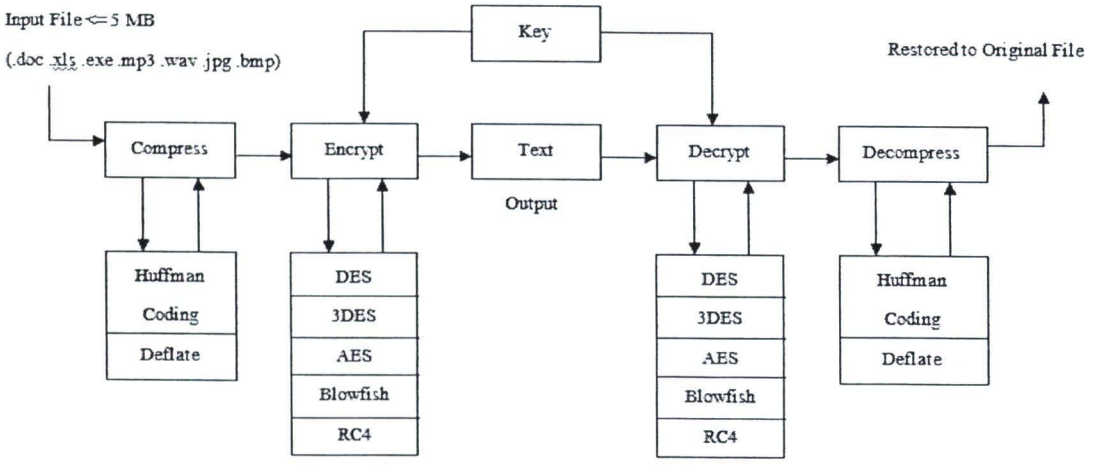
บทที่ 3
ขั้นตอนการออกแบบและพัฒนาโปรแกรม

ในส่วนของขั้นตอนการออกแบบและพัฒนาโปรแกรม ในการศึกษาครั้งนี้ได้ใช้โปรแกรมไมโครซอฟท์วิชวลเบสิค 2008 (Microsoft Visual Basic 2008) เป็นเครื่องมือในการพัฒนา โดยนำมาใช้พัฒนาเป็นโปรแกรมในการบีบอัดไฟล์พร้อมทั้งเข้ารหัสลับซึ่งสามารถให้ผู้ใช้กำหนดได้เลือกได้ว่า จะทำการบีบอัดข้อมูลและเข้ารหัสลับด้วยอัลกอริทึมแบบใด นอกจากนั้นยังนำมาใช้คำนวณเพื่อหาค่าตามกำหนดเกณฑ์วัดประสิทธิภาพในด้านของ เวลาที่ใช้ในการดำเนินการ ขนาดของผลลัพธ์ อัตราส่วนในการบีบอัด ประสิทธิภาพในการบีบอัด MIPS และ Brute force time อีกด้วย

3.1. กระบวนการทำงานของโปรแกรม

กระบวนการทำงานของโปรแกรมสามารถแสดงด้วยแผนภาพบล็อก (Block Diagram) ดังนี้



รูป 3.1 Block Diagram แสดงกระบวนการทำงานของโปรแกรม

(1) การนำเข้าข้อมูล สามารถเลือกใช้ไฟล์ประเภทต่างๆ ได้หลากหลายตามความต้องการโดยกำหนดให้มีขนาดไฟล์ไม่เกิน 5 เมกกะไบต์ เนื่องจากวัตถุประสงค์ของการศึกษาในครั้งประสงค์ให้โปรแกรมดังกล่าว เป็นโปรแกรมที่ใช้งานส่วนบุคคลที่เหมาะกับการทำงานพื้นฐานโดยทั่วไป ซึ่งแม้ว่าโดยหลักการแล้วจะสามารถกำหนดการจำกัดขนาดของไฟล์ได้

ได้มากกว่านี้ แต่ด้วยไฟล์ข้อมูลส่วนใหญ่ ไม่ว่าจะเป็นไฟล์ .mp3 ไฟล์เอกสาร หรือไฟล์รูปภาพ โดยทั่วไปแล้วก็มักจะมีขนาดไม่เกิน 5 เมกกะไบต์ทั้งสิ้น อีกทั้งในการทดลองเพื่อเก็บค่าสถิตินั้น หากใช้ไฟล์ตัวอย่างในการนำเข้าข้อมูลที่มีขนาดใหญ่ จะทำให้การประมวลผลข้อมูลดำเนินการไปได้ช้ามาก เพราะความเร็วในการดำเนินการ ขึ้นอยู่กับขนาดของไฟล์ต้นฉบับโดยตรง ดังนั้น การจำกัดขนาดของไฟล์นำเข้าไว้ที่ไม่เกิน 5 เมกกะไบต์ จึงเพียงพอต่อการใช้งานโดยทั่วไป และสะดวกต่อการทดลองในการศึกษาเพื่อเก็บค่าสถิติ

