International Journal of
Applied Mathematics,
Electronics and Computers
ISSN: 2147-8228 www.atscience.org/IJAMEC

Advanced Technology and Science

Original Research Paper

Audio Message Transmitter Secured Through Elliptical Curve Cryptosystem[#]

Artan Luma *[1], Besnik Selimi , Lirim Ameti

Accepted 15th August 2014

Abstract: Securing a communication is always a challenge for participants in it. A lot of applications available in the market claim to enable secure audio communication, but not always show the details of the technology used behind to encrypt the data. It is important for end users to understand the techniques used for encrypting the data, in order to trust it. Elliptic curve cryptography, an approach to public key cryptography, is now widely used in cryptographic systems. Hence, in this paper we propose using elliptic curve cryptography to secure the transmission of voice messages through the network. The system that is proposed and implemented enables the encryption of the voice message, its transmission through the network and its decryption at the other end.

Keywords: *privacy, cryptography, elliptic curves, voice message (.wav).*

1. Introduction

The success of the business processes today is closely linked to the efficiency of the privacy of communication between the parties. Our aim is to secure the transmission of voice messages. As mentioned in the abstract part, there are plenty of applications in the market who claim they enable a secure audio communication, without telling about the relying technology, and this makes us to suspect about their level of security.

Although known algorithms as RSA, are in wide use today, they are not suitable for use when dealing with large amounts of data upon which needs to be applied an efficient cryptographic system, always taking into account along with the privacy, the performance too.

Voice messages compared to text messages [1], contain a large amount of data and this is why the aforementioned algorithms are not efficient because of their performance.

Our proposed system addresses this issue. The elliptic curve cryptography is more than appropriate for achieving the goal. Compared to RSA algorithm, the efficiency of elliptic curve cryptography is stated as follows: "Safety of elliptical curves is based on elliptic curve discrete logarithm problem (ECDLP) which enables ECC to reach the same level of security with RSA for smaller keys and greater computation efficiency. ECC-160 provides security compared with RSA-1024 and ECC-224 provides security compared with RSA-2048 [2]". This fact is sufficient to build our cryptographic system based on elliptical curves, which is the main purpose this paper.

The rest of this paper is organized as follows: section 2 describes elliptical curve operations, section 3 describes the encryption and decryption with elliptical curves, section 4 describes the audio format (.wav) in a way as will be used in our system, section 5 explains the system setup and section 6 concludes this paper.

2. Elliptical Curve Operations

Elliptic curve operations which are of interest of our paper are: point addition, point subtraction, point doubling and point multiplication. For these operations to be faster, more accurate and more efficient, the elliptical curve cryptography is defined over two finite fields:

--Prime field F_p, where p is a prime,

--Binary field F_2^m, where m is a positive integer.

We use the prime field F_p, where as a case study we consider the following elliptical curve:

$$y^2 \equiv x^3 + x + 1 \ (mod\ 277), or\ E_{277}(1,1)$$

Elliptic curve operations as point addition, point subtraction, point doubling and point multiplication are defined as following:

2.1. Point addition

Consider two distinct points J and K such that $\mathrm{J} = (\mathrm{x_j}, \mathrm{y_j})$ and $\mathrm{K} = (\mathrm{x_k}, \mathrm{y_k})$.

Let $\mathrm{L} = \mathrm{J} + \mathrm{K}$, where $\mathrm{L} = (\mathrm{x_L}, \mathrm{y_L})$, then:

$$\mathrm{x_L} \equiv \mathrm{s}^2 - \mathrm{x_j} - \mathrm{x_k}\ (\mathrm{mod\ p})$$

$$\mathrm{y_L} \equiv -\mathrm{y_j} + \mathrm{s}(\mathrm{x_J} - \mathrm{x_L})(\mathrm{mod\ p})$$

$$\mathrm{s} \equiv (\mathrm{y_L} - \mathrm{y_K})/(\mathrm{x_J} - \mathrm{x_K})\ \ (\mathrm{mod\ p})$$

s is the slope of the line through J and K.

If $\mathrm{K} = -\mathrm{J}$, i.e. $\mathrm{K} \equiv (\mathrm{x_J}, -\mathrm{y_J})(\mathrm{mod\ p})$ then $\mathrm{J} + \mathrm{K} = \mathcal{O}$, where $\mathcal{O}$ is the point at infinity.

