Proceedings of the 8th International Conference on Applied Informatics
Eger, Hungary, January 27–30, 2010. Vol. 1. pp. 303–310.

Cryptographic Measurements on Java-Enabled Mobile Phones*

Norbert Bátfai[a], Péter Molnár[b], Bálint Rábai[c], István Tari[d]

[a]University of Debrecen, Department of Information Technology
e-mail: batfai.norbert@inf.unideb.hu

[b]University and National Library University of Debrecen
e-mail: pmolnar@lib.unideb.hu

[c]University of Debrecen, Faculty of Informatics
e-mail: rabai.balintos@vipmail.hu

[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.

Keywords: Java ME, Bouncy Castle, RSA, Debrecen Developer Network

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].

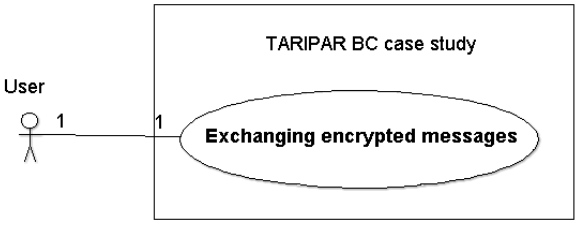


Figure 1: The use case of the developed application.

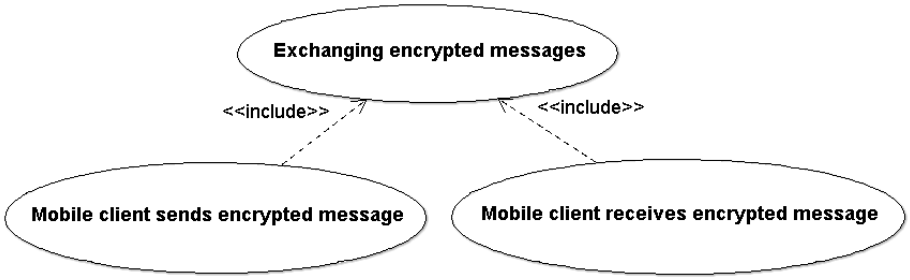


Figure 2: The use case of excahanging encrypted messages on the mobile client.

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.

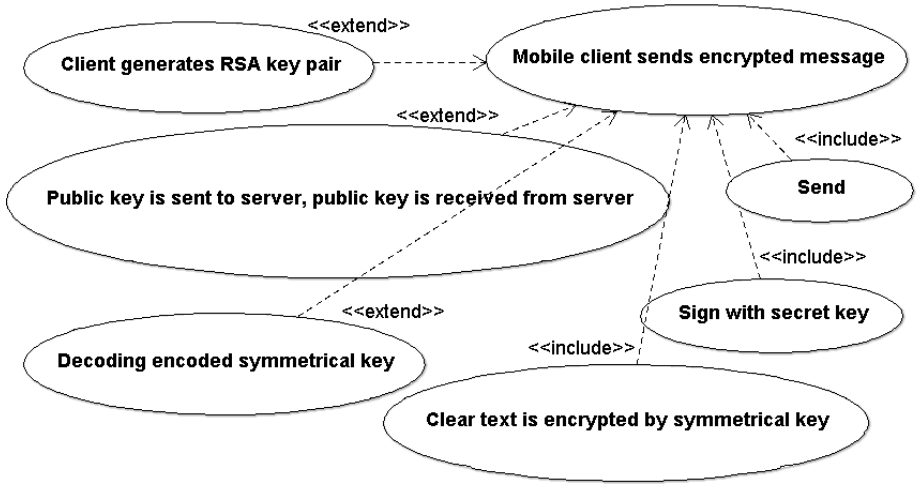


Figure 3: The use case of receiving an encrypted message on the mobile client.

