Annales Mathematicae et Informaticae
34 *(2007) pp. 61–70*
http://www.ektf.hu/tanszek/matematika/ami

Cryptographycal protocols in the Egerfood Information System*

Kálmán Liptai, Gábor Kusper, Tibor Radványi

EKF, Institute of Mathematics and Informatics
e-mail: liptaik@ektf.hu, gkusper@aries.ektf.hu, dream@aries.ektf.hu

Submitted 28 September 2007; Accepted 16 November 2007

Abstract

In this article we present the cryptography of the food safety tracking system of the Regional Knowledge Center (EGERFOOD), which can be found in Eger, Hungary at the Eszterházy Károly College. We analyzed its requirements for the underlying information system. To build a user friendly system, which serves quickly and cost effectively the costumers, the providers, and the effected authorities by information, is a complex task. Furthermore, the system has to fulfill the strict requirements which one put up for data-safety and -encryption in case of a tracking system. We considered also these ones by setting up the EGERFOOD information model.

Keywords: food safety, tracking, information systems, cryptography, AES-128, RSA, .NET, framework

MSC: 94A60, 68P30, 68N15, 68P25

1. Introduction

In the focus of the research-service activities nowadays stays the environment protection and the food analytical research, from which the most attractive (from viewpoint of R+D, economy, and society) works deal with the food analytical and food safety.

We have understood this at the Eszterházy Károly College and we have decided to setup a food safety and analytical monitoring center.

*Research supported by the Hungarian National Foundation for Scientific Research Grant. No. T 048945 MAT.

Before that in Hungary there were only segregated attempts to boost the safety some well-known products. However, these detached examinations could not achieve a new quality. There was no new quality- or safety-parameter introduction. However, there were some results but these ones were not yet integrated in a coherent and comprehensive food tracking system. In Hungary there is so far only one food tracking system, which is for red pepper. There was not nearly any attempt to adapt this system for other products.

Our goal regarding informatics: Building a system, which sits the costumer in the center, and which is able to process and send food safety information (firstly) to the costumers and (secondly) to the food producers and to the effected authorities in a fast, cost effective, and reliable way.

The information technology is a really important tool in every aspects of the project. From the communication that exists between the collaborating partners, through the food tracking system, to the food safety communication with the customers. There will be tasks for both the device developers (for example for solving the signal transferring problems) and the software developers (for example the internet based framework of the food tracking system).

Tasks of information technology are: Create and support continuously the web system, which operates the inner communication of the project. Its goal is to ensure the information-flow between those who work in the project.

Determine the structure of the food tracking database and create the hardware and software sides of the data transmission system. The backbone of the informatics system is the database of the food tracking system. We analyzed the collected data and requirements, and by that, we created the data model of the information system. [8]

We keep the connection with the customers through WAP and Internet. Install the data-collector hardware devices at the involved food provider companies. We connect the new gauging devices - which are in experimental stage - into the communication network of the project.

2. The construction of the information system

Now the system follows the lifecycle of 1-1 product of 6 companies and connects the results of the laboratory of the knowledgecenter into the system. These are the data sources. The main aims can be seen in Figure 1.

The side of outgoing data is layered. The public competence level can be reached through the internet or WAP. It is used to give information about the production and the origin of the product using an identification code. The protected competence level gives information to the participants of the project, whose ring is much narrower than the data of the public competence level. These data are useable in research and in the development of the production of the product. The inner competence level shows the companies' exclusive, inner used data. These data can be used by an ERP system.

Now let us see the construction of the system. So called bufferservers are

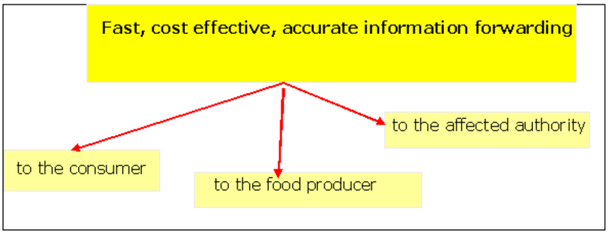

Fast, cost effective, accurate information forwarding
to the consumer
to the food producer
to the affected authority

Figure 1: Information forwarding

installed in every outer company and in research laboratory. Their tasks are the followings:

- store the members' data that belongs to the Egerfood project (e.g. corpus, meaning results, etc.)
- decode the incoming data
- process and store the incoming data
- encrypt the data and send it to the central data storehouse (through VPN connection)
- store backup copies.

