INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Cryptography

"Euclid's and Extended Euclid's Algorithm"

Abstract

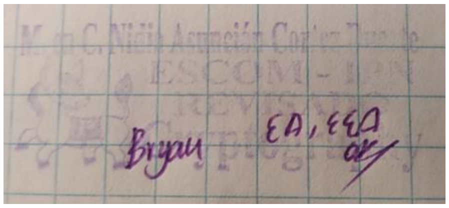
Euclid's Algorithm allows us to calculate de greatest common divisor between two numbers, and the extended version of this algorithm also allow us to express the gcd as a lineal combination of numbers. In this report I present a program that calculate gcd and give the numbers of the extended Euclidean algorithm.

By:

Bryan Dominguez de la Rosa

Professor:
MSc. NIDIA ASUNCIÓN CORTEZ DUARTE

September 2018



M. en C. Nidia Asunción Cortez Duarte
ESCOM - IPN
REVISADO
Cryptography

Bryan EA, EEA ok

Index

Contenido

Introduction:.. 1

Literature review:.. 1

Software (libraries, packages, tools): 1

Procedure:... 2

Results: 2

Discussion:.. 4

Conclusions:.. 4

References:.. 4

Code.. 4

Introduction:

The Euclid’s algorithm is a classic and fundamental mathematical knowledge that allow us to find de greatest common divisor between two numbers. The extended Euclid’s algorithm express the gcd as a lineal combination. This two methods are used in the affine cipher with an alphabet of length N, because if the gcd(a,N) is different of 1, the cihper is not going to work correctly, and also we can find the decrypt formula with the Euclid’s extended algorithm.

Literature review:

The cryptology is the science that deal with theoretical problems related with security in encrypted messages exchange from a sender to a receiver through a channel of communication (in informatic terms, this channel is usually a computer network). [1]

The greatest common divisor(g.c.d.) of the integers x and y is the largest integer d which divides both integers, denoted d = gcd(x, y).

The g.c.d. exists if at least one of the integers x and y is different of 0. Note that the g.c.d. is positive. (It’s often agreed, however, that gcd(0, 0) = 0) if gcd(x, y) = 1 then we say that x and y have no common divisors or that they are coprime. [2]

Using the Euclidean algorithm we can determine when a has an inverse modulo m by testing whether

$$\gcd(a, m) = 1$$

But we still do not know how to determine the inverse when it exists. To do this we use a variant of Euclid’s gcd algorithm, called the extended Euclidean algorithm.

The extended Euclidean algorithm takes as input *a* and *b* and outputs *rm, sm* and *tm* such that
rm = gcd(*a, b*) = *sma* + *tmb* [3]

Software (libraries, packages, tools):

I use Java programing language to develop this practice, so the tools that I used were:

- NetBeans IDE 8.2
- Java Development Kit (jdk) 8
- All libraries used were of the Java Standard

Procedure:

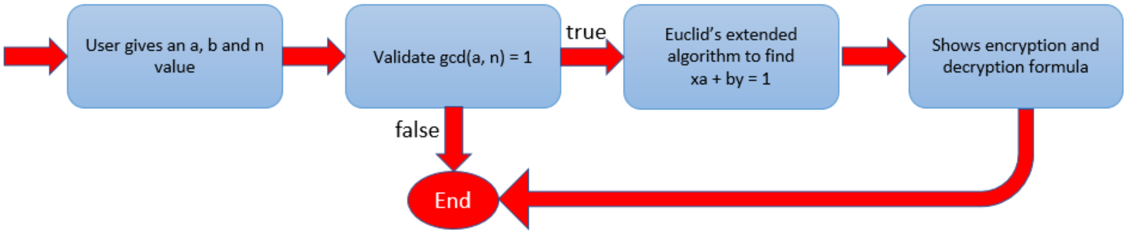

User gives an a, b and n value
Validate gcd(a, n) = 1
true
Euclid's extended algorithm to find xa + by = 1
Shows encryption and decryption formula
false
End