(2) เมื่อทำการเลือกไฟล์ที่จะนำเข้าแล้ว ระบบจะทำการบีบอัดข้อมูลโดยให้ผู้ใช้เลือกอัลกอริทึมในการบีบอัดได้ 2 แบบคือ Huffman coding หรือ Deflate

(3) หลังจากที่บีบอัดข้อมูลแล้ว โปรแกรมจะนำไฟล์ดังกล่าวมาทำการเข้ารหัสลับ ซึ่งผู้ใช้สามารถเลือกอัลกอริทึมในการเข้ารหัสลับได้ 5 แบบคือ DES, 3DES, AES, Blowfish และ RC4 ซึ่งการเข้ารหัสลับนั้น ผู้ใช้ต้องระบุ key ที่จะใช้ในการเข้ารหัสด้วย

(4) เมื่อเข้ารหัสลับเสร็จแล้ว โปรแกรมนำไฟล์ที่ได้มาเข้ารหัสแบบ Base64 เพื่อให้ได้ผลลัพธ์ออกมาอยู่ในรูปแบบของตัวอักษร ที่จะสามารถให้ผู้ใช้คัดลอกข้อความเหล่านั้นไปใช้งานได้โดยไม่ต้องใช้ไฟล์

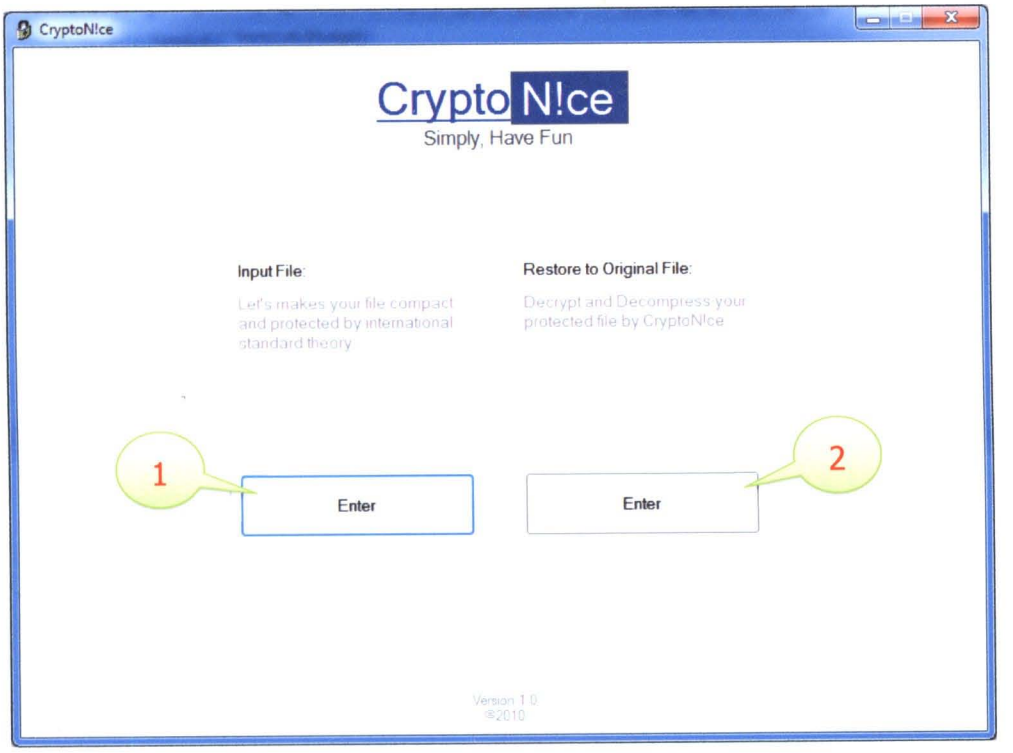
(5) ในส่วนของกระบวนการถอดรหัส เริ่มจากนำข้อความตัวอักษรที่ได้จากข้อ (4) มาวางในโปรแกรม เพื่อให้โปรแกรมถอดรหัสด้วย Base64 ให้ข้อความเหล่านั้นกลับมาอยู่ในรูปของไฟล์