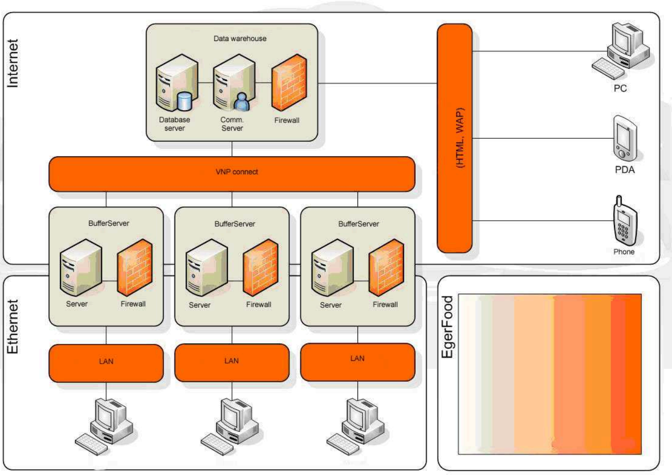


Figure 2: The informatical system

When we developed the information system of the project, the insurance of the suitable data security got a stressed function. For this reason, a three-storied encryption system came to development. In this way, starting from the creation of the data, every data is encoded by the algorithm AES-128. [1] (We shall detail the selection later.) When we send data we use the most modern method used in software technology, called Windows Communication Foundation, which makes encrypted communication. [4] The network data communication happens through a VPN network, so we can exploit the encryption provided by the VPN routers.

The base of the software system is the database developed by the researchers of this project, which make it possible that we can easily integrate any product of any company into the system.

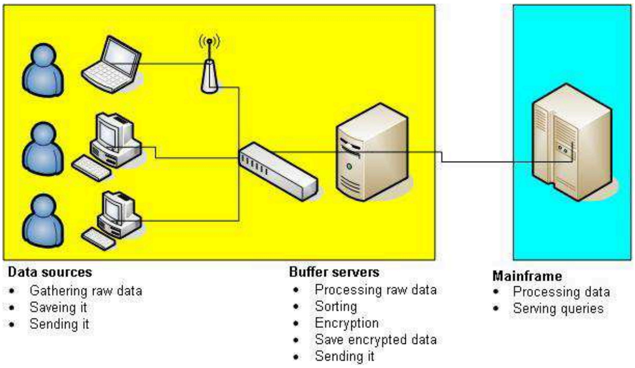


Figure 3: Puffer server

We made a program, called working process graph maker and analyser program, which output is a mass of data which generate the user surface of the clientprogram automatically. As a result of it, later enlarging can be done faster and smoothly, and the maintenance can perform with universal methods.

3. About the Encryption

Public networks such as the Internet do not provide a means of secure communication between entities. Communication over such networks is susceptible to being read or even modified by unauthorized third parties. In addition to file encryption and encryption on a local disk, cryptography helps you to create a secure means of communication over (otherwise insecure) channels, providing data integrity and authentication.

3.1. Possibility of the developer environment, the Microsoft's .NET 3.0

Cryptographic primitive	**Use**
a. Secret-key encryption (symmetric cryptography)	Performs a transformation on data, keeping the data from being read by third parties. This type of encryption uses a single shared, secret key to encrypt and decrypt data.
b. Public-key encryption (asymmetric cryptography)	Performs a transformation on data, keeping the data from being read by third parties. This type of encryption uses a public/private key pair to encrypt and decrypt data.
c. Cryptographic signing	Helps verify that data originates from a specific party by creating a digital signature that is unique to that party. This process also uses hash functions.
d. Cryptographic hashes	Maps data from any length to a fixed-length byte sequence. Hashes are statistically unique; a different two-byte sequence will not hash to the same value.

a. Symmetric cryptography

DESCryptoServiceProvider This algorithm supports a key length of 64 bits.

RC2CryptoServiceProvider The RC2CryptoServiceProvider implementation supports key lengths from 40 bits to 128 bits in increments of 8 bits.

RijndaelManaged This algorithm supports key lengths of 128, 192, or 256 bits.

TripleDESCryptoServiceProvider This algorithm supports key lengths from 128 bits to 192 bits in increments of 64 bits.

b. Asymmetric cryptography

DSACryptoServiceProvider You can use the DSACryptoServiceProvider class to create digital signatures and protect the integrity of your data. To use a public-key system to digitally sign a message, the sender first applies a hash function to the message to create a message digest. This algorithm supports key lengths from 512 bits to 1024 bits in increments of 64 bits.

RSACryptoServiceProvider This is the default implementation of RSA. The RSACryptoServiceProvider supports key lengths from 384 bits to 16384 bits in increments of 8 bits if you have the Microsoft Enhanced Cryptographic Provider installed. It supports key lengths from 384 bits to 512 bits in increments of 8 bits if you have the Microsoft Base Cryptographic Provider installed.

4. Why did we choose the AES?

We paid our attention the Advanced Encryption Standard (AES) announced in 2001 published by Joan Daemen and Vincent Rijmen [1] and the other was RSA published in 1976 by Ron Rivest, Adi Shamir and Leonard Adleman at MIT.