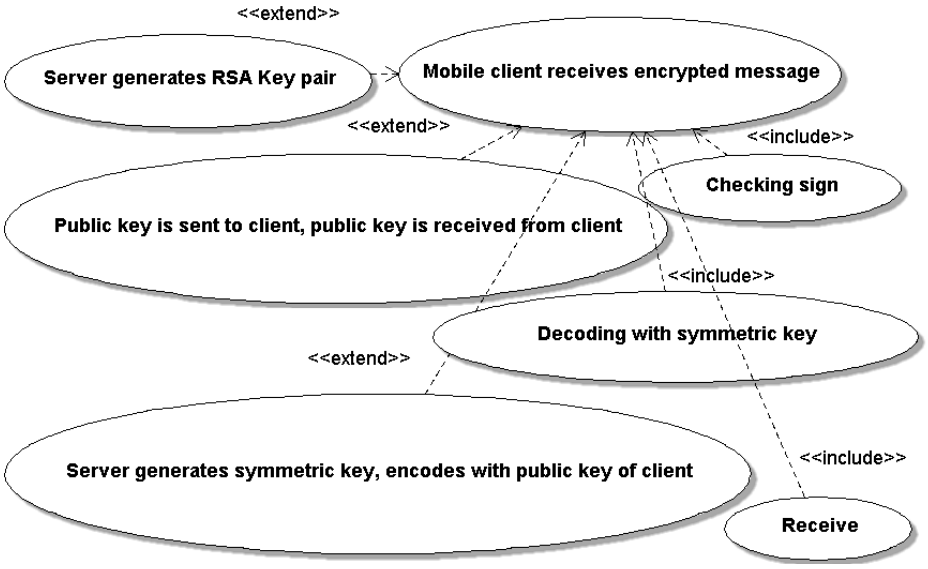


Figure 4: The use case of sending an encrypted message on the mobile client

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

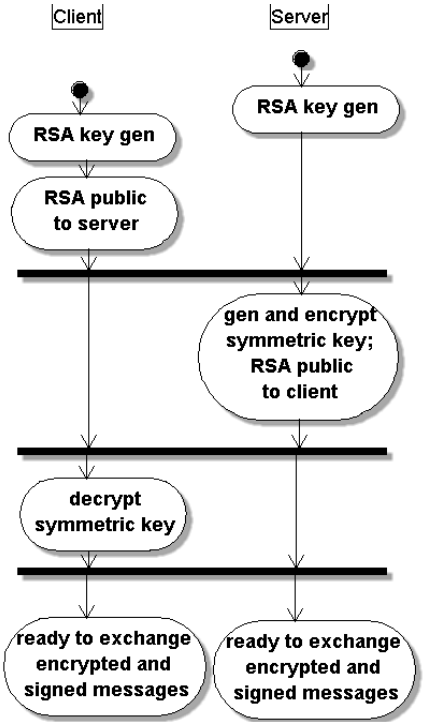


Figure 5: The activity diagram of building a connection.

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`.