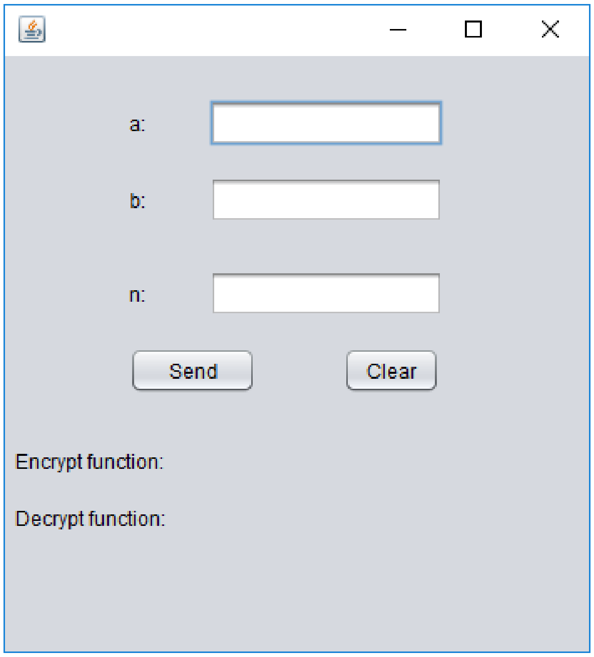
Fig. 1 Block diagram of the procedure of the program

The program displays an user interface that allows him to type the values of a, b and n. The program calculate gcd(a,n) and if it is different of one, the program shows a message saying that it's not possible to calculate the formulas if gcd(a,n) is different of one.

If the values pass the the first validation, then the program calculate the Euclid's extended algorithm to find x value and y value for $ax + by = 1$

Results:

The user interface is shown in Fig. 2. It is the main interface where the user type the values to use.

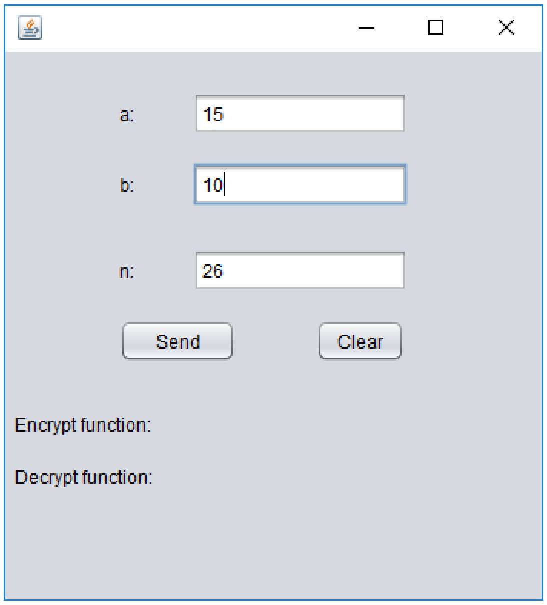


Encrypt function:

Decrypt function:

Fig. 2 The main interface of the program.

Once the user type the values required, the program is ready to start calculating, as can be seen is Fig 3.



a: 15

b: 10

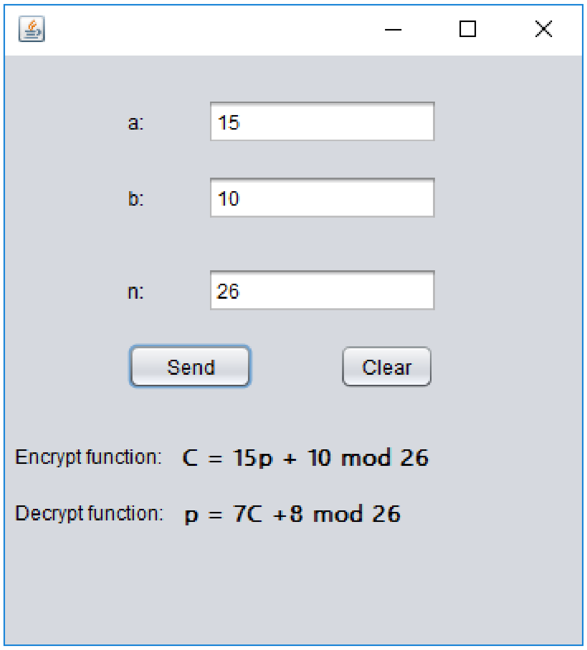
n: 26

Encrypt function:

Decrypt function:

Fig. 3 How the program looks when it’s ready

When the user gives all the values and press the Send button, the program starts to calculate, as in Fig 4.



a: 15

b: 10

n: 26

Encrypt function: C = 15p + 10 mod 26

Decrypt function: p = 7C +8 mod 26

Fig. 4 Results given by the program

In the Fig. 4 we can see the two results that the program give us. The first one is the affine cipher encryption formula with the values given by the user, and in second place we have the affine cipher decryption formula that correspond to the given values.

