

Forum:[Hack Forums](#)

Thread: [Massive amount of Encrypt/Decrypt functions \[All Working\]](#)

????? - June 4, 2012, 11:38 pm

I know how everyone is this .net section is SuPaH l33T crYpTeR c0D3Rz so I though I would share this huge list I had laying around.

I did not make these, of course, just sharing. Credits go to the makers.

AES:

```
Code Public Function AESEncrypt(ByVal input As String, ByVal pass As String) As String
    Dim AES As New System.Security.Cryptography.RijndaelManaged
    Dim Hash_AES As New System.Security.Cryptography.MD5CryptoServiceProvider
    Dim encrypted As String = ""
    Try
        Dim hash(31) As Byte
        Dim temp As Byte() =
Hash_AES.ComputeHash(System.Text.ASCIIEncoding.ASCII.GetBytes(pass))
        Array.Copy(temp, 0, hash, 0, 16)
        Array.Copy(temp, 0, hash, 15, 16)
        AES.Key = hash
        AES.Mode = Security.Cryptography.CipherMode.ECB
        Dim DESEncrypter As System.Security.Cryptography.ICryptoTransform =
AES.CreateEncryptor
        Dim Buffer As Byte() = System.Text.ASCIIEncoding.ASCII.GetBytes(input)
        encrypted = Convert.ToBase64String(DESEncrypter.TransformFinalBlock(Buffer, 0,
Buffer.Length))
        Return encrypted
    Catch ex As Exception
    End Try
End Function

Public Function AESDecrypt(ByVal input As String, ByVal pass As String) As String
    Dim AES As New System.Security.Cryptography.RijndaelManaged
    Dim Hash_AES As New System.Security.Cryptography.MD5CryptoServiceProvider
    Dim decrypted As String = ""
    Try
        Dim hash(31) As Byte
        Dim temp As Byte() =
Hash_AES.ComputeHash(System.Text.ASCIIEncoding.ASCII.GetBytes(pass))
        Array.Copy(temp, 0, hash, 0, 16)
        Array.Copy(temp, 0, hash, 15, 16)
        AES.Key = hash
        AES.Mode = Security.Cryptography.CipherMode.ECB
        Dim DESDecrypter As System.Security.Cryptography.ICryptoTransform =
AES.CreateDecryptor
        Dim Buffer As Byte() = Convert.FromBase64String(input)
        decrypted =
System.Text.ASCIIEncoding.ASCII.GetString(DESDecrypter.TransformFinalBlock(Buffer, 0,
Buffer.Length))
```

```

        Return decrypted
    Catch ex As Exception
    End Try
End Function
ATOM-128:
Code #Region "Atom 128"
    Public Function Atom128Encrypt(ByVal input As String) As String
        input = Uri.EscapeDataString(input)
        Dim key As String =
"/128GhloPQROSTeUbADfgHijKLM+n0pFWXY456xyzB7=39VaqrstJklmNuZvwcdEC"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs As Integer() = {0, 0, 0}
            For b As Integer = 0 To 2
                If i < input.Length Then chrs(b) = Asc(input(i))
                i += 1
            Next
            enc(0) = chrs(0) >> 2
            enc(1) = ((chrs(0) And 3) << 4) Or (chrs(1) >> 4)
            enc(2) = ((chrs(1) And 15) << 2) Or (chrs(2) >> 6)
            enc(3) = chrs(2) And 63
            If chrs(1) = 0 Then
                enc(2) = 64
                enc(3) = 64
            End If
            If chrs(2) = 0 Then
                enc(3) = 64
            End If
            For Each x As Integer In enc
                out.Append(key(x))
            Next
            Loop While i < input.Length
            Return out.ToString
        End Function
    Public Function Atom128Decrypt(ByVal input As String) As String
        Dim key As String =
"/128GhloPQROSTeUbADfgHijKLM+n0pFWXY456xyzB7=39VaqrstJklmNuZvwcdEC"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs() As Integer = {0, 0, 0}
            For b As Integer = 0 To 3
                enc(b) = key.IndexOf(input(i))
                i = i + 1
            Next
            chrs(0) = (enc(0) << 2) Or (enc(1) >> 4)
            chrs(1) = (enc(1) And 15) << 4 Or (enc(2) >> 2)
            chrs(2) = (enc(2) And 3) << 6 Or enc(3)
            out.Append(Chr(chrs(0)))
            If enc(2) <> 64 Then out.Append(Chr(chrs(1)))