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*

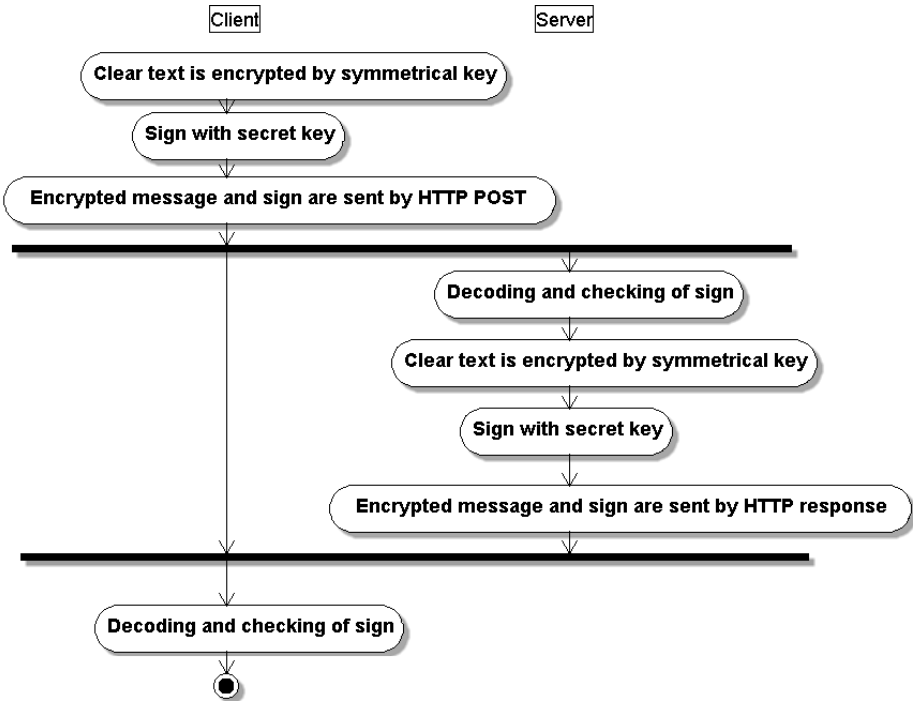


Figure 6: The activity diagram of exchanging messages.

SDK. Further details can be found at `http://www.inf.unideb.hu/~nbatfai/k/TARIPAR_BN.pdf`.

3. The Results

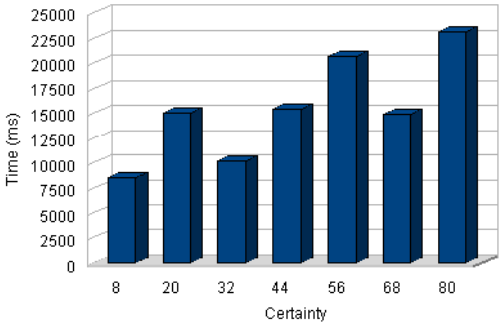
The measurement results are shown in Table 1. Here, c denotes the certainty in the sense of `http://java.sun.com/javase/7/docs/api/java/math/BigInteger.html#isProbablePrime(int)`and *nbits* means the number of bits of keys. During the testing, we took five different measurements with given certainty 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.

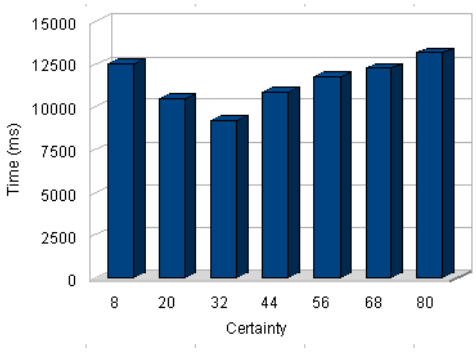
Nokia E51			MotorolaRazr2 V8			Sony Ericson W580i		
c	nbits	[ms]	c	nbits	[ms]	c	nbits	[ms]

8	1024	11016	8	1024	15878	8	1024	37741
8	1024	6351	8	1024	17108	8	1024	24193
8	1024	11483	8	1024	9344	8	1024	85468
8	1024	5814	8	1024	13157	8	1024	60325
8	1024	7657	8	1024	7232	8	1024	128730
20	1024	16507	20	1024	10993	20	1024	63869
20	1024	34630	20	1024	15886	20	1024	111665
20	1024	11200	20	1024	8602	20	1024	123841
20	1024	9590	20	1024	9468	20	1024	174916
20	1024	3061	20	1024	7469	20	1024	264951
32	1024	6568	32	1024	9562	32	1024	274493
32	1024	7720	32	1024	16915	32	1024	41928
32	1024	11272	32	1024	9369	32	1024	76933
32	1024	17874	32	1024	4695	32	1024	29315
32	1024	7442	32	1024	5444	32	1024	301531
44	1024	23921	44	1024	15516	44	1024	319736
44	1024	8704	44	1024	16385	44	1024	47083
44	1024	8863	44	1024	10550	44	1024	73242
44	1024	7320	44	1024	7450	44	1024	83186
44	1024	28506	44	1024	4137	44	1024	77818
56	1024	13781	56	1024	10149	56	1024	72695
56	1024	8354	56	1024	5929	56	1024	99792
56	1024	7340	56	1024	24212	56	1024	137171
56	1024	16732	56	1024	10980	56	1024	85026
56	1024	56901	56	1024	7607	56	1024	140985
68	1024	29491	68	1024	14728	68	1024	137654
68	1024	13801	68	1024	8213	68	1024	492774
68	1024	10381	68	1024	11460	68	1024	391591
68	1024	13198	68	1024	8850	68	1024	192462
68	1024	7798	68	1024	18165	68	1024	214158
80	1024	21671	80	1024	12696	80	1024	203674
80	1024	11954	80	1024	18384	80	1024	227494
80	1024	11202	80	1024	14715	80	1024	315002
80	1024	42570	80	1024	9345	80	1024	615518
80	1024	28369	80	1024	10730	80	1024	506218