Discussion:

The validations and algorithms that this program present are really important to make the affine cipher work correctly, because if we don't have a and n coprime, we are not going to be able to achieve a good performance in the encryption algorithm. And then the Euclid's extended algorithm is an easy way to find which values we are going to use to decrypt the message ciphered by affine cipher.

Conclusions:

This practice is a really interesting one, first because Euclid's algorithm is a mathematical knowledge that every school teach, and the way to program it is very easy. And then if we talk about the extended algorithm, the level increases. One thing that could be done and I think it's very simple, is to make an "animation" that prints every single operation of the two algorithms, in this way the program could be use to teach people how the algorithm really works, and to explain it step by step.

References:

[1] S. Fernández, "La criptografía clásica", Sigma, no. 24, pp. 119-142, 2004.

[2] K. Ruohonen, Mathematical Cryptology. 2014.

[3] N. Smart, Cryptography. London: McGraw-Hill, 2003.

Code

EuclidesValidation.java

```java
package com.algorithm;

public class EuclidesValidation {

    private int a;
    private int b;
    private int n;
    private double inv_a;
    private int inv_b;

    public EuclidesValidation() {
    }

    public EuclidesValidation(int a, int b, int n) {
        this.a = a;
        this.b = b;
        this.n = n;
        this.inv_b = (n - b);
    }

    public int getInvBetha(){
        return this.inv_b = (int) ((n-b)*getInv_a()%getN());
    }

    public int gdc() {
        int aAux = getA();
        int nAux = getN();

        while (aAux != nAux) {
            if (aAux < nAux) {
                nAux = nAux - aAux;
            } else {
                aAux = aAux - nAux;
            }
        }
        return (aAux);
    }

    public void euclidesExtendido() {
        double aAux = getA();
        double nAux = getN();

        double x = 0, y = 0, d = 0;
        double x2 = 1, x1 = 0, y2 = 0, y1 = 1;
        double q = 0, r = 0;

        while (nAux > 0) {
            q = Math.floor(aAux / nAux);
            r = aAux - q * nAux;
            x = x2 - q * x1;
            y = y2 - q * y1;
            aAux = nAux;
            nAux = r;
            x2 = x1;
            x1 = x;
            y2 = y1;
            y1 = y;
        }
        inv_a = x2;
    }

    public int getInv_b() {
        return (this.n - b);
    }

    public int getA() {
        return a;
    }

    public void setA(int a) {
        this.a = a;
    }

    public int getB() {
        return b;
    }

    public void setB(int b) {
        this.b = b;
    }

    public double getInv_a() {
        return inv_a;
    }

    public void setInv_a(int inv_a) {
        this.inv_a = inv_a;
    }

    public int getN() {
        return n;
    }

    public void setN(int n) {
        this.n = n;
    }

}
```

