an original text, called plain text, in an unreadable text, called cipher text. It is then possible for the receiver to decrypt this ciphertext, that is, to perform the reverse process and retrieve the original information [1] [22].

Typically, new algorithms are opened to the community as they are developed and confidentiality of information is ensured by the key, which must be kept secret and be offered only to relevant entities. The key size in this case is very important, since it determines the encryption level [1]. Furthermore, based on the type of key, one can classify the cryptography in symmetric key or public-key.

The symmetric key has this name because the processes of encryption and decryption are performed using a single key, that is, both the sender and receiver have the same key and it should be kept secret in order to ensure an acceptable level of safety. The main advantage of symmetric key encryption is that the algorithms of this type are fast and can operate on messages of arbitrary sizes [23]. On the other hand, the disadvantage of this kind of encryption is the difficulty to manage the shared key, which must be sent to all authorized users before messages can be exchanged and must still be kept secret [22].

The asymmetric encryption, also known as public-key cryptography, uses a pair of keys called public-key and private-key. Any key can be used to encrypt the data, but it

cannot be used to decrypt it, that is, if the encryption was done with the public-key, only the private key may perform decryption, or vice versa. In order to make this type of encryption successful, it is essential that the private-key be kept secret while the public-key should be disseminated to other users who want to communicate [23]. This work uses an implementation of an asymmetric algorithm.

4. Digital Signature

Digital signature is an authentication method of digital information which is typically treated, at times, as having the same level of confidence as the physical signature on paper.

According to [13]: "The safety, which is now the biggest concern of all those who trade by electronic means and the credibility of these documents is primarily attached to their originality and ensures that there has not been any alterations in the information in the paths that run from the user to a destination."

The concept of digital signature [21] suggests that it is a computer-based equivalent of physical written signatures. Although there are similarities between handwritten and digital signatures there are also fundamental differences. The main similarity is that both types of signatures can provide evidence of authenticity of a document. The differences are due to the radically different nature of paper based documents on the one hand and digital documents on the other. In paper-based transactions a document consists of text printed as ink on a piece of paper, where the text represents the information and the paper represents the storage medium. In this way the information and the storage medium are inseparable. The validity of a paper-based document is authenticated by a signature written in ink on the same piece of paper. The signature serves as evidence of the signer's agreement to the text on the paper, and the verification of signatures can be done directly without any complex instruments.

According to [14] the four keywords used to describe all the different roles that encryption plays in modern systems of information are:

- **Confidentiality and Privacy**: used to scramble information sent by tele-transmission in open channels and stored on a server, so that malicious people cannot access the contents of the data.
- **Authentication:** ensures the identity of the person who is sending the message. The recipient of the message can verify the identity of the person who signed it.
- **Integrity:** ensures that the message content is not modified in transit.
- **Non - repudiation:** has the function of ensuring that the author of a message cannot falsely deny having sent it.

In the approach of Public Key Infrastructure (PKI) there is the concept of digital certificate, which in short is a data structure that ties user data such as name, country, and organization to its public key.

A digital certificate is normally used to link an entity to a public key. To ensure digitally, in the case of a Public Key Infrastructure (PKI), the certificate is signed by the Certification Authority that issued it and in the case of the standard Web of Trust, the certificate is signed by the entity and by others who say they trust in that entity. In both cases the signatures contained in a certificate are a statement made by an entity that says trust the data contained in that certificate.

The use of digital signature without any certification method of the public key could allow attacks where the authenticity of a user is forged, e.g., an attack of "man in the middle."

The attack "man in the middle" occurs when a malicious user performs independent connections with the victims and relays messages between them, making them believe they are talking directly with each other over a private connection when in fact the entire conversation is being observed by the attacker. The person responsible for the attack should be able to intercept all messages exchanged by the victims and, if necessary, manipulate them. Usually the attacker is within the means of communication between two nodes, being part of the communication channel.

5. DISIMOD

With the use of increasingly comprehensive means of communication, it is necessary to have resources to ensure the security of communication and the integrity and confidentiality of information as well as the origin of the communication. Thus, aiming to improve the transfer of information on mobile devices and especially to provide security and reliability of the information, this section presents a cryptographic system for mobile devices.

