2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops

Self-Proxy Mobile Signature

A new client-based mobile signature model

Mohammad Hasan Samadani, Mehdi Shajari, Mohammad Mehdi Ahaniha
Computer and IT Engineering Dept.
Amirkabir University of Technology (Tehran Polytechnic)
Tehran, Iran
{mhssamadani, shajari, mm.ahaniha}@aut.ac.ir

***Abstract*—The application of a fast and secure mobile signature model is an essential issue for the development of the mobile electronic commerce since digital signatures can provide authentication, data integrity, and non-repudiation. There are several technologies and models with the aim of implementing signature processes for mobile devices. In this paper, we categorize them into client-based and server-based models. We will comment on the most important properties of each solution and analyze the advantages and disadvantages, with a special focus on the private key security, performance of the signature generation process, and application of digital certificates. Furthermore, we present, analyze, and develop a new client-based mobile signature model, based on the concept of proxy certificates, which guarantees the security of user's private key as well as improving the speed of signature generation process. This model can be extended in order to use it for mobile partial identification, and also, to develop applications like secure mobile auctions that need several signature generations in a short period of time.**

SPMS, Mobile signature, digital certificate, proxy certificate, SIM card

I. INTRODUCTION

Most of the mobile commerce applications or services need to use some security services to guarantee the safety of the transactions that they perform. The most important security services are authentication, data integrity, and non-repudiation, because of providing essential evidences proving participation of the particular user in a specific transaction. Authentication, data integrity, and non-repudiation properties can be satisfied using digital signatures and digital certificates.

Mobile applications, in order to satisfy their security requirements, may need one or more digital signatures in a specific period of time. For example, a mobile banking protocol proposed by Li et al., [1] needs one digital signature in each of its transactions, but a secure mobile auction application may need to send several bids in a short period of time. Therefore, it needs to generate several signatures in that period. Hence, the delay and length of signing process is very important in many types of mobile applications.

The private key protection and computational costs are the main challenges of efficiently deploying of digital signatures in m-commerce. In the first generation of mobile clients, it was hard (or not supported) to generate digital signatures due to the limited cryptographic and computational capabilities of these clients. However, currently mobile clients are able to generate digital signatures based on asymmetric cryptography [2].

There are many models and approaches to generate digital signature in mobile environments. We can classify them according to several criteria such as signature platform, technologies, standards and supported features [2]. Furthermore, we can classify mobile signatures based on where to generate the signature. Based on this, there are two possible mobile signing approaches: client-based and server-based mobile signatures.

In this paper, we discuss client-based and server-based mobile signatures. We compare these models with a special focus on the performance and the private key security. In the rest of this paper, we present, develop and analyze a new client-based mobile signature generation model, called the Self-Proxy Mobile Signature (SPMS) model. This model guarantees the security of private key and the speed of signature generation. The presented model can be used in mobile applications that need several mobile signatures in a short period of time, e.g. stock marketing, pervasive environments and mobile auctions.

The rest of this paper is organized as follows. Section II and III generally discuss server and client-based mobile signature approaches, and summarize the related works. Section IV presents the comparison of different signature models, and discusses the requirements of a secure and fast mobile signature model. Section V presents Self-Proxy Mobile Signature which satisfies these requirements. The architecture of the SPMS model is presented in section VI. In section VII the model analysis and the test results are presented. Other model alternatives are discussed in section VIII. Section IX presents some examples of the usage of the SMPS model. Finally, section X concludes the paper and outlines future work.

II. SERVER-BASED MOBILE SIGNATURES

A server-based digital signature is a signature created by a signature service provider for a mobile client. In general, this model has been proposed for those devices that have not enough computational resources. This type of signatures is categorized into certificate-based and certificate-less signatures.

978-0-7695-4019-1/10 $26.00 © 2010 IEEE
DOI 10.1109/WAINA.2010.125

A. *Certificate based server side mobile signatures*

Three types of certificates can be used in certificate based server side mobile signatures: the client's certificate, server's certificate, and proxy certificate. Next we explain them.