The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. The AES algorithm is capable of using cryptographic keys of 128, 192 and 256 bits to encrypt and decrypt in blocks of 128 bits. Usually an implementation of AES algorithm supports least one of the three key lengths. The algorithm specified in this standard has been implemented in different languages. In bibliographies there are lots of suggestions on how to efficiently implement the AES algorithms on a variety of platforms. We have a 32 bit system so we chose the AES-128 version to try. In this paper we don't want to detail the whole process, but we mention the main steps. When we use the AES standard we follow the following main steps.

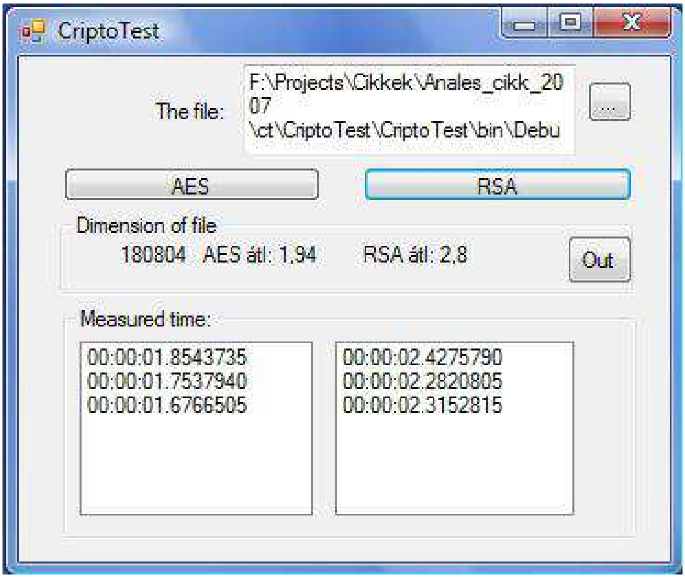
1. A non-linear substitution step where each byte is replaced with another according to a given table (SubBytes).
2. A transposition step where each row of the state is shifted cyclically a certain number of steps (ShiftRows).
3. A mixing operation which operates on the columns of the state, combining the four bytes in each column (MixColumns).
4. Each byte of the state is combined with the round key, each round key is derived from the cipher key using a key schedule (AddRoundKey).

These points are completed with an Initial Round and a Final Round, where we use the previous steps with slightly modification. In our case we repeat the rounds ten times (one of them is the Final Round).

The RSA cryptosystem is based on two mathematical problems, the problem of factoring large numbers and discrete logarithm problem. It is known that RSA is much slower than DES and other symmetric cryptosystem, but we investigated this fact in our case.

We analysed our choice with a program. [5] We wanted to know how long does it take to encrypt files with different size in one hand with the algorithm AES, on the other hand with algorithm RSA. Both of the algorithms are implemented in the system of framework 2.0 and 3.0. the surface of the test program is easy to use.

On this picture can be seen that after a file is selected the program execute three times both the AES and RSA encryptions, and measures the passed time. [7] Than calculates the arithmetic mean of it. The measured results can be saved into a file with one only click. A size of the file and the averaged times belongs to it.[2][3] This program was used in computer with following configuration:



The file: F:\Projects\Cikkek\Anales_cikk_2007\ct\CriptoTest\CriptoTest\bin\Debu

Dimension of file
180804 AES átl: 1,94 RSA átl: 2,8

Measured time:

```
00:00:01.8543735        00:00:02.4275790
00:00:01.7537940        00:00:02.2820805
00:00:01.6766505        00:00:02.3152815
```

Figure 4: The test program

Intel i915G; Pentium4 530 (3GHz, 1Mb L2 cache, 800MHz FSB) processzor; 512Mb (2x256Mb) 400MHz dual channel DDR RAM; 160Gb SATA HDD, MS Windows XP Home operation system.

We got this table as a result of measure. The items are represented in a diagram. [8] It can be seen very well that the curve belongs to the RSA stays above the curve belongs to the AES during the time of measuring. It can be clearly seen that the algorithm RSA needs more time to encrypt the same sized file.

The results got approached with a power function. Let f be a function which shows how many time need to encrypt a file the function of file size in the case of algorithm RSA. Let g be a function which shows how many time need to encrypt the function of file size in the case of algorithm AES. We get

$$f(x) = 1,340686 \cdot 10^{-10} \cdot x^{2,003325}$$
$$g(x) = 1,518159 \cdot 10^{-10} \cdot x^{1,975274}$$

We got the result that we expected beforehand so that the symmetrical key algorithm AES is more efficient in our case and with the growth of the size of the encrypted file is more conspicuous. We found substantial differences so we gave up applying RSA in our system because the quickness is very important point of view of the companies. We remark that we had chance to chose the key in AES-128 but we left this for the implemented program. Our data can be seen in following table and figures:

Size (byte)	AES (sec)	RSA (sec)
116362	2,01	2,307
136356	2,125	2,244
156352	2,229	3,322
176346	3,208	4,302
196342	4	5,265
216336	5,223	6,52
236332	6,255	7,229
256326	7,104	9,14
276320	8,317	10,244
296316	9,234	12
316328	11,328	14,307
326444	12,26	15,276
326444	12,62	15,156
345474	13,71	17,584
365470	15,32	18,834
405460	18,312	23,297
445450	22,148	28,1

Table 1: Measured data

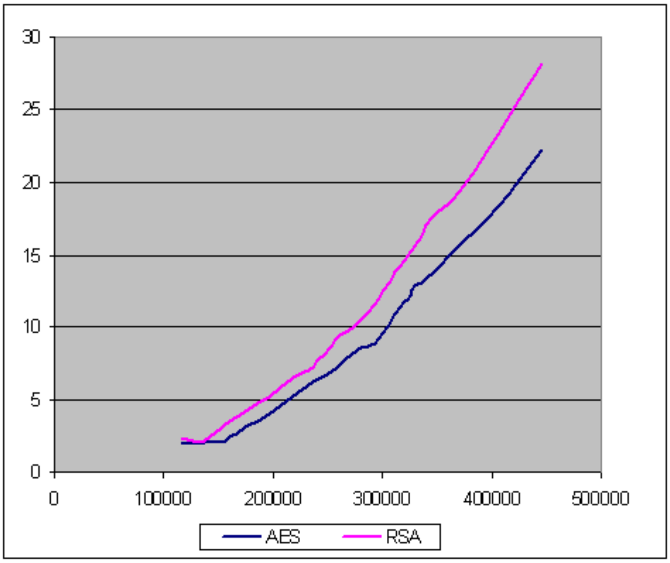


Figure 5: Measured data on the graph

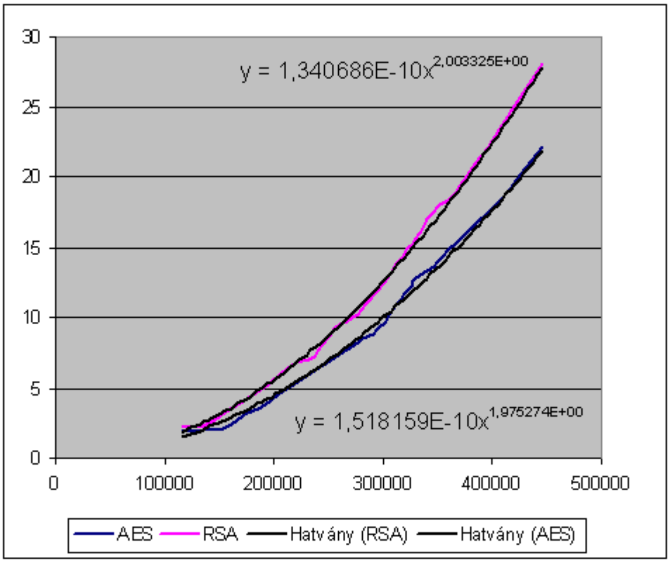


Figure 6: The interpolated curves

References

[1] Announcing the Advanced encryption standard (AES), *Federal Information Processing Standards Publication* 197, 2001.

[2] Schneier, B., Applied Cryptography, *John Wiley & Sons*, 1996.

[3] Menezes, A.J., van Oorschot, P.C., Vanstone, S.A., Handbook of Applied Cryptography, *CRC Press* 1996.

[4] Microsoft: Improving .NET Application Performance and Scalability, (2004), 639–682.

[5] Tsai, M., Kulkarni, C., Sauer, C., Shah, N., Kurt Keutzer University of California, Berkeley, Infineon Technologies, CPR ST, Munich A Benchmarking Methodology for Network Processors 1st Network Processor Workshop, 8th Int. Symp. on High Performance Computer Architectures (HPCA), Feb. 3rd 2002 Boston, MA.

[6] Radványi, T., Kusper, G., Requirement Analyses and a Database Model for the Project EGERFOOD Food Safety Knowledge Center, *ICAI2007*, Eger Hungary.

[7] Baraani-Dastjerdi, A., Pieprzyk, J., Safavi-Naini, R., Getta, J.R., Using Cryptographic Hash Functions for Discretionary Access Control in Object-Oriented Databases, *Journal of Universal Computer Science*, vol. 3, no. 6 (1997), 730–753.

[8] Radványi, T., Examination of the MSSQL Server from the user'point of view considering data insertion, *Acta Academiae Paedagogicae Agriensis, Sec. Mathematicae* (2004), 69–77.

Kálmán Liptai
Gábor Kusper
Tibor Radványi
Eszterházy Károly College
Institute of Mathematics and Informatics
P.O. Box 43
H-3300 Eger
Hungary