The DISIMOD - Digital Signature for Mobile Devices - is an implementation aimed at increasing security in transactions and communications by means of portable devices such as cell phones, Smartphones and PDAs.

The communication occurs via Bluetooth, which enables a wide range of devices to have access to this feature and is able to hold messaging. Bluetooth technology basically requires two primary factors: knowing the devices on the "neighborhood" and the existence of a predetermined circuit [25].

Since this implementation was designed specifically for cell phones, which is a type of embedded system, it is necessary to consider resource constraints. While working with these systems, we must consider the following criteria [24]:

- **Energy** - amount of energy required for the processes of encryption and decryption of data;
- **Program memory** - amount of memory needed to store the encryption algorithm;
- **Buffer** - amount of RAM required for running the encryption algorithm;
- **Execution time** - time needed to perform the procedures for encryption and decryption of data;
- **Memory parameters** - amount of memory needed to save the keys used by the encryption function.

The implementation of DISIMOD made use of a library "bouncycastle" [15] which implements a library for working with RSA, allowing work on projects for digital signature.

The main feature of the proposal for a digital signature system with mobile devices has been the increase in the use of the media seen as unreliable, such as structure of the cell phone networks and Wi-Fi.

The DISIMOD is based on RSA public key algorithm, as it has extensive documentation and support for Java, and a satisfactory level of security because it is an algorithm in which the security is not only based on the project, but rather on the distribution of keys.

RSA has strong cryptographic features and can be used for both encryption (reason for the original development of the algorithm) and for digital signature encrypting of the message because it allows breaking the public key for private (conventional encryption) and the private key for the public (digital signature). RSA is widely used by

financial institutions with the aim of ensuring reliability in transactions, trying to avoid bank fraud, and to capture information in various networks like the Internet, using sniffers techniques. It has become almost synonymous with public-key cryptography, although there are many other potentially useful algorithms [4].

The implementation of RSA for DISIMOD was developed using Java and J2ME environments.

In addition to RSA, a message digest function, called a hash function, is used. Hashing function is a kind of fingerprint and therefore it serves to ensure the integrity of the message content it represents. So after calculating the hash of a message by using a hash function, any change in its content - even just a bit of the message - will be detected, because a recalculation of the hash value on the modified content will result in a very different hash value. The algorithms that implement the most widely used hash functions are SHA-1 and MD5.

The hash algorithm chosen was the MD5, which is widely used (MD5 is used in transactions with passwords and verifies the integrity of content on the Internet) and has native support for Java as well as being widely used in the market, thus demonstrating the feasibility of its use, and allows easy implementation in J2ME environments.

5.1. DISIMOD Development

The DISIMOD was developed using Java 2 Mobile Edition, because it is a development platform conceived by Sun Microsystems in order to enable the development of applications for devices with low processing power and memory storage.

The minimum requirement needed for the implementation of DISIMOD is that the mobile device has support for Java Virtual Machine and the library JSR-82 (present in most of the devices with JVM), which is required to use the device's Bluetooth feature.

Bouncy Castle is a class library developed in Java for applications with safety requirements. Bouncy Castle implements routines in various cryptographic algorithms in the market [16].

The Bouncy Castle Library presents important features that facilitate the work of developing security applications. Among the many classes included in the library, we use the symmetric cryptographic ciphers IDEA, AES, DES.

The library already has implementations for hash functions MD5 and SHA-1 as well as algorithms for asymmetric encryption and digital signature.

Among the features offered by the library, one of the most desired that allows increasing productivity is the data type "BigInteger". Once in Java this data type is not native, it was necessary to create an abstract data type to support a numerical value that allows full manipulation of extremely large data.

Among the classes used the Bouncy Castle we can list:

- org.bouncycastle.crypto;
- org.bouncycastle.crypto.generators;
- org.bouncycastle.crypto.signers;
- org.bouncycastle.crypto.params;
- org.bouncycastle.crypto.digests;
- org.bouncycastle.crypto.engines;
- org.bouncycastle.util.encoders;
- org.bouncycastle.crypto.encodings.

The classes described above have been developed in Java and J2ME also to allow the use and development of mobile applications.

