Cryptographic Measurements on Java-Enabled Mobile Phones*

Norbert Bátfai\textsuperscript{a}, Péter Molnár\textsuperscript{b}, Bálint Rábai\textsuperscript{c}, István Tari\textsuperscript{d}

\textsuperscript{a}University of Debrecen, Department of Information Technology
e-mail: batfai.norbert@inf.unideb.hu

\textsuperscript{b}University and National Library University of Debrecen
e-mail: pmolnar@lib.unideb.hu

\textsuperscript{c}University of Debrecen, Faculty of Informatics
e-mail: rabai.balintos@vipmail.hu

\textsuperscript{d}University of Debrecen, Faculty of Informatics
e-mail: diablosteven@hotmail.com

Abstract

In this article we present a hybrid cryptographic web application with Java ME MIDP mobile clients using Bouncy Castle Crypto API. During the testing of application we have measured the running time of functional parts of the program on different Java-enabled mobile phones. This project is implemented as a Debrecen Developer Network (DDN) project and our work is partly supported by project TARIPAR3.

\textit{Keywords:} Java ME, Bouncy Castle, RSA, Debrecen Developer Network

\textit{MSC:} 94-04 Explicit machine computation and programs, 94A60 Cryptography

1. Introduction

In the beginning of 2010, the Debrecen Developer Network (DDN) was created within the framework of the first author's PhD dissertation Department of Information Technology of University of Debrecen [10], [11]. The area of mobile developments represents a key priority of this young student community of our university. A such application is presented in this paper.

*This development is supported by project TARIPAR3 granted by the Hungarian National Office for Research and Technology.
1.1. Previous and Related Work

We have experience and a growing number of references in developing mobile applications. For example, these are presented in the following work and papers: [6], [7], [8], [9], [10], [11], [12] and [9]. The development a simple web application with mobile clients using Bouncy Castle library was introduced in the technical article of Motorola [1]. Our application differs from the example mentioned above in that it entirely works a symmetric way in both the server and the client. Our developed hybrid cryptographic web application was presented at the conference ICAI 2010, the 8th International Conference on Applied Informatics.

Mobile cryptography has a huge literature, to mention but a few references, see [2], [3], [4] or [5].

![Figure 1: The use case of the developed application.]

![Figure 2: The use case of exchanging encrypted messages on the mobile client.]

2. The Measurements

The developed hybrid cryptographic web application provides the possibility of measuring of the running time of any of its functional parts. In this work we focus only the timing of RSA key generation on different real mobile phones.
Figure 3: The use case of receiving an encrypted message on the mobile client.

Figure 4: The use case of sending an encrypted message on the mobile client

2.1. The Developed Application

The functional parts of our application can well be seen in the use case diagrams shown in Figure 3 and 4. In these figures, the parts in question are presented in a...
stepwise fashion. The UML activity diagrams are shown Figure 5 and 6.

2.2. Devices and Software Used

Tests were run on the following phones: Nokia E51, 6212 Classic, 5000, 2600c, 6600, 5310 XpressMusic, Samsung C3050, Motorola V3, K1, Razr2 V8, U9 and Sony Ericson W580i, C702. These phones are all Java enabled. To be more precise, these are Java ME MIDP (Java Micro Edition, Mobile Information Device Profile) devices. We have used the Bouncy Castle Crypto package for Java ME (Crypto-j2me-143). It can be downloaded from the project page [http://www.bouncycastle.org/latest_releases.html](http://www.bouncycastle.org/latest_releases.html).

The server-side is implemented as a simple Java Servlet with both Sun GlassFish Enterprise Server v3 Prelude and Apache Tomcat 6.0.20. These are well known and open source solutions in server side.

For the development, we have used the NetBeans IDE 6.7.1, JDK 6 Update 16 and Java ME SDK 3.0. During the testing, the following tools were also used: Sun Java Wireless Toolkit for CLDC 2.5.2_01, Sony Ericsson SDK 2.5.0.5 for the Java ME Platform, Motorola Java ME SDK 6.4 and Series 40 Nokia 6212 NFC.
3. The Results

The measurement results are shown in Table 1. Here, \( c \) denotes the certainty in the sense of \( \texttt{http://java.sun.com/javase/7/docs/api/java/math/BigInteger.html#isProbablePrime(int)} \) and \( n\text{bits} \) means the number of bits of keys. During the testing, we took five different measurements with given certainty parameters. As it can be seen from the Table 1, the generating time is increasing with increasing parameter values. There are variant results in the same parameter values, it follows from the properties of the used prime test (in Bouncy Castle). Table 1 shows the average of the time of generating the key pairs.

The averaged measurement results are summarized in diagram form in Figure 7.
Table 1: The measurement results of RSA key generation on some real mobile phones.
Cryptographic Measurements on Java-Enabled Mobile Phones

(a) The average measurements of the Nokia E51.
(b) The average measurements of the MotorolaRazr2 V8.
(c) The average measurements of the Sony Ericson W580i.

Figure 7: The average measurements.

4. Conclusions and further work

In this work, we have pointed out that RSA keys can be generated effectively by some mobile phones of our time. In the further work, we are porting the test application in question to Google’s Android mobile platform.

References


[13] Bátfai, Norbert, Mobil programozás, Nehogy már megint a mobilod nyomkodjon Téged!, Kempelen Farkas Student Digital Library, for the present in manuscript, (2010).

Norbert Bátfai
Hungary, 4032 Debrecen, Egyetem tér 1.

Péter Molnár
Hungary, 4032 Debrecen, Egyetem tér 1.

Bálint Rábai
Hungary, 4032 Debrecen, Egyetem tér 1.

István Tari
Hungary, 4032 Debrecen, Egyetem tér 1.