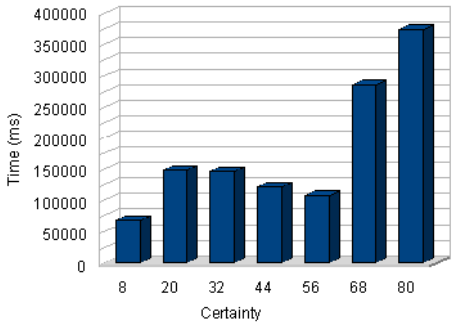
Table 1: The measurement results of RSA key generation on some real mobile phones.



(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

[1] Motodev.com, Using crypto apis for secure communications, (2010),`http://developer.motorola.com/docstools/articles/crypto_apis/`.

[2] Kuaté, Pierre Henri and Lo, Johnny Li-Chang and Bishop, Judith, Secure Asynchronous Communication for Mobile Devices, *WUP '09: Proceedings of the Warm Up Workshop for ACM/IEEE ICSE 2010*, (2009), 5–8.

[3] Stefan Tillich and Johann Groschädl, A Survey of Public-Key Cryptography on J2ME-Enabled Mobile Devices, *Lecture Notes in Computer Science*, Vol. 3280 (2004), 935–944

[4] Neil Daswani, Cryptographic Execution Time for WTLS Handshakes on Palm OS Devices, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.4179&rep=rep1&type=pdf.

[5] Peter Langendoerfer, Zoya Dyka, Oliver Maye and Rolf Kraemer, A Low Power Security Architecture for Mobile Commerce, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.72.3423&rep=rep1&type=pdf.

[6] Bátfai, N., Mobiltelefonos játékok tervezése és fejlesztése (Mobile Game Design and Development, hungarian), *PhD Dissertation and Thesis*, (2010), http://www.inf.unideb.hu/~nbatfai/phd

[7] Bátfai, N., Footballer and Football Simulation Markup Language and related Simulation Software Development, *Journal of Computer Science and Control Systems*, accepted, (2010).

[8] Bátfai, N., Open Source Mobile Games for Education (Conference lecture), *8th International Conference on Applied Informatics, Eger*, (2010).

[9] N. Bátfai, P. Molnár, B. Rábai, I Tari, Cryptographic measurements on Java-enabled mobile phones (Conference lecture), *8th International Conference on Applied Informatics, Eger*, (2010).

[10] Bátfai, N., Bátfai, E., Pšenáková, I., Jávácska One: Open Source Mobile Games to Revolutionize Education of Programming, *Teaching Mathematics and Computer Science*, submitted, (2010).

[11] Bátfai, N., Bátfai, E., A mobil játékfejlesztés elméleti és gyakorlati momentumai (Theoretic and practical issues in m-game development, hungarian), *Híradástechnika*, Vol. 5, (2005), http://www.hiradastechnika.hu/data/upload/file/2005/2005_5/HT_0505-7.pdf.

[12] Bátfai, Norbert, Nehogy már a mobilod nyomkodjon Téged!, *DEENK, Debrecen*, (2010), http://www.eurosmobil.hu/NehogyMar.

[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.