The implementation of DISIMOD has a method that performs the generation of keys, which are kept in a plain text file in the following format: user.prikey, where the user is always identified by the nick that is chosen at the beginning session. This file contains the user private key USER and in turn it also has the file containing the public key in the following format: user.pubkey. This format allows the use of keys to facilitate the exchange of keys using Bluetooth or a repository, therefore users can collect the key for a particular user to perform the exchange of information. Is also implemented the method used to generate the hash which aims to ensure that the original message is not altered at any point of the communication process, since Bluetooth does not guarantee the inviolability of the information.

5.2. Class Diagram

In this section we describe the development process through the class diagram and the functionality at each step of the process. The class diagram is published in [26].

- RSASigUtil: Class which, regardless of the package, has methods for the application of the signature using the RSA;
- RSASigTest: Class which aims to generate the keys by their methods and usage model;
- BTListener: Class that uses Bluetooth to enable the exchange of messages between devices;
- MessageUI: Class which provides the setting for the exchange of messages between participating hosts;
- ChatPacket: Contains methods that perform communication via CHAT aiming to accomplish the transfer of information;
- Sender: Class that is intended to send the message using bluetooth;
- Reader: Class that performs the reading of the message received through the process of networking bluetooth;
- EndPoint: It has methods for manipulating Bluetooth network nodes;
- InputUI: Class which contains the methods that interface with input information from users;
- NameUI: interface has methods to add the names of the participants of the communication;
- Util: Class that keeps some variables static and helps in debugging;
- NetLayer: Class that has methods that perform a scan using the bluetooth in a search for servers;
- ChatMain: The ChatMain class is the main one, which performs the instance of other classes that are required for all processes. This class needs to perform the instantiation of classes RSASigUtil, RSASigTest, BTListener, NetLayer, NameUI, InputUI, Util, Sender, MessageUI, ChatPacket, Reader, and EndPoint.

5.3 DISIMOD Operation

The DISIMOD operates through the use of the private key, generating the encrypted message, and thus there is no guarantee of the confidentiality of information, because as it has been said earlier, the RSA works with two keys, a public one and a private one.

Generating information with the private key is thought to guarantee the originality of the message, thus avoiding action repudiation of messages in which the sender will not be able to deny having generated the information.

After generating the information with the private key, the user sends the message to whomever he/she wants and informs the public key. Upon receiving the message, the recipient will have to decrypt the message with the sender public key.

To decrypt the message with their public key of the sender, there is the certainty that the origin of the message cannot be denied.

With this use, it is possible to verify the origin of the message, thus ensuring that the sender, not the content of the message is genuine or has been read by some external source. Also, it does not guarantee that there is not means to read or, even worse, change the information, because as it has been pointed out, encrypting information using the private key is not a guarantee of confidentiality, but only the authenticity of the message.

To ensure the uniqueness of the message it is necessary to use another computational technique, called hash. With the use of hash functions the integrity of the message can be ensured, ensuring that the message has not been altered in transit between sender and recipient.

With the use of RSA and MD5 we can ensure the integrity, originality and authenticity of the message. Thus, there is the certainty of the origin of the message by using the RSA using the private key to generate the encrypted text. Applying the MD5 hashing algorithm in the cipher text it will generate a summary from the message encrypted.

When the receiver receives the message, it decrypts the message and checks if the message has been altered during transit, making verification of the hash. If the hash does not match the hash generated, it means that the message was altered in the path between sender and recipient, intentionally or accidentally.

When verifying the hash, it is possible to see that the message traffic is intact and unchanged and may now be decrypted without harming the recipient and ensuring the origin of the message.

The use of hash after the encryption of the message reduces the computational cost in time to decrypt the message, because the RSA algorithm, as well as functional, consumes too much processing capability. So, before using the decryption algorithm, there is the integrity of the message, and if this is confirmed, means that the document is intact and therefore ready to be read and ensure the authenticity and originality.

The DISIMOD performs the sending of messages using the dynamic generation of keys, as each user is responsible for generating its own pair of keys for communication. It is not necessary an entity verifier in this case, so it is possible for a malicious user to intercept the communication with a key pair itself indicating that it belongs to someone else.

In DISIMOD this threat is minimized by the fact that Bluetooth communication requires all connected users to be physically close, and in an environment without obstacles.