(6) โปรแกรมจะทำการถอดรหัสลับไฟล์โดยผู้ใช้จะต้องทำการระบุอัลกอริทึมที่ใช้ให้ถูกต้องตามที่ไฟล์ต้นฉบับได้เคยเข้ารหัสมา ซึ่งผู้ใช้จะต้องระบุ key ให้ตรงกันกับ key ที่ใช้ในการเข้ารหัสลับด้วย

(7) เมื่อถอดรหัสลับได้แล้ว โปรแกรมก็จะทำการคลายการบีบอัดไฟล์ดังกล่าวให้กลับมาอยู่ในรูปของไฟล์ต้นฉบับ ซึ่งผู้ใช้ก็ต้องระบุอัลกอริทึมในการบีบอัดให้ตรงกลับการบีบอัดครั้งแรกด้วยเช่นกัน

3.2. การออกแบบหน้าจอส่วนติดต่อผู้ใช้

การออกแบบหน้าจอส่วนติดต่อผู้ใช้ (Graphic User Interface) โปรแกรมจะประกอบด้วย 2 ส่วนหลักๆ คือ ส่วนของการนำเข้าข้อมูล และส่วนของการนำกลับมาเป็นไฟล์ต้นฉบับ โดยหน้าจอหลักจะแบ่งทางเลือกทั้งสองระบบให้ผู้ใช้ได้เลือกตั้งแต่แรก ดังรูป 3.2

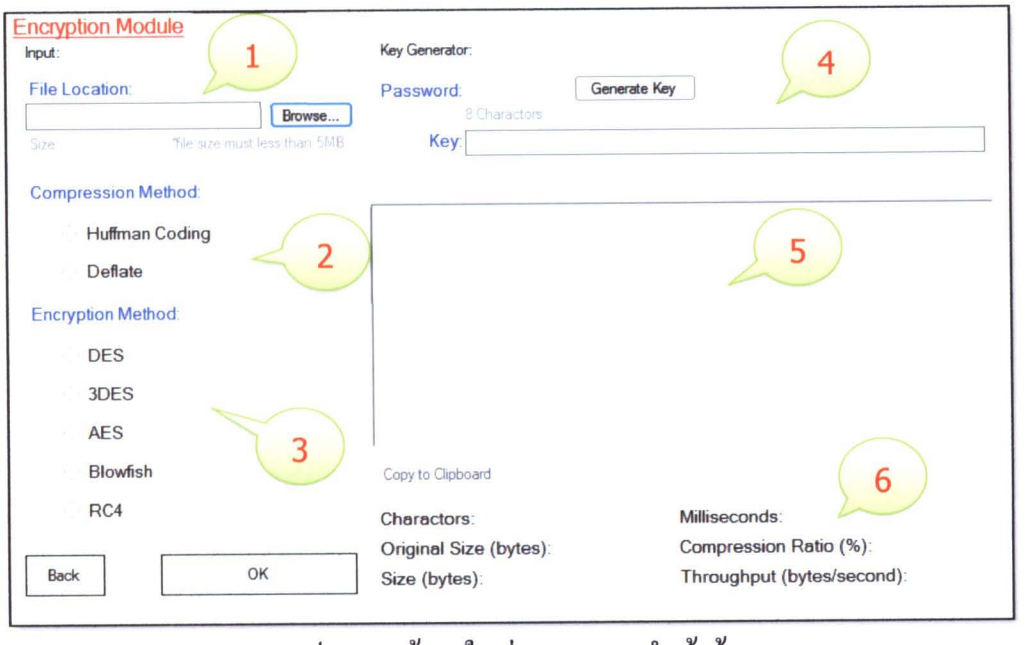


รูป 3.2 หน้าจอหลักของโปรแกรม

(1) เข้าสู่ระบบนำเข้าข้อมูล

(2) เข้าสู่ระบบนำกลับคืนมาเป็นไฟล์ต้นฉบับ

ในส่วนของระบบนำเข้าข้อมูล จะต้องให้ผู้ใช้สามารถเลือกไฟล์ที่ต้องการนำเข้า และเลือกใช้อัลกอริทึมในแต่ละแบบได้