1) *Server-based signatures with client's certificates*

By using client's certificate, the client's private key will be revealed to the server. Therefore, this type of signature cannot be considered legally equivalent to the handwritten signature of the client [3]. Figure 1 illustrates such a server-based signature. This model is like the SET Wallet Server model [4].

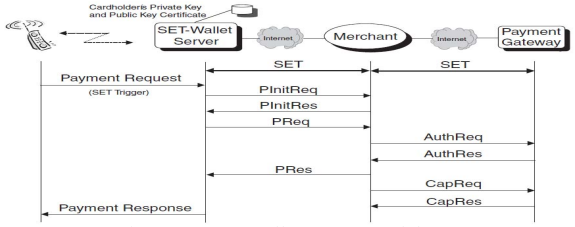

Figure 1. SET-Wallet Server model [4]

2) *Server-based signatures with server's certificates*

The second type of server-based signatures is produced using the certificates issued to the service provider. In fact, the signature service provider acts as a replacement for the client. Based on the signature of the provider, it cannot be verified that the client really authorized the signature and so this type of signature cannot be legally equivalent to the handwritten signature [3]. Figure 2 illustrates this type of server-based signature generation.

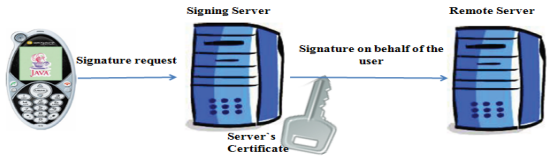

Figure 2. Server-based model with server's certificate

Chen et al. [5] proposed a server-based mobile signature scheme that uses server's certificate. After that, Wang et al. [6] proposed a server-based mobile signature that can be categorized as this type of server-based signatures.

3) *Server-based signatures with proxy certificates*

This type of server-based mobile signatures is based on delegating the signing power to mobile agents using proxy certificates [7]. In this case, the user generates a key pair and the corresponding proxy certificate for the mobile agent and sends it to the service provider. In the remote server, the mobile agent has a private key and a proxy certificate. Therefore, it can sign on behalf of the client. This signature can be assumed as the client's proxy signature.

In this scheme, the user's private key is not revealed to the server. However, an attacker can attack the agent and, therefore, the agent's private key may be revealed or misused. Furthermore, the agent will be forced to generate illegal signatures. To solve this vulnerability, the validity period of proxy certificate is kept short and its signing capabilities and the number of times that the certificate can be used to sign are limited. These mechanisms can reduce the vulnerability of proxy signature and costs of misusing. However, they cannot solve the problem completely [8].

Romao and Silva [9] used proxy certificates to delegate signature capability to a mobile agent. After that, Bamasak and Zhang [10] proposed a secure method for signature delegation to mobile agents. Their work is different from other related proxy signature schemes in that in addition to providing confidentiality protection to the proxy key, the method provides non-repudiation services to all the parties involved. Finally, Ou and Ou [8] used proxy certificates to delegate signature capability to a mobile agent in a mobile payment system.

B. *Certificate-less server side mobile signatures*

A certificate-less server-based mobile signatures consist of those signatures that are not based on regular signing algorithms like RSA and DSA. These innovative algorithms are based on dividing the signature generation computations in such a way that a signature is generated only with the cooperation of both client and server and the client's computational load is very low.

Bicakci and Baykal [11] proposed the SAOTS signature model for pervasive computing. After that, they proposed Improved Server Assisted Signature [12]. Later, He and Zhang [13] proposed a novel server-based signature protocol called Joint Signature, which improved to the Improved Joint Signature (IJS) [14]. The signature generated by IJS could be assumed as a joint signature of client and server which is generated by the server as according to user request. Furthermore, Lei et al. [15] proposed the SBS model to generate the signature in a remote server for mobile devices. The SBS protocol does not use the general public/private key and standard digital certificates. Furthermore, Ding et al. [16] proposed the SAS signature model, which relies on partially trusted servers to generate signatures for regular mobile users.

In addition to the key establishing problem, the most important drawback of these protocols is that they must be supported by the sender, recipient and server. Moreover, these protocols do not support the digital certificates and so cannot be used widely.