In order to perform tests, a static key embedded in the code was established because the generation time of the keys is high, making the test difficult.

The DISIMOD uses Bluetooth for transmission of information between different devices, which enables communication at low cost and a reasonable transmission rate.

Bluetooth has an open architecture, which facilitates its implementation, expansion and use by facilitating the implementation of new technologies and exchange of

information between existing devices [17]. However, Bluetooth does not natively have security features, so it does not fulfill the needs of secure communication, and therefore it is necessary to implement a solution. Thus, the DISIMOD uses the RSA digital signature to perform at that time.

The DISIMOD enables the exchange of public keys via a server, which serves as a repository of keys in order to allow the acquisition of public keys and ascertain the originality of the message and non-repudiation.

The DISIMOD was developed with technology based on CHAT, and thus can permit the exchange of information and can evaluate their use with a volume of variable information allowing single or multiple messages sent sequentially.

Figure 1 shows the communication process in multiple devices.

Figure 1 – Multiple Device Communication Using DISIMOD.

The DISIMOD receives the connection request from a second device, which verifies the ability to receive this connection. With the authorization, the system allows the entry of a new element in communication.

During the process of communication in every information exchange between devices, the signature is verified in the device issuing the message. If the message does not have a valid signature, the message is rejected, but if the signature is valid, the communication is allowed.

The process of connection between devices is as follows:

- The device "A" connects to the CHAT that composes the DISIMOD and waits for a connection request;
- Then a second device, here called "B", requests a connection in the CHAT that is allowed without any restriction, which is displayed when the connection is established;
- Both devices involved in the negotiation are started by waiting for the communication process as both parties are ready and with the already established connection.

The device has a MAC address to be identified by the other members of the communication process. The device "A" has the MAC address 00:00:1E:20:11:12, which must be transmitted to the device "B".

After establishing the connection, both clients are on standby to receive information from the devices. At the moment that the connection is established between the participants, the signature is not checked, neither the source information. Only during the exchange of messages both the signature and the original information are checked.

Figure 2 shows the process of sending a message between two devices.

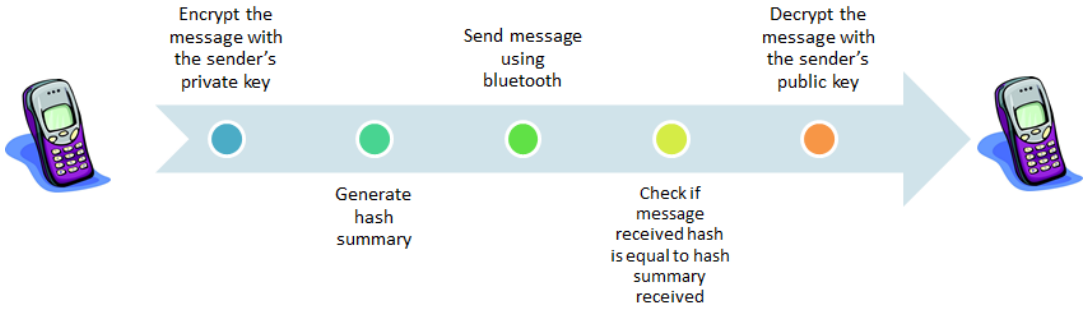

Figure 2 – Steps While Sending a Message.

6. Experimental Results

The DISIMOD was tested in a real environment where the system was installed on real devices. The usability and system behavior in active devices were verified. The models used are described below as well as their architecture and elapsed time in each test.

The DISIMOD was configured with the static key in the code to perform the tests, since the devices with low processing power, such as the most common models of mobile phones in the market, generate a 1024 bit key and takes a long while to get that. The time to generate the 1024 bit key, in most cases, is longer than 5 minutes, since the system which is installed several mobile devices are the ones commonly found in the market.

If we had prior knowledge about the generation times of the keys in order to ascertain its feasibility, we would be able to use real equipment, without emulation, since the emulation does not allow the same fidelity as in obtained results from real physical devices.

In a Motorola phone brand, model K1, it was not possible to generate keys in an adequate time, since it has exceeded 30 minutes. In some measurements, this time reached 55 minutes to completely generate the keys.

