Secure SMS Communications for M-Learning Services

CĂTĂLIN BOJA, PAUL POCATILU, ALIN ZAMFIROIU
Department of Economic Informatics
Bucharest Academy of Economic Studies
Piata Romana 6, Bucharest
ROMANIA
catalin.boja@ie.ase.ro, ppaul@ase.ro, alinz10@yahoo.com

Abstract: - Education processes have evolved in the last decades by integrating new technology. The m-learning field has been defined new educational services that deliver knowledge and content on mobile devices, highlighting a dreamed educational concept, anywhere, anytime. Many m-learning services use SMS to deliver simple text content because it has a low cost, it is available on all mobile devices and has wide coverage and availability. As more and more sensitive data is processed by m-learning services, we propose a solution that will secure the SMS content and that will have minimal impact on the device performance. The solution implements a symmetric encryption scheme and is used by a distributed m-learning architecture.

Key-Words: Security, Cryptography, Mobile, Message, Text, SMS

1 Introduction
One of the most used facilities offered by the mobile devices to communicate, with over 6.1 trillion messages in 2010 [1], is short written messages. For that, phone companies provide communication through Short Message Service (SMS). These forms of communication are widely used in adverse conditions such as those in which there is a lot of noise or verbal communication is not allowed. Communication via messages is preferred for privacy reasons. Conversation voice could be heard by others and for practical reasons and noise could hinder the conversation. Many people choose to use text messages that are cheaper than voice call.

Since the definition of digital natives concept by Marc Prensky, [17] and the analysis of the impact of technology on the youth daily activities, research has been conducted on using this technology to increase the quality and the output of the educational process. A distinct approach is to use mobile devices to deliver educational content in anytime, anywhere scenario. Today, this scenario is something that can be achieved with small effort because:

- the majority of students use in daily activities mobile devices to socialize, search data on Internet, play, download multimedia content;
- the costs needed to acquire mobile technology decreased and also the lowest device specifications include data connections capabilities through Wi-Fi, Bluetooth and IR; it is estimated, [1] that until 2015 the Internet will be accessed by more mobile devices than desktop computers.

M-learning services are provided through standalone Java or Windows Mobile applications or in distributed environments, using different communication technologies like, Bluetooth in PicoNets, data connections for accessing Web applications or SMS [9]. Despite a large pool of communications technologies that provides mobile connections, m-learning solutions are restricted by the cost of using such technologies. The cost can be represented by the fees of using data connections, SMS services, Wi-Fi private networks, or by the availability of the service in a location, like a Wi-Fi or Bluetooth access point. Based on these pros and cons, the technology that is the cheapest and assures an anywhere coverage, hence assuring mobility, is the SMS service. These are the reasons for which, many mobile learning, social, banking and health services are based on SMS. Furthermore the service cost supported by the user can be more reduced by collaborations between the mobile carrier and the mobile services authority.
2. Secure M-Learning Services based on Short Message Service

M-Learning services based on SMS communication deliver in real time different announcements, information, alerts, tasks. Other use of SMS is to define a mobile assessment architecture based on short quiz tests. Students receive questions and send the answer also in a SMS. SMS services can be used to provide a content delivery architecture that responds to requests send by text messages.

Security of sending SMS is becoming increasingly important when there is a risk of the interception of information or the data is sensitive like in a mobile banking service. The proposed SMSEncrypt solution solves the problem of message security through advanced and standardized symmetric encryption algorithms [5]. It uses symmetric encryption based on a secret key known by both parties. The disadvantage of this approach is in transmission of passwords between the two communication parties. The channel used to transmit the encryption/decryption key must be a secure one [13]. The Advanced Encryption Standard (AES) is also known under the name Rijndael. This is a standard algorithm for symmetric encryption, adopted as a standard, by the US National Institute of Standards and Technology (NIST) which defines data security procedures in governmental institutions, [5]. The algorithm, if it used correctly, has not been broken. It uses different size keys 128, 192, or 256 bits and the latter is still very difficult to break even with today technology.

The proposed solution, SMSEncrypt, described in figure 1, is composed of two parts:

- the sender has a user key – UK used to encrypt the plaintext text message – M; in order to provide message integrity, the encrypted message contains the message hash;
- the receiver, which is another user or a service in the distributed m-learning architecture, uses the same key, UK, to decrypt the SMS; once the content is obtained, the receiver checks the message hash and responds if it is a service; because a symmetric key encryption solution is vulnerable in distributing the key to multiple users, the proposed m-learning architecture manages users and their associated keys;

Fig. 1. The architecture of communicating through short encrypted messages

Encryption key and decryption key must be the same and should be known only by his message sender and receiver.