III. CLIENT-BASED MOBILE SIGNATURES

In the first generation of mobile devices, it was hard to generate digital signatures due to the limited cryptographic and computational capabilities of these devices. However, currently the mobile clients are able to generate digital signatures based on asymmetric cryptography [2, 21].

A mobile client consists of a mobile device and a SIM card. In short, a SIM card is a smart card with an application which implements the GSM11.11 specification.

As both the mobile device and the SIM card can store the keys and generate the signatures, we can assume three models for client-based signing based on key storing and signature generation location(s).

A. SIM-based signature

In the SIM-based model both of key storing and signature generation operations are carried out within the SIM card. Figure 3 illustrates this model.

As the SIM card has a very secure environment, this model guarantees the security of keys. The private key will never leave from the SIM. Therefore, it will not be revealed to anyone [2, 3]. However, the SIM cards are slower than mobile devices in generating digital signatures [2]. The invocation time of the SIM cards by the mobile device is also significant [17]. Therefore, this model is very secure, but it is slow.

The SATSA-PKI-based signature [18], WIM application-based signature [2], and Handset-based SET Wallet [4] are important examples of this type of signatures.

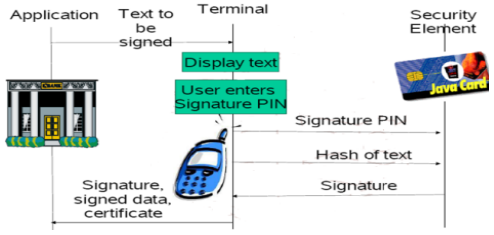

Figure 3. SIM based model

B. Device-based signature

Another way to handle mobile digital signatures is the device based signature model. In this model, both the key storing and signature generation operations happen in the device. Storing the keys in the file system or some other storage locations in the device is not very secure because of security weaknesses of mobile devices and their operation systems. However, as the CPU of mobile devices is more powerful than SIM cards, the signature generation is faster. Therefore, this model is very fast, but it is also quite vulnerable at the same time. Figure 4 illustrates this model.

There are several libraries and tools that enable a mobile device to generate digital signatures. The most important libraries are Bouncy Castle [19], IAIK Micro Edition [20], and SATSA-CRYPTO package [18].

Figure 4. Device based model

C. Hybrid signature

There are a large number of services that cannot be implemented completely in the SIM card side or in the device side. These services usually need to take advantage of the device characteristics (rich user interface and high processing capabilities) as well as the security of the SIM card [2].

In the hybrid model the keys are stored in the SIM card, but the processing of digital signature generation takes place in the device. As the keys are necessary for signing, in the time of signature generation, the keys have to leave the SIM card and be revealed to the device. This method is almost as fast as the device-based signature. However, with regards to key security, it is less secure than the SIM-based method and more secure than the device-based signature. Figure 5 illustrates hybrid signature model.

This type of mobile signature is proper when the SIM card is only capable to store the keys, and it cannot generate digital signatures.

Figure 5. Hybrid model

IV. COMPARISON OF MOBILE SIGNATURE METHODS

In the previous sections of this paper, we have discussed about two types of mobile signatures, the server-based and client-based models. As mentioned earlier, the server-based mobile signatures are not promising signatures because of their shortcomings and drawbacks [3].

Ruiz-Martínez et al. [2, 21] and Rossnagel [3] have mentioned that the future mobile signatures must be client-based. As mentioned previously, there are three types of client-based mobile signatures in three different levels of security and performance. The SIM-based one is the most secure as well as the most time consuming one. The device based model is very fast, but is not very secure because of vulnerability of the private key. The other one, the hybrid model, is almost as fast as device based model but with higher security. However, it is less secure than the SIM based model because the private key must be revealed to the device and, therefore, an attacker can take advantage of the revealed key.

There are some kinds of m-commerce applications that need several digital signatures in a short period of time, e.g. mobile auctions [22]. These types of applications need fast, low delay and secure signatures. As mentioned before, the current mobile signature models are suffering from security or performance. Therefore, none of previously discussed mobile signature models is secure and fast enough to be used for these types of mobile applications.