The model K1 Motorola has 722 KB of RAM, and this is the amount of total memory, but the device delivers only 354 KB for use in other applications. This device has a 23.4 MHz processor Motorola and 1.3 MHz of processing dedicated to the Java Virtual Machine [18].

By using a device with more features and superior processing power, the time can now be considered acceptable.

In another test case, a model Nokia E61 was used and the times obtained had large variation in the generation and validation of keys. The times ranged from 33 seconds to 1 minute and 5 seconds. Measurements were made 15 times at different times, and this oscillation occurs just in time for the measures being taken in these different situations: the unit completely idle, and the device in use.

The times measured take into account both the process of generating the keys and the process of signature validation.

Even with such oscillation, the time that the model of Nokia took to generate and validate a signature can be considered within tolerable limits, because it is a key 1024 that is being dynamically generated on a device with low memory and low power processing.

The Nokia E61 has an ARM 9 processor 220 MHz [19], which comes to represent 9.4 times more processing than the K1 model, Motorola. It is 85 MHz processor dedicated to the Java Virtual Machine and 1727 KB of total memory cell, of which 648 KB is free for use by other processes.

There are test cases with other models from Motorola. The A1200 model already has a more robust architecture than the other models previously tested. The model of the Motorola A1200 has an Intel Xscale 312 MHz and 85 MHz processor dedicated to the Java Virtual Machine. It also has 1727 KB of memory cell, and 648 KB of available memory. However the model of the Motorola K1 has a Linux operating system, which produced a performance difference compared to the model of the Nokia E61.

The times of the model A1200 Motorola had less variation, the time of generation and validation of the keys was around 16 seconds, a time that can be seen as satisfactory since it is still a device without a great processing power when compared to a personal computer.

Tests were also performed using the simulators Sun, personal computers, but the Wireless Tool Kit (WTK) showed no reliability in the execution of the same test cases, because the times had large surge, that these times were 39 seconds to 1 minute and 5 seconds in the simulator.

Taking into account the average time in tests, the WTK simulator made the generation and verification of keys in an average time of 44 seconds, but could not evaluate the simulator faithfully reproduces the actual application runs on mobile devices.

Figure 3 allows for a better view as to the times to generate the keys, when using the same in each test case.

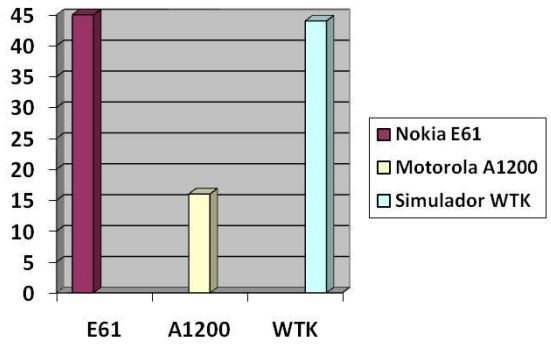

Figure 3 – Keys Generation Times

According to the results obtained, we can observe and conclude that the DISIMOD allows multiple applications, and among them, one can always cite as the main focus to increase the guarantee on security in various transactions that involve communication using Bluetooth. You can see also that there is a direct influence on generation time and validate the key directly dependent on the processing capability of the device in question.

It should be emphasized that the objective in using this survey has been achieved, so that one has, from the development of DISIMOD, greater reliability and security when using mobile devices to carry out activities involving commercial and personal risk.

7. Conclusion and Future Works

The theme of security in data communication should never be underestimated, since it is known that there is no absolute security, just as there are several ways to deal with security flaws or lack thereof.

This is due to the existence of a globalized world, in which every day there is an increasingly strong desire for connectivity, and as everyone yearns for greater security in the means of existing connections, this study proposed a digital signature for mobile devices.

Based on the human need to always be connected, to always have access to applications, e-mails, always keep informed about the latest news of nowadays, as well as financial quotes, it is essential to try to establish a device that has increased security in these transactions which are already so common.

Thus, the DISIMOD represents one of several security solutions that can solve these problems, thus allowing a real connectivity, such as IPSEC, VPN, and where they can be used handheld devices like PDA and Cell phones.