If $\mathrm{K} = \mathrm{J}$ then $\mathrm{J} + \mathrm{K} = 2 \cdot \mathrm{J}$ then point doubling operations are used. Also:

$$\mathrm{J} + \mathrm{K} = \mathrm{K} + \mathrm{J}$$

Let $\mathrm{J} = (1,130)$ and $\mathrm{K} = (4,30)$ then the point $\mathrm{L}(\mathrm{x_L}, \mathrm{y_L})$ can be calculated as:

First we calculate s:

[1] *Contemporary Sciences and Technologies, South East European University, Campus, 1200, Tetovo/Macedonia*
* *Corresponding Author: Email: a.luma@seeu.edu.mk*

[#] *This paper has been presented at the International Conference on Advanced Technology&Sciences (ICAT'14) held in Antalya (Turkey), August 12-15, 2014.*

$$s \equiv \frac{30-130}{4-1}\ (\text{mod } 277)$$

$$s \equiv \frac{-100}{3}\ (\text{mod } 277)$$

$$s \equiv -100 \cdot \frac{1}{3}\ (\text{mod } 277)$$

$$s \equiv (-100) \cdot (-92)\ (\text{mod } 277)$$

$$s \equiv 9200\ (\text{mod } 277)$$

$$s \equiv 59\ (\text{mod } 277)$$

then

$$x_L \equiv (59^2 - 1 - 4)\ (\text{mod } 277)$$

$$x_L \equiv (3481 - 5)\ (\text{mod } 277)$$

$$x_L \equiv 3476\ (\text{mod } 277)$$

$$x_L \equiv 152\ (\text{mod } 277)$$

and

$$y_L \equiv (59 \cdot (1 - 152) - 130)\ (\text{mod } 277)$$

$$y_L \equiv (59 \cdot (-151) - 130)(\text{ mod } 277)$$

$$y_L \equiv (-8909 - 130)\ (\text{mod } 277)$$

$$y_L \equiv -9039\ (\text{mod } 277)$$

$$y_L \equiv 102\ (\text{mod } 277)$$

Hence the result of point addition of $(1,130)$ and $(4,30)$ for the elliptic group $E_{277}(1,1)$ is $(152,102)$.

2.2. Point subtraction

Consider two distinct points J and K such that $J = (x_J, y_J)$ and $K = (x_K, y_K)$, then $J - K = J + (-K)$, where $-K \equiv (x_K, -y_K)(\text{mod } p)$.

Let $J = (1,130)$ and $K = (4,30)$, then $-K \equiv (4,-30) \equiv (4{,}247)(\text{mod } 277)$ and

$$L = J - K = J + (-K)$$

$$L = (1,130) + (4,247) = (131,63)$$

Hence with the subtraction of $J = (1,130)$ and $K = (4,30)$, i.e. $L = J - K$, as a result is gained the point $L = (131,63)$ which also lies in our elliptical curve.

2.3. Point doubling

Consider a point J such that $J = (x_J, y_J)$, where $y_J \neq 0$. Let $L = 2 \cdot J$, where $L = (x_L, y_L)$, then:

$$x_L \equiv s^2 - 2 \cdot x_J\ (\text{mod p})$$

$$y_L \equiv -y_J + s \cdot (x_J - x_L)(\text{mod p})$$

$$s \equiv (3 \cdot x_J^2 + a)/(2 \cdot y_J)\ \ (\text{mod p})$$

s is the tangent at point J and a is one of the parameters chosen with the elliptic curve.

If $y_J = 0$, then $2 \cdot J = O$, where O is the point at infinity. Let $J = (1,130)$ and $K = (4,30)$ then the point $L(xL, yL)$ can be calculated as:

Let $J = (1,130)$ and we calculate the point L, i.e. $L = 2 \cdot J$. Firstly we calculate s:

$$s \equiv \left(\frac{3 \cdot 1^2 + 1}{2 \cdot 130}\right)\ (\text{mod } 277)$$

$$s \equiv \left(\frac{4}{260}\right)(\text{mod } 277)$$

$$s \equiv \left(4 \cdot \frac{1}{260}\right)(\text{mod } 277)$$

$$s \equiv (4 \cdot 114)\ (\text{mod } 277)$$

$$s \equiv 456\ (\text{mod } 277)$$

$$s \equiv 179\ (\text{mod } 277)$$

then

$$x_L \equiv (179^2 - 2 \cdot 1)\ (\text{mod } 277)$$

$$x_L \equiv (32041 - 2)\ (\text{mod } 277)$$

$$x_L \equiv 32039\ (\text{mod } 277)$$

$$x_L \equiv 184\ (\text{mod } 277)$$

and

$$y_L \equiv \left(s \cdot (x_J - x_L) - y_J\right)(\text{mod } 277)$$

$$y_L \equiv (179 \cdot (1 - 184) - 130)\ (\text{mod } 277)$$

$$y_L \equiv (179 \cdot (-183) - 130)\ (\text{mod } 277)$$

$$y_L \equiv (-32757 - 130)\ (\text{mod } 277)$$

$$y_L \equiv -32887\ (\text{mod } 277)$$

$$y_L \equiv 76\ (\text{mod } 277)$$

Hence the result of doubling of the point $(1,130)$ for the group $E_{277}(1,1)$ is the point $(184,76)$ [3].

2.4. Point multiplication

We calculate point multiplication by combining point addition and point multiplication, an algorithm called as double-and-add which functions like following:

```
T = P
for i = t-1 downto 0
  T ≡ T+T (mod n)
      if di = 1 then
             T ≡ T + P (mod n)
      end if
  end for
return T
```

where P is a point in the elliptical curve, T is the variable where the result is stored, t is the binary width of the scalar which multiplies the point and d_i is the bit with the index i [4].

Consider the point $P(1,130)$ that lies in the curve. Let us take a scalar $d = 47$, i.e. $d = (101111)_2$ and $d_i = [1,1,1,1,0,1]$, then $T = d \cdot P$, i.e. $T = 47 \cdot (1,130)$. Based on the aforementioned algorithm are following the calculations:

$$T = (1,130)$$

$$i = 6 - 1 = 5,\ \ T = ((1,130) + (1,130))\ (\text{mod } 277)$$

$$d_5 = 1,\ \ T = ((184,76) + (1,130))\ (\text{mod } 277)$$

$$i = 5 - 1 = 4,\ \ T = ((67,3) + (67,3))\ (\text{mod } 277)$$

$$d_4 = 0,\ \ /$$

$i = 4 - 1 = 3, \quad T = ((103,73) + (103,73)) \pmod{277}$

$d_3 = 1, \quad T = ((228,100) + (1,130)) \pmod{277}$

$i = 3 - 1 = 2, \quad T = ((137,10) + (137,10)) \pmod{277}$

$d_2 = 1, \quad T = ((103,204) + (1,130)) \pmod{277}$

$i = 2 - 1 = 1, \quad T = ((60,1) + (60,1)) \pmod{277}$

$d_1 = 1, \quad T = ((158,97) + (1,130)) \pmod{277}$

$i = 1 - 1 = 0, \quad T = ((192,46) + (192,46)) \pmod{277}$

$d_0 = 1, \quad T = ((227,102) + (1,130)) \pmod{277}$

$T = (46,106)$

Hence the multiplication of point $P = (1,130)$ with the scalar $d = 47$, in the elliptical group $E_{277}(1,1)$ gives as a result the point $T = d \cdot P = (46,106)$.

3. ECC Encryption / Decryption

Elliptic curve cryptography can be used to encrypt a plaintext message, say M, into ciphertext. The plaintext message M is encoded into a point P_M from the finite set of points in the elliptic group, $E_P(a,b)$. The first step consists in choosing a generator point, $G \in E_P(a,b)$ such that the smaller value of n for which $n \cdot G = O$ is a very large prime number. The elliptic group $E_P(a,b)$ and the generator point G are made public.
Each user select a private key,$n_A < n$ and compute the public key $P_A = n_A \cdot G$. To encrypt the message point P_M for B, A choses a random integer k and compute the ciphertext pair of points P_C using B's public key P_B:

$$P_C = [(k \cdot G), (P_M + k \cdot P_B)]$$

After receiving the ciphertext pair of points P_C, B multiplies the first point, $(k \cdot G)$ with his private key n_B and then adds the result to the second point in the ciphertext pair of points, $(P_M + k \cdot P_B)$:

$$(P_M + k \cdot P_B) - (n_B \cdot k \cdot G) =$$
$$= (P_M + k \cdot n_B \cdot G) - (n_B \cdot k \cdot G) = P_M$$

which is the plaintext point, corresponding to the plaintext message M. Only B, knowing the private key n_B, can remove $n_B \cdot (k \cdot G)$ from the second point of the ciphertext pair of point, i.e. $(P_M + k \cdot P_B)$, and hence retrieve the plaintext information P_M [5].
Consider our elliptic curve:
$y^2 \equiv x^3 + x + 1 \pmod{277}$
That is $a = 1$, $b = 1$, $p = 277$. The elliptic curve group generated by the above elliptic curve is $E_p(a,b) = E_{277}(1,1)$.
Let the generator point $G = (0,276)$, then multiplies $k \cdot G$ of the generator point G are (for $1 \leq k \leq 277$):