รูป 3.3 หน้าจอในส่วนของการนำเข้าข้อมูล

(1) ให้ผู้ใช้เลือกไฟล์นำเข้า

(2) อัลกอริทึมในการบีบอัด

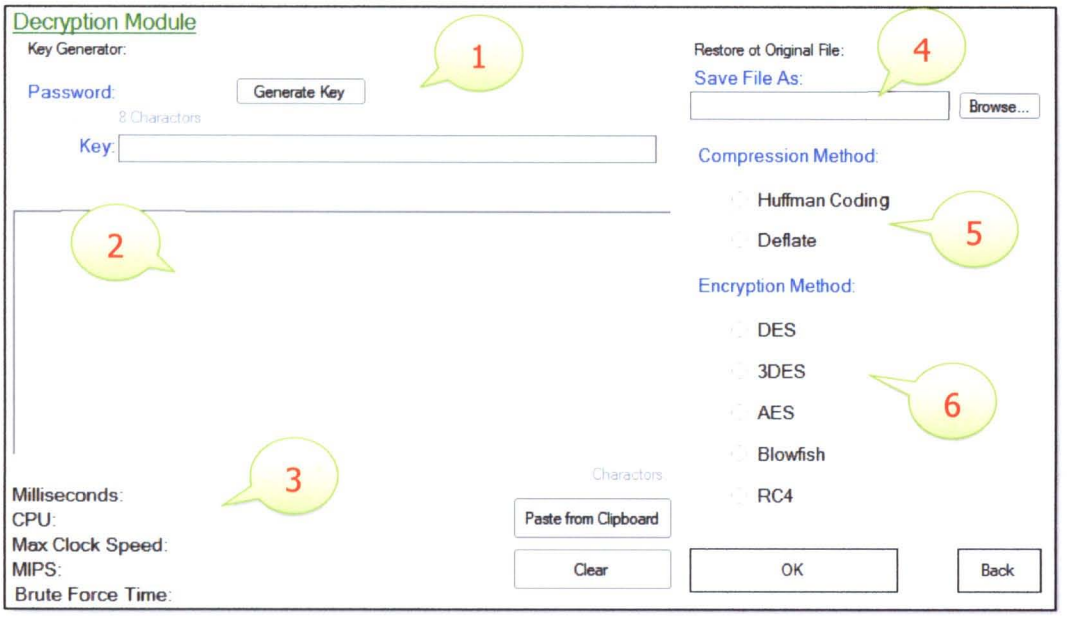
(3) อัลกอริทึมในการเข้ารหัส

(4) ส่วนที่ให้ผู้ใช้ระบุ key

(5) ส่วนแสดงผลลัพธ์ในรูปแบบของตัวอักษรให้ผู้ใช้สามารถคัดลอกได้

(6) ข้อมูลสถิติต่างๆ

ในส่วนของระบบนำกลับคืนมาเป็นไฟล์ต้นฉบับ ก็คล้ายกับการนำเข้าข้อมูล คือจะต้องมีส่วนให้ผู้ใช้วางข้อความตัวอักษร เลือกอัลกอริทึมในการคลายบีบอัด อัลกอริทึมในการถอดรหัสลับ พร้อมทั้งให้ระบุ key และระบุที่บันทึกไฟล์ ดังรูป 3.4



รูป 3.4 หน้าจอในส่วนของระบบนำกลับคืนมาเป็นไฟล์ต้นฉบับ

(1) ส่วนที่ให้ผู้ใช้ระบุ key

(2) กล่องข้อความที่ให้ผู้ใช้วางตัวอักษรที่ผ่านการเข้ารหัสมาแล้ว

(3) ค่าสถิติต่างๆ

(4) ให้ผู้ใช้สามารถเลือกตำแหน่งและชื่อไฟล์ที่ต้องการบันทึก

(5) อัลกอริทึมที่ใช้ในการคลายการบีบอัด

(6) อัลกอริทึมที่ใช้ในการถอดรหัสลับ

3.3. การเขียนโปรแกรม

ในการเขียนโปรแกรมนี้ ได้ใช้ภาษา Visual Basic เป็นหลัก โดยแต่ละระบบจะถูกแยกออกเป็น Sub, Function หรือ Class ตามแต่ความสะดวกในการเรียกใช้งานเป็นหลัก ซึ่งแต่ละระบบจะขอยกมานำเสนอเฉพาะส่วนสำคัญ ดังนี้