V. OUR PROPOSED METHOD (THE SELF-PROXY MOBILE SIGNATURE)

In this section we present a new client-based mobile signature, based on the concept of the proxy certificates. The proposed method, called the self-proxy mobile signature, guarantees the security of private key and speed of signature generation. It also offers some other beneficial features, such as lesser use of the user's private key, more signature flexibility and, possibility of using different key sizes.

A. *Proxy certificate*

Credential delegation and single sign-on are some of the most interesting features of Grid Security Infrastructure [23]. They are achieved using a special type of certificates, called *proxy certificates* [24].

The proxy certificate allows the holder of the certificate to act on user's behalf. In fact, it is very similar to the X.509 digital certificates, except that it is not signed by a Certificate Authority. Indeed, a proxy certificate is signed by an end user [25].

Although we have centered on the advantages of proxy certificates for delegation, these certificates have other features that make them suitable for other purposes. For example, they can be used locally: generating a proxy certificate that authorizes the user to act on her behalf. This is very helpful since the user can use the proxy certificate for all her secure conversations, instead of using her public/private key pair directly. This reduces the risk of having user's conversations compromised because, an attacker would only have a chance to crack the proxy's key pair, and not the user's personal one which would only be used to generate the proxy certificate [25].

We can be sure that the certificate is comes from user by checking its signature and user's certificate. The process of validating a proxy certificate is practically identical to the process of validating an ordinary certificate (you only need an additional certificate validation). The main difference is that the proxy certificate is not signed by a Certificate Authority. Indeed, it is signed by a user.

B. *The Self-Proxy Mobile Signature model (SPMS)*

As mentioned before, a model that the user's private key is stored in the secure environment of SIM card and the signature generation process is performed in the powerful environment of mobile device is a fast and low delay mobile digital signature. However, because the private key of the user must be leave from the SIM card, this model is vulnerable. The hybrid model that we discussed before is based on this idea.

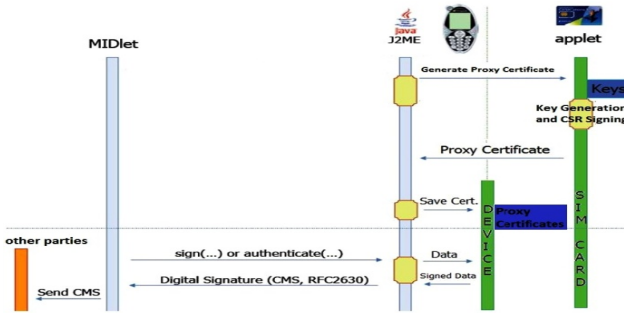

Figure 6. Self- Proxy Mobile Signature Model

In this section we propose a secure hybrid model based on this idea as a secure alternative. This model is called the *Self-Proxy Mobile Signature (SPMS)* model in this paper.

The *SPMS* model is based on the proxy certificates. In this model the user's private key is stored in the secure environment of SIM card but the process of signature generation is performed in the powerful environment of mobile device. On the basis of proxy certificate concept, in our model, the SIM card acts as a Certificate Authority for the mobile device and issues a proxy certificate to the device. Thus, the user's private key will never leave the SIM card, and instead, the proxy's private key is used for signature generation in the mobile device. Figure 6 illustrates the SPMS model.

1) *SPMS workflow*

In this section we discuss the SPMS model in detail. We can clearly distinguish two completely separated phases in this model, namely the proxy certificate issuing phase and the signature generation phase.

a) *Proxy certificate issuing phase*

In the proxy issuing phase the SIM card acts as a Certificate Authority and issues a proxy certificate to the device, according to the request of the device.

This phase consists of key generation and certificate signing operations. These operations take place in the SIM card.

Key generation is a time consuming operation and will take a long time to be executed in the SIM card. However, this phase might be performed offline. Therefore, it has no negative effect on the time of online signature generation.

When the SIM card has the keys, it can generate a proxy certificate for the device to delegate the signature generation power to the device. This certificate and its associated private key are passed to the device and stored in a proper location. After that, the device can sign the data on behalf of the SIM card.

The proxy certificate issuing phase can be executed on different basis. Either a regular basis, e.g. daily, etc., or on an irregular basis, e.g. per transaction, network, application, server, or completely based on the user's request.