To send a message sender goes through the following steps:

- writes the message;
- specify the message encryption key; the key is uniquely associated with this user;
- specifies the recipient;
- the application generated a hash value based on the message text; the hash is used to allow integrity check; the hash is added to the initial message and the content is encrypted;

To receive a message and read it, the receiver goes through the following steps:

- the encrypted message is sent over the network;
- the application or the M-Learning service intercepts the message;
- the sender is identified;
- the sender key is retrieved from a repository and is used to decrypt the cipher text;
- the integrity of the message is verified and the plaintext message is delivered.

These steps are shown in Figure 2.
The quality of the m-learning service is directly influenced by the content quality and structure, and by the application quality. Regarding application quality there are defined multiple sets of quality characteristics [18]. The impact of a SMS based service on the mobile device performance and its power consumption is minimal. Also, the encryption stage has been analyzed and the results have shown that:

- generating a MD5 hash value is faster than a SHA-1;
- using a symmetric algorithm requires a less overhead than an asymmetric one, like RSA, that requires a public key – private key pair;

3. Technical solution

In order to receive the encrypted message in a manner that allows the user to decrypt the message, the solution is based on message interception techniques:

- for Java ME platform, the same solution is using the Security and Trust Services API (SATSA) [8].

In order to support cryptographic services, Java ME platform includes a package, the Security and Trust Services API (SATSA), [8], that is flexible enough to run with many types of cryptographic algorithms and protocols. The SATSA framework has been designed to run on any Java ME-based virtual machine, including the CDC and CLDC virtual machines. This Java standard specification has been defined by the Java Community Process (JCP) in JSR 177 [11].

The API provides interfaces that allow developers to implement secure solutions based on a smart card, the mobile device or a combination of the two. This survey concentrates only on the second solution, using only the mobile device processing unit, because there are other restrictions, legal and technical, that will not allow a smart card solution intended for a wide range of devices.

From all the SATSA packages, the one that does not require a smart card is the SATSA-CRYPTO package. It provides classes for implementing data security architectures based on message digests, digital signatures and symmetric and asymmetric encryption/decryption algorithms.

Classes from SATSA packages like javax.crypto and javax.crypto.spec.SecretKeySpec provides the tools for symmetric encryption:

```java
Cipher ecipher;
byte[] b = "SECRET_1SECRET_2SECRET_3".getBytes();
ecipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
```
ecipher.init(Cipher.ENCRYPT_MODE, new
SecretKeySpec(b, 0, b.length, "AES");

or for hash functions:

MessageDigest md;
md = MessageDigest.getInstance("MD5");

.NET CF includes the System.Security.Cryptography namespace. .NET CF supports SHA1, TripleDES, DSA, RSA and other algorithms. Also, platform invocation of native CryptoAPI functions and third party libraries can be used.

For example, the class MD5CryptoServiceProvider is used for MD5 hash and the class and the class RSACryptoServiceProvider for RSA encryption.

.NET CF provides the a namespace that include the MessageInterceptor class, used to intercept SMS messages based on user condition: phone number, sender name, text content etc. The message interception is managed through Outlook Mobile API. The OutlookSession class is associated to the OutlookMobile application.

For example, the following code is used to intercept a SMS message and to define the function that will receive the message.

using Microsoft.WindowsMobile.PocketOutlook;
MessageInterception;
OutlookSession outlookSession =
    new OutlookSession();
MessageInterceptor mi =
    new MessageInterceptor();
mi.InterceptionAction =
    InterceptionAction.Notify;
mi.MessageCondition =
    new MessageCondition(
        MessageProperty.Sender,
        "+0711123456");

mi.MessageReceived+=new
MessageInterceptorEventHandler(mi_Mess
ageReceived);

The function mi_MessageReceived will receive the message bases on criteria and the implementation is used to decrypt message:

void mi_MessageReceived(object sender,
MessageInterceptorEventArgs e)
{
    SmsMessage message =

4 Conclusions
Text encryption for sensitive data (e.g. personal information, passwords, and marks) is very important for distributed m-learning applications. Data optimization and processing speed are crucial factors to distributed m-learning applications success and these have to be taken into account.

Both Java ME and .NET CF provides mechanisms for data encryption and message interception.

Nevertheless the limitations of mobile devices are not taking away any of the benefits of m-learning. Mobile devices industry is progressing at very fast pace, current limitations will be overcome and m-learning applications will be enhanced even further.

The next step is to test several implementations and to extend the solution to other platforms like Android, iPhone and Windows Phone.

Acknowledgement
This work was supported by CNCSIS –UEFISCSEU, project number PNII – IDEI 2637/2008, project title: Project management methodologies for the development of mobile applications in the educational system.

References:
Encryption-in-C.aspx