3.3.1. การบีบอัดข้อมูล

การเขียนชุดคำสั่งที่ใช้บีบอัดข้อมูลของทั้งสองแบบ คือ Deflate และ Huffman มีรูปแบบการเขียนชุดคำสั่งที่คล้ายคลึงกันคือจะใช้ฟังก์ชั่นสำเร็จรูปที่โปรแกรม Visual Basic 2008 มีไว้ให้ เป็นตัวเข้ารหัส เช่น การบีบอัดข้อมูล และการคลายการบีบอัดด้วยวิธี Deflate มีชุดคำสั่ง ดังนี้

Deflate Compression:

```
Public Sub CompressDeflate(ByVal sourceFile As String, ByVal outFile As String)
    Dim msg As String
    Dim destFile As FileStream = File.Create(outFile)

    Dim infile As FileStream
    Try
        ' Open the file as a FileStream object.
        infile = New FileStream(sourceFile, FileMode.Open, FileAccess.Read, FileShare.Read)
        Dim buffer(infile.Length - 1) As Byte
        ' Read the file to ensure it is readable.
        Dim count As Integer = infile.Read(buffer, 0, buffer.Length)
        If count <> buffer.Length Then
            infile.Close()
            msg = "Test Failed: Unable to read data from file"
            MsgBox(msg)
            Return
        End If
```

```
        Dim compressedzipStream As New DeflateStream(destFile,
CompressionMode.Compress, True)
        compressedzipStream.Write(buffer, 0, buffer.Length)

        Dim myByte As Integer = infile.ReadByte()
        While myByte <> -1
            compressedzipStream.WriteByte(CType(myByte, Byte))
            myByte = infile.ReadByte()
        End While

        ' Close the stream.
        compressedzipStream.Close()
        infile.Close()
        msg = "Original size: " & buffer.Length & ", Compressed size: " & destFile.Length
        destFile.Position = 0
        destFile.Close()

    Catch er As Exception
        msg = "Error: The file being read contains invalid data."
        MsgBox(msg)
    End Try
End Sub
```

Deflate Decompression:

```
Public Sub DecompressDef(ByVal inFile As String, ByVal outFile As String)
    Dim sourceFile As FileStream = File.OpenRead(inFile)
    Dim destFile As FileStream = File.Create(outFile)
    Dim zipStream As New DeflateStream(sourceFile, CompressionMode.Decompress)
    Dim myByte As Integer = zipStream.ReadByte()

    While myByte <> -1
      destFile.WriteByte(CType(myByte, Byte))
      myByte = zipStream.ReadByte()
    End While
    zipStream.Close()
    sourceFile.Close()
    destFile.Close()
End Sub
```

3.3.2. การเข้ารหัสข้อมูล

การเขียนชุดคำสั่งให้กับการเข้ารหัสข้อมูลในแบบต่างๆ ก็สามารถใช้ฟังก์ชั่นสำเร็จรูปที่ Visual Basic 2008 มีไว้ให้เช่นกัน เช่น ชุดคำสั่งการเข้ารหัสลับและการถอดรหัสลับของอัลกอริทึม DES ดังนี้

DES Encryption:

```
Private Shared Sub EncryptData(ByVal inName As String, ByVal outName As String, ByVal
desKey As String)
    'Create the file streams to handle the input and output files.
    Dim fin As New FileStream(inName, FileMode.Open, FileAccess.Read)
    Dim fout As New FileStream(outName, FileMode.OpenOrCreate, _
      FileAccess.Write)
    fout.SetLength(0)
```

```
'Create variables to help with read and write.
Dim bin(4096) As Byte 'This is intermediate storage for the encryption.
Dim rdlen As Long = 0 'This is the total number of bytes written.
Dim totlen As Long = fin.Length 'Total length of the input file.
Dim len As Integer 'This is the number of bytes to be written at a time.
Dim des As New DESCryptoServiceProvider()

des.Key = ASCIIEncoding.ASCII.GetBytes(desKey)
des.IV = ASCIIEncoding.ASCII.GetBytes(desKey)

Dim encStream As New CryptoStream(fout, _
   des.CreateEncryptor(des.Key, des.IV), CryptoStreamMode.Write)

Console.WriteLine("Encrypting...")

'Read from the input file, then encrypt and write to the output file.
While rdlen < totlen
   len = fin.Read(bin, 0, 4096)
   encStream.Write(bin, 0, len)
   rdlen = Convert.ToInt32(rdlen + len / des.BlockSize * des.BlockSize)
   Console.WriteLine("Processed {0} bytes, {1} bytes total", len, _
      rdlen)
End While

encStream.Close()

MsgBox("Success")
End Sub
```