A mobile user can have several proxy certificates available at the same time for different usages. For example, a user could have three proxy certificates: one for her mobile auction application, another for the mailing service and, finally, the last one for daily usage. In this case, the mobile auction application's proxy certificate might be renewed on per auction basis, the mailing service's certificate on the user's demand basis and the daily usage certificate on a regular daily basis.

It is also to point out that these certificates might have different key length, validity periods and constraints. This will depend on application and/or user's preferences.

b) *Digital signature generation*

In the signature generation phase, the device has the power of signature generation on behalf of the SIM card. The process of proxy signature generation is similar to usual signature generation in the device, but it uses proxy keys and certificates instead of user's regular key and certificate.

2) *Signature verification*

The process of signature verification is performed in the signature receiver side and is similar to regular signature verification with one more stage. In this extra stage, the verifier must check the signature of proxy certificate and ensure that the user has actually delegated her signature power to her device.

VI. SYSTEM ARCHITECTURE

Pisko [26] proposed a new architecture for mobile services which integrates the use of mobile signatures. This architecture comprises the whole mobile signature service as an application unit on mobile devices.

We can integrate our model with this architecture by adding the proxy manager module to the Mobile Service Application. Furthermore, the Java Card Applet must be modified to fulfill the requirements of our model. Figure 7 illustrates the integrated architecture.

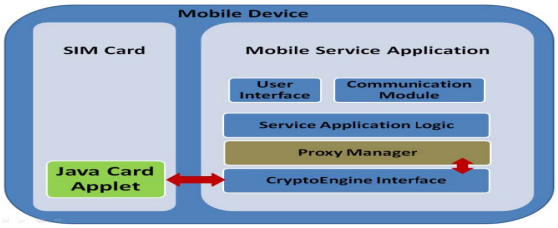

Figure 7. The SPMS model integrated with mobile service application architecture

VII. ANALYSIS AND PERFORMANCE EVALUATION

A. Performance analysis

We developed the proposed model as two separated components.

The first component was developed as a midlet in the Java Mobile Edition (Java ME) MIDP 2.1 [27]. This midlet was signed with a code-signing certificate and installed on a Nokia S40v5 edition device [28]. This component is responsible for managing generated proxy certificates, requesting new proxy certificates and, communicating with the second component.

The second component was developed as a Java Card applet and installed on a Java card from Infineon SLE88 family [29]. This part acts as a CA and is responsible for issuing proxy certificates. For communication between these two components, we have make use of the SATSA-APDU package [18].

Table 1. The time of cryptographic operations in the SIM card (ms)

Operation	1024 bit	2048 bit
RSA key generation	8000	20000
RSA signature generation	2700	6000

Table 2. The time of signature generation in the mobile device (ms)

Operation	1024 bit	2048 bit
RSA signature generation	<100	<400

Table 3. The time of operations of the SPMS model (ms)

Phase	Operation	1024 bit	2048 bit	Location
Issuing a new proxy certificate	RSA key generation	8000	20000	SIM card
	Certificate Signing (RSA1024 bit Signature generation)	2700	2700[a]	
	total	<10700	<22700	
Signing	RSA signature generation	< 100	< 400	Mobile device

a. the user's private key is 1024 bit, but the newly generated proxy certificate has 2048 bit key length

Table 1 shows the time of cryptographic operations in the used SIM card. Table 2 shows the time of RSA signature generation in the selected mobile device. Table 3 shows the time of operations of the SPMS model. Note that these times are the whole operation time, i.e. the times include the invocation time, data transmission time and computation time. As the results show, using the SPMS model enables the user to generate signatures faster than SIM based model.

B. Security analysis

In the SPMS, the user's private key is stored in the secure environment of the SIM card. This key will never leave the SIM card, and therefore, will never be revealed to anyone. The proxy private key, that is used to sign in the mobile device, has a short validity period with some constraints. This key is protected by a secret code. Therefore, the SPMS model is more secured than device-based and hybrid mobile signature methods. Moreover, this model is more secure than server-based mobile signature with proxy certificates, because nothing will ever leaves the user's device. However, the SPMS model is less secure than the SIM-based mobile signature, because the proxy private key is stored in the device, and in the signing time is revealed to the device.