$G = (0,276) \quad 2G = (208,105) \quad 3G = (72,220)$

$4G = (274,91) \quad 5G = (174,74) \quad 6G = (84,3)$

$7G = (117,35) \quad 8G = (47,157) \quad 9G = (146,241)$

$10G = (122,201) \quad \ldots\ldots\ldots \quad 274G = (121,180)$

$275G = (258,34) \quad 276G = (149,188) \quad 277G = (175,175)$

If A wants to send to B the message M which is encoded as the plaintext point $P_M = (18,158) \in E_{277}(1,1)$. A must use B's public key to encrypt it. Suppose that B's secret key is $n_B = 85$, then B's public key will be:

$P_B = n_B \cdot G = 85 \cdot (0,276)$

$P_B = (237,15)$

A selects a random number $k = 113$ and uses B's public key $P_B = (237,15)$ to encrypt the message point into the ciphertext pair of points:

$P_C = [(k \cdot G), (P_M + k \cdot P_B)]$

$P_C = [113 \cdot (0,276), (18,158) + 113 \cdot (237,15)]$

$P_C = [(260,67), (18,158) + (253,130)]$

$P_C = [(260,67), (68,178)]$

Upon receiving the ciphertext pair of points $P_C = [(260,67), (68,178)]$, B uses his private key $n_B = 85$, to compute the plaintext point P_M as follows:

$(P_M + k \cdot P_B) - [n_B \cdot (k \cdot G)] = (68,178) - [85 \cdot (260,67)]$

$(P_M + k \cdot P_B) - [n_B \cdot (k \cdot G)] = (68,178) - (253,130)$

$(P_M + k \cdot P_B) - [n_B \cdot (k \cdot G)] = (68,178) + (253,-130)$

because $-P = (xP, -yP)$

$(P_M + k \cdot P_B) - [n_B(k \cdot G)] = (68,178) + (253,147)$

because $-130 \equiv 147 \pmod{277}$

$(P_M + k \cdot P_B) - [n_B(k \cdot G)] = (18,158)$

and then maps the plaintext point $P_M = (18,158)$ back into the original plaintext message M.

4. Message Format

Since the purpose of the whole paper is to build a cryptosystem based on elliptical curves to secure the transmission of voice messages, in this section we will examine the integral structure of one particular file format (WAVE) upon which we will apply encryption and decryption.
The WAVE file format is a subset of Microsoft's RIFF specification for the storage of multimedia files. A RIFF file starts out with a file header followed by a sequence of data chunks. A WAVE file is often just a RIFF file with a single "WAVE" chunk which consists of two sub-chunks -- a "fmt" chunk specifying the data format and a "data" chunk containing the actual sample data [6].

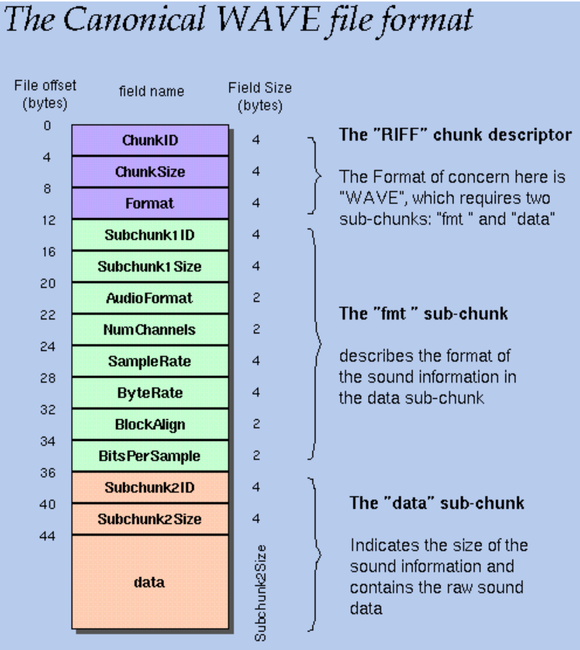
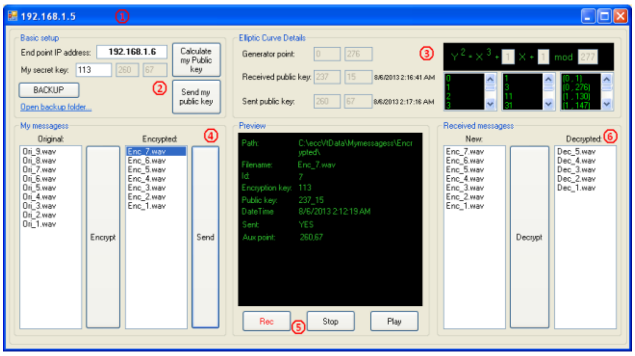