DES Decryption:

```
Private Shared Sub DecryptData(ByVal inName As String, ByVal outName As String, ByVal
desKey As String)
    On Error GoTo errBadData
    'Create the file streams to handle the input and output files.
    Dim fin As New FileStream(inName, FileMode.Open, FileAccess.Read)
    Dim fout As New FileStream(outName, FileMode.OpenOrCreate, _
      FileAccess.Write)
    fout.SetLength(0)

    'Create variables to help with read and write.
    Dim bin(4096) As Byte 'This is intermediate storage for the encryption.
    Dim rdlen As Long = 0 'This is the total number of bytes written.
    Dim totlen As Long = fin.Length 'Total length of the input file.
    Dim len As Integer 'This is the number of bytes to be written at a time.
    Dim des As New DESCryptoServiceProvider()

    des.Key = ASCIIEncoding.ASCII.GetBytes(desKey)
    des.IV = ASCIIEncoding.ASCII.GetBytes(desKey)

    Dim encStream As New CryptoStream(fout, _
      des.CreateDecryptor(des.Key, des.IV), CryptoStreamMode.Write)

    Console.WriteLine("Decrypting...")

    'Read from the input file, then encrypt and write to the output file.
    While rdlen < totlen
      len = fin.Read(bin, 0, 4096)
      encStream.Write(bin, 0, len)
      rdlen = Convert.ToInt32(rdlen + len / des.BlockSize * des.BlockSize)
```

```
        Console.WriteLine("Processed {0} bytes, {1} bytes total", len, rdlen)
    End While
    encStream.Close()
    MsgBox("Success")
errBadData:
    If Err.Number = 5 Then
        MsgBox("Your key is not correct")
    End If
End Sub
```

3.3.3. การเข้ารหัสด้วย Base64

การใช้ Base64 ในโปรแกรมนี้ จะเป็นการทำให้ผลลัพธ์ออกมาในรูปของข้อความ และนำเอาข้อความนั้นไปใช้งาน ซึ่งใน Visual Basic 2008 ก็มีฟังก์ชั่นที่รองรับการทำงานนี้ด้วยเช่นกัน ซึ่งได้เขียนชุดคำสั่งอยู่ในรูปของ class และ function ดังนี้

Base64 Encoding:

```
Imports System.IO
Public Class clsBase64
    Public Function EncodeToBase64(ByRef aSource As String) As String
        Dim aReadB As Byte()
        aReadB = File.ReadAllBytes(aSource)
        Return Convert.ToBase64String(aReadB)
    End Function
End Class
```

Base64 Decoding:

```
    Function Decode64(ByVal bData As String) As Byte()
        Return Convert.FromBase64String(bData)
    End Function
```

3.3.4. เวลาที่ใช้ในการดำเนินการ

เวลาที่ใช้ในการดำเนินการ ในโปรแกรมนี้ได้กำหนดให้ใช้หน่วยเป็น Millisecond ซึ่งการจะเก็บเวลาการทำงานของ process นั้น ต้องอาศัย StopWatch ซึ่งเป็น class ที่มีอยู่ใน visual basic 2008 เป็นตัวจับเวลา ดังนี้

```
Dim sWatch As Stopwatch = New Stopwatch
sWatch.Reset()
sWatch.Start()
    ......
    ......
    ......
    ......
sWatch.Stop()
labMillisec.Text = "Milliseconds: " & Format(sWatch.ElapsedMilliseconds, "#,#")
```

3.3.5. ขนาดของผลลัพธ์

คือขนาดของไฟล์ที่ผ่านการเข้ารหัสและบีบอัดเสร็จเรียบร้อยแล้ว โดยสามารถวัดได้จาก function ที่เรียกค่าขนาดของไฟล์มาแสดงผล ดังนี้

```
Private Function GetFileSize(ByVal MyFilePath As String) As Long
    Dim MyFile As New FileInfo(MyFilePath)
    Dim FileSize As Long = Math.Ceiling(MyFile.Length)
    Return FileSize
End Function
```