C. Other features

Using the SPMS model has some other useful features:

1- Reduced usage of the user's private key: The user's private key is only used to sign the proxy certificate.
2- Lower risk of the key compromising: This model reduces the risk of having user's conversations compromised, because an attacker would only have a chance to crack the proxy's key pair, and not the user's personal one which would be used only to generate the proxy certificate.
3- Larger keys: The proxy certificates could have different key lengths that are not supported by the SIM card, or they are very large for the SIM card to handle efficiently.
4- Different key lengths: The key length of proxy certificates may be different in respect of their usage requirements.
5- Optimization of the signature algorithm: If the signature procedure is performed in the device, it is affordable to optimize the signing algorithm to achieve better performance.
6- Reducing the need to very fast advanced SIM cards.
7- No need to change the SIM for new certificates: In some situations, when a newly issued certificate must be copied on the SIM card, the SIM card must be changed. In the SPMS model, these newly issued certificates can be derived from user's personal certificate as a proxy certificate. This will reduce the need to change the SIM card.

VIII. MODEL ALTERNATIVES

The discussed SPMS model stores the proxy certificates in the device, and then, uses them in a similar way to the device-based mobile signatures. However, this model can be

treated in two other different ways. First, we can store the proxy certificates in the SIM card and treat as the SIM-based signature model. This case does not speed up the signature process, and also, has a certificate issuing overhead. But, this increases the security and safety of the user's private key, because of very lower use of it. Second, both of the user's certificate and the proxy certificate can be stored in the device. This will reduce the compromising risk of the user's private key.

IX. MODEL USAGE

There are several applications and usages that can benefit from the concept of self-proxy certificates. Here, we present two different scenarios of applications of this model.

A. Secure Mobile Auction

A secure mobile auction application needs to send several non-repudiable bids in an auction. As the auction nears to be closed, the time space between the bids gets shorter, and therefore, the bid generation process must be performed as soon as possible. This model can be used by secure mobile auction applications to generate signatures with very low delay and very fast.

B. Mobile partial identification

A mobile device can be used as an identification tool [30]. As an extension for the SPMS model, the generated proxy can hold a partial non-repudiable profile of the user. Indeed, the user can assign a partial set of her identity attributes to the generated proxy. This will enhance the privacy of the user.

X. CONCLUSION

Delegating the signing power to a more powerful entity enables the weak user to use digital signatures as she needs. However, the security risks of this delegation must be considered.

In this paper, we presented, developed and analyzed the SPMS model which uses the delegation concept locally. In this model, the user's SIM card delegates its signing power to the user's mobile device using proxy certificates. This will increase the efficiency and flexibility of mobile digital signatures without exiting the user's private key from the SIM card. The presented model can be used in several kinds of application, especially those that need several signature generations in a transaction, e.g. in mobile auction application as well as applications like the mobile partial identification can be developed based on this concept.

Our further work on this topic will be aimed at the development of a secure mobile auction application, and a mobile partial identification application as a proof of concept and to check its features.

XI. ACKNOWLEDGMENTS

This work was supported by Iran Telecommunication Research Center. We must also be thankful of Antonio Ruiz-Martínez for his valuable comments and notes during this project.

REFERENCES

[1] D. Li, D. Lin, G. Zhao, and B. Huang, “Design and correctness proof of a security protocol for mobile banking”, Bell Labs Tech. Journal, vol. 14(1), 2009, pp. 259-266, Wiley Periodicals.

[2] A. Ruiz-Martínez, D. Sánchez-Martínez, M. Martínez-Montesinos, A.F. Gómez-Skameta, “A survey of electronic signature solutions in mobile devices”, Theoretical and applied electronic commerce research,vol.2,issue3, 2007, pp. 94-109.

[3] H. Rossnagel, “Mobile qualified electronic signatures and certification on demand”, in Proc. of the 1st European PKI Workshop, Samos, Greece, June 2004.

[4] D. O'Mahony, M. Peirce and H. Tewary, “Electronic payment systems for e-commerce”, Artech House, 2001, pp. 302-325.