The DISIMOD was designed and developed with the aim of providing security in a mean of communication that is of great insecurity, despite the numerous projects that involve data communication via WI-FI, with many different technologies and means of access. Despite these various attempts, to date, no solution has been envisaged in the open literature that could be highly effective and functional, thus the need of always looking for a more appropriate solution.

In data transmission using hybrid networks (mobile phones, Wi-Fi), security concerns go ever further, with the possibility of having intercepted signal, changes in content, or even denial of transmission.

The solution proposed by DISIMOD provides security during the transmission process and can always ensure of the origin of the message and avoid false messages and receiving from unauthorized sources.

The DISIMOD allows use in real environments can be used to trigger distributed applications, or even monitor the results in an application.

The DISIMOD can also be used for commercial purposes, such as establishing communication between two devices that wish to make the exchange of documents, use the digital signature process payments or perform banking transactions using e-banking services.

To make the DISIMOD, it has been considered the need to connect to applications that use database based on the Internet and what can be done besides consultations, alterations and additions. The possibility of work with sales systems using the Internet as a basis for transactions and marketing products and solutions was also weighed.

The implementation of DISIMOD signature in different test cases demonstrated the feasibility of using digital signatures using asymmetric cryptographic algorithms. It can be observed that the viability and performance of the signature are related and totally dependent on the processing capability of the device in question.

To minimize the problem in relation to generation time of the key pair in the mobile device it is possible to adopt a strategy of generating keys out of the device, through an authentication server, thus creating an entity responsible for creating the keys and move them securely to the mobile device.

Thus the mobile device would be responsible only for encrypting and decrypting the data, processes with a computational cost much lower than the generation of key pairs.

As future work we will provide a functional bank application using DISIMOD as digital signature, providing users buy things from a mobile device in a security way. We can use this digital signature in critical embedded systems like communication in a convoy or in communication in unmanned aircraft vehicles (UAV). We plan to carry out real tests in those embedded platforms.

Also as future work, it is possible to change the communication protocol used. An option that could give advantages is Zigbee, which intends to be simpler and less expensive than Bluetooth. It would be also possible to evaluate and compare performance between these protocols.

Acknowledgments

The authors acknowledge the support granted by CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology - Critical Embedded Systems - Brazil), processes 573963/2008-9 and 08/57870-9. The authors also acknowledge the support granted by FAPESP, relative to the process 2010/07943-0.

References

[1] TANEMBAUM, A. S., “Redes de Computadores”. 4º Edição, 2003.

[2] TAURION, Cezar, “Internet Móvel Tecnologias, Aplicações e Modelos”. Rio de Janeiro – RJ. Editora Campus 2002.

[3] International Telecommunication Union. Available in http://www.itu.int/ITU-/ICTEYE/Indicators/Indicators.aspx.

[4] Kurose, J. F.; Ross K. W. “Computer Networking: A Top-Down Approach”. Addison Wesley Publishing Company.

[5] FERREIRA, Lucas C.; DAHAB, Ricardo. Blinded-Key Signatures: securing private keys embedded in mobile agents. Março, 2002.

[6] CLAESSENS, Joris. PRENEEL, Bart. VANDEWALLE, Joos. (How) Can Mobile Agents Do Secure Electronic Transactions on Untrusted Hosts? A Survey of the Security Issues and the Current Solutions. ACM Transactions on Internet Technology (TOIT). Vol. 3. Fevereiro, 2003.

[7] KUMAR, S. B. R.; Rabara, S. A.; MARTIN, J. R., “A Secure Mobile Payment Consortia System for higher educational institutions”. St.Joseph’s College, Bishop Heber College, 2009.

[8] DING, X.; MAZZOCHI, D.; TSUDIK, G., "Equipping Smart Devices with Public Key Signatures". Singapore Management University, Istituto Superiore Mario Boella, University of California, 2007.

[9] KOTZANIKOLAU, P.; BURMESTER, M.; CHRISSIKOPOULOS, V., "Secure Transactions with Mobile Agents in Hostile Environments". University of Piraeus, Royal Holloway, University of London, 2000.

[10] CAMPBELL, Scott,; "Supporting Digital Signatures in Mobile Environments". Miami University, 2003.

[11] ZHOU, Jianying. DENG, Robert, “On the Validity of Digital Signatures”. Kent Ridge Digital Labs21 Heng Mui Keng Terrace Singapure, 2000.