MainFrame.java

```java
package com.gui;

import com.algorithm.EuclidesValidation;
import javax.swing.JOptionPane;

/**
 *
 * @author Bryan
 */
public class MainFrame extends javax.swing.JFrame {

    public MainFrame() {
        setLocationRelativeTo(null);
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        bTxt = new javax.swing.JTextField();
        nTxt = new javax.swing.JTextField();
        aTxt = new javax.swing.JTextField();
        sendBtn = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        encryptLabel = new javax.swing.JLabel();
        decryptLabel = new javax.swing.JLabel();
        clearBtn = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setText("a:");

        jLabel2.setText("b:");

        jLabel3.setText("n:");

        sendBtn.setText("Send");
        sendBtn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                sendBtnActionPerformed(evt);
            }
        });

        jLabel4.setText("Encrypt function:");

        jLabel5.setText("Decrypt function:");

        encryptLabel.setFont(new java.awt.Font("Malgun Gothic Semilight", 1, 14)); // NOI18N

        decryptLabel.setFont(new java.awt.Font("Malgun Gothic Semilight", 1, 14)); // NOI18N

        clearBtn.setText("Clear");
        clearBtn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                clearBtnActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap(74, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(sendBtn, javax.swing.GroupLayout.PREFERRED_SIZE, 76, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(52, 52, 52)
                        .addComponent(clearBtn))
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel3)
                            .addGap(38, 38, 38)
                            .addComponent(nTxt, javax.swing.GroupLayout.PREFERRED_SIZE, 140, javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(layout.createSequentialGroup()
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                                .addComponent(jLabel2)
                                .addComponent(jLabel1))
                            .addGap(38, 38, 38)
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addComponent(aTxt, javax.swing.GroupLayout.PREFERRED_SIZE, 140, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(bTxt, javax.swing.GroupLayout.PREFERRED_SIZE, 140, javax.swing.GroupLayout.PREFERRED_SIZE)))))
                .addGap(87, 87, 87))
```

```
.addGroup(layout.createSequentialGroup()
    .addContainerGap()
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel4)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                  .addComponent(encryptLabel, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
              .addGroup(layout.createSequentialGroup()
                  .addComponent(jLabel5)
                  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                  .addComponent(decryptLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 205, javax.swing.GroupLayout.PREFERRED_SIZE)))
          .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
  );
  layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
          .addGap(26, 26, 26)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(aTxt, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel1))
          .addGap(18, 18, 18)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(bTxt, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel2))
          .addGap(28, 28, 28)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(nTxt, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel3))
          .addGap(18, 18, 18)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(sendBtn)
              .addComponent(clearBtn))
          .addGap(32, 32, 32)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
              .addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
              .addComponent(encryptLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 16, javax.swing.GroupLayout.PREFERRED_SIZE))
          .addGap(18, 18, 18)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
              .addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
                    .addComponent(decryptLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 16, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(72, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void sendBtnActionPerformed(java.awt.event.ActionEvent evt) {
        if (!aTxt.getText().isEmpty() && !bTxt.getText().isEmpty() && !nTxt.getText().isEmpty()) {
            int a = Integer.parseInt(aTxt.getText());
            int b = Integer.parseInt(bTxt.getText());
            int n = Integer.parseInt(nTxt.getText());
            b = b % n;
            EuclidesValidation ev = new EuclidesValidation(a, b, n);
            int gcd;
            if (n == 0) {
                JOptionPane.showMessageDialog(this, "n cannot be 0",
                        "Info", JOptionPane.INFORMATION_MESSAGE, null);
            } else if (a > n) {
                JOptionPane.showMessageDialog(this, "a must be less or equal than n",
                        "Info", JOptionPane.INFORMATION_MESSAGE, null);
            } else if ((gcd = ev.gdc()) != 1) {
                JOptionPane.showMessageDialog(this, "Invalid a and n, gcd(a,n) is " + gcd
                        + ". To continue, gcd(a,n) must be 1", "Info", JOptionPane.INFORMATION_MESSAGE, null);
            } else {
                ev.euclidesExtendido();
                encryptLabel.setText("C = " + ev.getA() + "p + " + ev.getB() + " mod " + ev.getN());
                decryptLabel.setText("p = " + (int) ev.getInv_a() + "C +" + ev.getInvBetha() + " mod " + ev.getN());
            }
        } else {
            JOptionPane.showMessageDialog(this,
                    "Type a value for a, n and b",
                    "Error", JOptionPane.ERROR_MESSAGE, null);
        }


    }

    private void clearBtnActionPerformed(java.awt.event.ActionEvent evt) {
        aTxt.setText("");
        bTxt.setText("");
        nTxt.setText("");
        encryptLabel.setText("");
        decryptLabel.setText("");
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
```

```java
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new MainFrame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JTextField aTxt;
    private javax.swing.JTextField bTxt;
    private javax.swing.JButton clearBtn;
    private javax.swing.JLabel decryptLabel;
    private javax.swing.JLabel encryptLabel;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JTextField nTxt;
    private javax.swing.JButton sendBtn;
    // End of variables declaration
}
```