[5] C.L. Chen, C.C. Chen, L.C. Liu, G. Horng, “A server-aided signature scheme for mobile commerce”, in Proc. of the International Wireless Communications and Mobile Computing Conference (IWCMC07), ACM, 2007, USA.

[6] Z. Wang, Z. Guo, Y. Wang, “Security research on J2ME-based mobile payment”, Proc. of ISECS Int. Colloquium on Computing, Communication, Control, and Management, 2008, pp. 644-648.

[7] J. Claessens, B. Preneel, J. Vandewalle, “(How) Can mobile agents do secure electronic transactions on untrusted hosts?”, ACM Trans. On Internet Technology, vol.3,No.1,2003, pp.28-48.

[8] C.M. Ou and C.R. Ou, “Adaptation of proxy certificates to non-repudiation protocol of agent-based mobile payment systems”, Applied Intelligence, v.30 n.3, 2009, p.233-243.

[9] A. Romao and M. Mira da Silva, “Secure mobile agent digital signatures with proxy certificates”, E-commerce Agents, LNAI 2033, Springer, 2001, pp. 206-220.

[10] O. Bamasak and N. Zahng, “A secure method for signature delegation to mobile agents”, ACM Symposium on Applied Computing (SAC04), ACM, Mar. 2004, pp. 813-818.

[11] K. Bicakci and N. Baykal. “SAOTS: A New Efficient Server Assisted Signature Scheme for Pervasive Computing”, LNCS No. 2802, Germany, 2003.

[12] K. Bicakci and N. Baykal. “Improved Server Assisted Signatures”, Journal of Computer Networks, Elsevier, 2005, vol. 47, No. 3 pp. 351-366.

[13] L. He, and N. Zhang, “A New Signature Scheme: Joint Signature”, Proceedings of the 2004 ACM symposium on Applied computing, Cyprus, 2004, pp. 807-828.

[14] L. He, N. Zhang, L. He, I. Rogers, “Secure m-commerce transactions: a third party based signature protocol”, third int. Symposium on Information Assurance and Security, IEEE, 2007.

[15] Y. Lei, D. Chen, Z. Jiang, “Generating digital signatures on mobile devices”, Proc. 18th International Conference on Advanced Information Networking and Application , IEEE, 2004.

[16] X. Ding, D. Mazzocchi, G. Tsudik, “Equipping smart devices with public key signatures”, ACM trans. On Internet Technology, vol. 7, No. 1, Article 3, Feb. 2007.

[17] S.T. Chanson, T.W. Cheung, “Design and implementation of a PKI-based end-to-end secure infrastructure for mobile e-commerce”, World Wide Web, vol. 4, 2001, pp. 235-253.

[18] Sun, "Security and Trust Services API for J2ME (SATSA)", available at: http://java.sun.com/j2me/docs/satsa-dg

[19] Bouncy Castle, http://www.bouncycastle.org

[20] IAIK JCE Provider, http://jce.iaik.tugraz.at

[21] A. Ruiz-Martínez, Daniel Sánchez-Martínez, María Martínez-Montensiones, A. F. Gómez-Skarmeta. “Mobile Signature Solutions for Guaranteeing Non-Repudiation in Mobile Business and Mobile Commerce”. Mobile and Ubiquitous Commerce: Advanced E-Business Methods: Volume 4 of Advances in Electronic Business Series. IGI Global Publishers. May 2009.

[22] http://en.wikipedia.org/wiki/Mobile_commerce

[23] http://www.globus.org/security/overview.html

[24] S. Tuecke et al, "Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile", http://www.ietf.org/rfc/rfc3820.txt

[25] http://gdp.globus.org/gt4-tutorial/multiplehtml/ch10s05.html

[26] E. Pisko, “Mobile electronic signatures: progression from mobile service to mobile application unit”, Proc. of 6th Int. Conf. on the Management of Mobile Business (ICMB07), IEEE, 2007 .

[27] MIDP2, http://java.sun.com/products/midp/

[28] Nokia devices, http://www.forum.nokia.com/devices

[29] Infineon co., http://www.infineon.com

[30] G. Muller, S. Wohlgemuth, “Study on mobile identity management”, FIDIS – Future of Identity in the Information Society, deliverable 3.3, May 2005.