The Canonical WAVE file format
File offset (bytes)
field name
Field Size (bytes)
0
4
8
12
16
20
22
24
28
32
34
36
40
44
ChunkID
ChunkSize
Format
Subchunk1 ID
Subchunk1 Size
AudioFormat
NumChannels
SampleRate
ByteRate
BlockAlign
BitsPerSample
Subchunk2ID
Subchunk2Size
data
4
4
4
4
4
2
2
4
4
2
2
4
4
Subchunk2Size
The "RIFF" chunk descriptor
The Format of concern here is "WAVE", which requires two sub-chunks: "fmt " and "data"
The "fmt " sub-chunk
describes the format of the sound information in the data sub-chunk
The "data" sub-chunk
Indicates the size of the sound information and contains the raw sound data

Figure 1. The structure of .wav file format.

The screen above clearly shows that the actual data is stored after the 44-th byte and for illustrative reasons in our implementation will encrypt only the part of actual data which is stored from the 45-th byte till the end of the file, thus allowing the file to be playable but producing meaningless noise (encrypted voice).

5. Implementation

Software solution for the introduced system in this paper is implemented in C# language, with the interface as in the following figure:



End point IP address: 192.168.1.6
My secret key: 113 260 67

Generator point: 0 276
Received public key: 237 15 8/6/2013 2:16:41 AM
Sent public key: 260 67 8/6/2013 2:17:16 AM

$Y^2 = X^3 + 1 X + 1 \mod 277$

```
0    1    (0 , 1)
1    3    (0 , 276)
2    11   (1 , 130)
3    31   (1 , 147)
```

Original:
Ori_9.wav
Ori_8.wav
Ori_7.wav
Ori_6.wav
Ori_5.wav
Ori_4.wav
Ori_3.wav
Ori_2.wav
Ori_1.wav

Encrypted:
Enc_7.wav
Enc_6.wav
Enc_5.wav
Enc_4.wav
Enc_3.wav
Enc_2.wav
Enc_1.wav

```
Path:            C:\eccVtData\Mymessagess\Encrypted\
Filename:        Enc_7.wav
Id:              7
Encryption key:  113
Public key:      237_15
DateTime         8/6/2013 2:12:19 AM
Sent:            YES
Aux point:       260,67
```

New:
Enc_7.wav
Enc_6.wav
Enc_5.wav
Enc_4.wav
Enc_3.wav
Enc_2.wav
Enc_1.wav

Decrypted:
Dec_5.wav
Dec_4.wav
Dec_3.wav
Dec_2.wav
Dec_1.wav

Figure 2. System interface.

5.1. Encryption of voice message

The voice message is read and its bytes are stored in an array lsOriTB. The array is then split into two other arrays where the first array lsOriH contains the first 44 bytes which represent the header bytes, while the second array lsOriD contains the remaining bytes which represent the actual data of the voice.

lsOriH won't be encrypted in order to enable playing the file. The bytes from lsOriD will be mapped into corresponding points of the elliptic curve which are stored in the arrays lsMapX and lsMapY.

The encryption algorithm described in section 3 is applied upon the stored points in the arrays lsMapX and lsMapY. The encrypted points are stored in the arrays lsEX and lsEY. The bytes from lsEX and lsEY are mapped back into the corresponding points and stored into the array lsED.

lsOriH and lsED are merged into lsEncTB and the content is written in a .wav file. The file represents the encrypted message which plays a meaningless noise, and this way could be securely transmitted through the network.