[12] SHI, Yang; CAO, Liming; WANG, Xiaoping; "A Security Scheme of Electronic Commerce for Mobile Agents Uses Undetachable Digital Signatures". Tongji University, 2004.

[13] BRASIL, Â ngela Bittencourt. Assinatura Digital Não é assinatura Formal. Ministério Público Federal. 2001.

[14] GARFINKEL, Simson; SPAFFORD, Gene. Comércio e & Segurança na Web. São Paulo, Market Press:1999.

[15] BOUNCY CASTLE. Available in: http://bouncycastle.org Last access 06/2009.

[16] SUN MICROSYSTEM. Available in http://java.sun.com/J2ME: Last access 06/2010.

[17] BLUETOOTH. Available in http://bluetooth.com/Bluetooth/Press/SIG/Bluetooth_Special_Interest_Group_Launches_Bluetooth_Core_Specification_Version_20__Enhanced_Data_Rat.htm) Last access 06/2010.

[18] Club-Java. Available in: http;//www.club-java.com Last access 06/2010.

[19] GSM Arena. Available in http://www.gsmarena.com Last access 05/2010.

[20] ANDERSON, Ross. BERGADANO, Francesco. CRISPO, Bruno. LEE, Jong-Hyeon. MANIFAVAS, Charalampos. NEEDHAM, Roger. A New Family of Authentication Protocols. ACM Sigops Operating Systems Review. Vol 32. Issue 4. Outubro, 1998.

[21] DIFFIE, W.; HELLMAN, M. E.; "New directions in Cryptography". IEEE Transactions on Information Theory, 1976.

[22] MORENO, Edward David; PEREIRA, Fábio Dacêncio; CHIARAMONTE, Rodolfo Barros. Criptografia em Software e Hardware. São Paulo: Novatec, 2005.

[23] ROSENBERG, Jothy; REMY, David. Securing Web Services with WS-Security. Canada: Sams Publishing, 2004.

[24] ROUSAN, Mohammad AL; RJOUB, A.; AHMAD, Baset. A Low-Energy Security Algorithm for Exchanging Information in Wireless Sensor Networks, 2008.

[25] LABIOD, Houda; AFIFI, Hossam; SANTIS, Costantino De. Wi-Fi, Bluetooth, ZigBee and WiMax. The Netherlands: Springer, 2007.

[26] SCHOABA, Vagner; SIKANSI, Felipe; PIGATTO, Daniel F.; BRANCO, Kalinka R. L. J. C.; BRANCO, Luiz C. DISIMOD – Digital Signature for Mobile Devices. ICHIT, 2010.

Authors

Vagner Schoaba

M.Sc. in Computer Science at Univem - Marília/SP (2008). B.Sc. in Information Technology at Faculdades Associadas de Ariquemes (2005). Currently he is coordinator of information technology course at FAP-CE. His main research topics are: Computer Networks, Computer Security, Encryption and Digital Signature.

Felipe Eduardo Gomes Sikansi

B. Sc. Candidate at the Institute of Mathematics Sciences and Computer, Department of Computer Science, University of Sao Paulo, Sao Carlos, Brazil. His main research topics are: Computer Security, Encryption and Digital Signature.

Daniel Fernando Pigatto

M.Sc. Candidate in Computer Science and Computational Mathematics at Institute of Mathematics and Computer Science, University of Sao Paulo - ICMC-USP/Sao Carlos. B.Sc. in Computer Science at URI – Erechim/RS (2009). His main research topics are: Embedded systems, Performance Evaluation, Computer Security, Encryption and Digital Signature.

Kalinka Regina Lucas Jaquie Castelo Branco

Assistant Professor of the Institute of Mathematics and Computer Science - ICMC - USP, working in the department of Computer Systems. She has experience in Computer Science, with emphasis on Distributed Computing Systems and Parallel Computer, working mainly in the following areas: distributed systems, computer networks, security, performance evaluation and processes scheduling. She is member of Brazilian Computer Society.

Luiz Henrique Castelo Branco

Works at LG Electronics in Sao Paulo/SP. He has experience in Computer Science, working mainly in the following areas: Global Positioning System, Intelligent Transportation Systems, Mobile Applications.