นำมาแสดงผลด้วยคำสั่งที่ส่งค่าไปคำนวณที่ฟังก์ชั่น GetFileSize

```
labSize.Text = "Size (bytes): " & Format(GetFileSize("D:\inCompEncrypt"), "#,#")
```

3.3.6. อัตราส่วนในการบีบอัด

อัตราส่วนในการบีบอัด (Compression Ratio) จะแสดงผลออกมาเป็นเปอร์เซ็นต์ โดยคิดเป็นร้อยละว่าไฟล์ที่นำเข้าเสร็จแล้ว สามารถทำการบีบอัดได้เป็นร้อยละเท่าไหร่จากไฟล์ต้นฉบับ

```
Dim originalF As New FileInfo(labBrowse.Text)
labOriginal.Text = "Original Size (bytes): " & Format(originalF.Length, "#,#")
labRatio.Text = "Compression Ratio (%): " & Format(((originalF.Length -
GetFileSize("D:\inCompEncrypt")) / originalF.Length) * 100, "#.##")
```

3.3.7. ประสิทธิภาพในการบีบอัด

ประสิทธิภาพในการบีบอัด (Throughput) มีหน่วยวัดเป็น (bytes/second) ใช้บอกว่าเวลาใน 1 วินาที ระบบสามารถดำเนินการไปได้กี่ไบท์ ซึ่งในการคำนวณได้ใช้ค่าเวลาที่ได้จาก StopWatch มาร่วมด้วย

```
labThroughput.Text = "Throughput (bytes/second): " & Format(((originalF.Length -
GetFileSize("D:\inCompEncrypt")) / sWatch.ElapsedMilliseconds) * 1000, "#,#")
```

3.3.8. MIPS

MIPS เป็นการคำนวณเพื่อหาค่าประสิทธิภาพของ CPU เครื่องที่กำลังทำงานเพื่อที่จะได้นำไปวัดประสิทธิภาพในการป้องกันการถอดรหัสอีกครั้งหนึ่ง

ในการหาค่า MIPS จะทำในส่วนของการนำกลับไปเป็นไฟล์ต้นฉบับ โดยต้องเรียกฟังก์ชั่นที่อยู่ใน class ของ Win32_Processor เข้ามาช่วยในการอ่านค่าต่างๆ ของ CPU

```
Dim moReturn As Management.ManagementObjectCollection
Dim moSearch As Management.ManagementObjectSearcher
Dim mo As Management.ManagementObject
moSearch = New Management.ManagementObjectSearcher("Select * from
Win32_Processor")
moReturn = moSearch.Get
Dim clkSpeed As Integer, clkCPI As Decimal, cMIPS As Long, cBruteForce As Double
Dim clkName As String
```

```
For Each mo In moReturn
    clkSpeed = CInt(mo("maxclockspeed"))
    clkCPI = Format(((mo("loadpercentage") * 5) + (mo("store") * 4) + (mo("arithmetic") *
1)) / (mo("loadpercentage") + mo("store") + mo("arithmetic")), "#.####")
    cMIPS = clkSpeed / clkCPI
Next
labMIPS.Text = "MIPS: " & cMIPS
```

3.3.9. Brute force time

ใช้เพื่อวัดระยะเวลาในการถอดรหัสแบบทุกความเป็นไปได้ในทุกตำแหน่ง ซึ่งในการศึกษาครั้งนี้ได้ใช้กุญแจแบบสมมาตรในการเข้ารหัสลับ ดังนั้นระยะเวลาในการถอดรหัสแบบ brute force จึงขึ้นอยู่กับจำนวน key ที่ใช้ในการเข้ารหัส และประสิทธิภาพของ CPU ในการประมวลผล ที่เราได้ใช้ค่า MIPS เป็นตัวแทนมาใช้ในการคำนวณ

โดยเริ่มแรกจะกำหนดให้ key เป็น 0 ต่อมาเมื่อมีการเข้ารหัสแล้ว จะทำการเก็บค่า key ไว้ในกระบวนการ ว่าผู้ใช้เลือกอัลกอริทึมใดในการเข้ารหัส เช่น หากผู้ใช้เลือก DES ก็จะส่งค่า key ให้เป็น 56 เป็นต้น

```
Dim nKey As Long
nKey = 0  'initial key value
    ........
    ........
    ........
Dim cBruteForce As Double
cBruteForce = (2 ^ nKey) / (cMIPS * 1000000 * 3600 * 24 * 365)
```