5.2. Decryption of voice message

The encrypted voice message is read and its bytes are stored in the array lsEncTB. The array then is split into two other arrays where the first array lsOriH contains the first 44 bytes which represent the header bytes, while the second array lsEncD contains the following bytes which represent the actual data of the encrypted voice.

lsOriH won't be decrypted since it represents the original header. The bytes from lsEncD are mapped into corresponding points of the elliptic curve which are stored in the arrays lsMapX and lsMapY.

The decryption algorithm described in section 3 is applied upon the stored points in the arrays lsMapX and lsMapY. The decrypted points are stored in the arrays lsOX and lsOY.

The bytes from lsOX and lsOY are mapped back into the corresponding points and stored into the array lsOD.

lsOriH and lsOD are merged into lsOriTB and the content is written in a .wav file. The file represents the decrypted message which plays the original message, hence the goal of this paper is achieved.

5.3. Transmission of voice message

Transmission of messages is done using sockets [7]. The main path of the system operation is illustrated as follows:

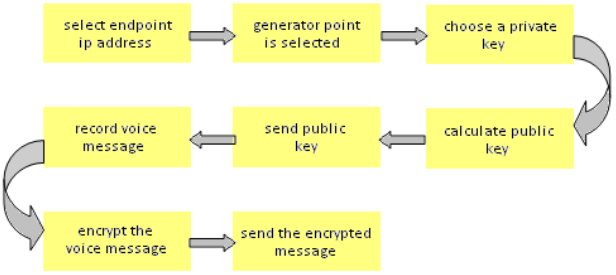

select endpoint ip address
generator point is selected
choose a private key
calculate public key
send public key
record voice message
encrypt the voice message
send the encrypted message

Figure 3. System main path operation.

To make the system more stable, during transmission of the voice message, encryption public data related to the message are also transmitted, which gives flexibility in changing and exchanging keys.

5.4. Data organization

The data that our system operates with are organized in files and folders. Also there is a folder for data backup. The organization of the data is better depicted by the following figure:

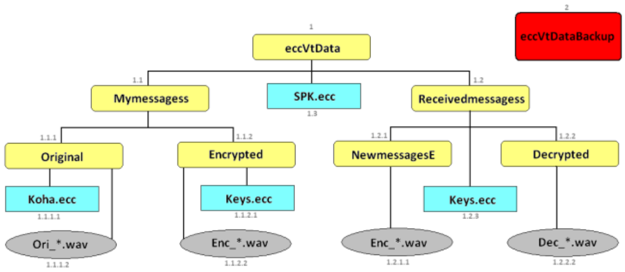


Figure 4. Data organization.

Files of type .ecc are files in which are stored voice message details such as: recording time, encryption and decryption keys, transmission time etc.

6. Conclusion and Future Work

In this paper we have proposed and implemented a cryptographic system based on elliptic curves, adapted to provide secure audio communication between communicating parties.

Using a similar approach, one can use elliptic curves for encryption of other types of data like image, video, text. The advantage of elliptic curves relies in the fact that using a smaller-length key results in a stronger encryption compared to RSA encryption.

Considering the prospect of elliptic curves in terms of cryptosystems, there remain to work on optimizing the provided solution and adapt it for an implementation which will enable secure real time mobile communication.

References

[1] Shoewu, O. and S.O. Olatinwo. 2013. "Securing Text Messages using Elliptic Curve Cryptography Orthogonal Frequency Division Multiplexing".

[2] Nils Gura, Arun Patel, Arvinderpal Wander, Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs, 2004.

[3] Fuwen Liu, A Tutorial on Elliptic Curve Cryptography (ECC),https://www-rnks.informatik.tu-cottbus.de/content/unrestricted/staff/lfw/A%20tutorial%20of%20elliptic%20curve%20cryptography.pdf [15.12.2013].

[4] Christof Paar and Jan Pelzi, Understanding Cryptography, http://wiki.crypto.rub.de/Buch/download/Understanding_Cryptography_Chptr_9---ECC.pdf [15.12.2013].

[5] Design of Secure Computer Systems CSI4138/CEG4394, Dr Jean-Yves Chouinard, 2002.

[6] WAVE PCM soundfile format. https://ccrma.stanford.edu/courses/422/projects/WaveFormat/ [15.12.2013].

[7] Socket Class, http://msdn.microsoft.com/en-us/library/system.net.sockets.socket(v=vs.110).aspx [15.12.2013]