```

```

        If enc(3) <> 64 Then out.Append(Chr(chrs(2)))
    Loop While i < input.Length
    Return out.ToString
End Function
#End Region
HAZZ 15:
Code #Region "HAZZ 15"
    Public Function HAZZ15_Encode(ByVal input As String) As String
        input = Uri.EscapeDataString(input)
        Dim key As String =
"HNO4klm6ij9n+J2hyf0gzA8uvwDEq3X1Q7ZKeFrWcVTts/MRGYbdxSo=ILaUpPBC5"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs As Integer() = {0, 0, 0}
            For b As Integer = 0 To 2
                If i < input.Length Then chrs(b) = Asc(input(i))
                i += 1
            Next
            enc(0) = chrs(0) >> 2
            enc(1) = ((chrs(0) And 3) << 4 Or (chrs(1) >> 4)
            enc(2) = ((chrs(1) And 15) << 2 Or (chrs(2) >> 6)
            enc(3) = chrs(2) And 63
            If chrs(1) = 0 Then
                enc(2) = 64
                enc(3) = 64
            End If
            If chrs(2) = 0 Then
                enc(3) = 64
            End If
            For Each x As Integer In enc
                out.Append(key(x))
            Next
        Loop While i < input.Length
        Return out.ToString
    End Function
    Public Function HAZZ15_Decode(ByVal input As String) As String
        Dim key As String =
"HNO4klm6ij9n+J2hyf0gzA8uvwDEq3X1Q7ZKeFrWcVTts/MRGYbdxSo=ILaUpPBC5"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs() As Integer = {0, 0, 0}
            For b As Integer = 0 To 3
                enc(b) = key.IndexOf(input(i))
                i = i + 1
            Next
            chrs(0) = (enc(0) << 2 Or (enc(1) >> 4)
            chrs(1) = (enc(1) And 15) << 4 Or (enc(2) >> 2)
            chrs(2) = (enc(2) And 3) << 6 Or enc(3)
            out.Append(Chr(chrs(0)))

```

```

        If enc(2) <> 64 Then out.Append(Chr(chrs(1)))
        If enc(3) <> 64 Then out.Append(Chr(chrs(2)))
    Loop While i < input.Length
    Return out.ToString
End Function
#End Region
GILA 7:
Code #Region "GILA 7"
    Public Function GILA7_Encode(ByVal input As String) As String
        input = Uri.EscapeDataString(input)
        Dim key As String =
"7ZSTJK+W=cVtBCasyf0gzA8uvwDEq3XH/1RMNOILPQU4klm65YbdeFrX2hij9nopG"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs As Integer() = {0, 0, 0}
            For b As Integer = 0 To 2
                If i < input.Length Then chrs(b) = Asc(input(i))
                i += 1
            Next
            enc(0) = chrs(0) >> 2
            enc(1) = ((chrs(0) And 3) << 4) Or (chrs(1) >> 4)
            enc(2) = ((chrs(1) And 15) << 2) Or (chrs(2) >> 6)
            enc(3) = chrs(2) And 63
            If chrs(1) = 0 Then
                enc(2) = 64
                enc(3) = 64
            End If
            If chrs(2) = 0 Then
                enc(3) = 64
            End If
            For Each x As Integer In enc
                out.Append(key(x))
            Next
        Loop While i < input.Length
        Return out.ToString
    End Function
    Public Function GILA7_Decode(ByVal input As String) As String
        Dim key As String =
"7ZSTJK+W=cVtBCasyf0gzA8uvwDEq3XH/1RMNOILPQU4klm65YbdeFrX2hij9nopG"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs() As Integer = {0, 0, 0}
            For b As Integer = 0 To 3
                enc(b) = key.IndexOf(input(i))
                i = i + 1
            Next
            chrs(0) = (enc(0) << 2) Or (enc(1) >> 4)
            chrs(1) = (enc(1) And 15) << 4 Or (enc(2) >> 2)
            chrs(2) = (enc(2) And 3) << 6 Or enc(3)

```

```

        out.Append(Chr(chrs(0)))
        If enc(2) <> 64 Then out.Append(Chr(chrs(1)))
        If enc(3) <> 64 Then out.Append(Chr(chrs(2)))
    Loop While i < input.Length
    Return out.ToString
End Function
#End Region
ESAB 46:
Code #Region "ESAB 46"
    Public Function ESAB46_Encode(ByVal input As String) As String
        input = Uri.EscapeDataString(input)
        Dim key As String =
"ABCDqrs456tuvNOPwxyz012KLM3789=+QRSTUVWXYZabcdefghijklmnopEFGHIJ/"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs As Integer() = {0, 0, 0}
            For b As Integer = 0 To 2
                If i < input.Length Then chrs(b) = Asc(input(i))
                i += 1
            Next
            enc(0) = chrs(0) >> 2
            enc(1) = ((chrs(0) And 3) << 4) Or (chrs(1) >> 4)
            enc(2) = ((chrs(1) And 15) << 2) Or (chrs(2) >> 6)
            enc(3) = chrs(2) And 63
            If chrs(1) = 0 Then
                enc(2) = 64
                enc(3) = 64
            End If
            If chrs(2) = 0 Then
                enc(3) = 64
            End If
            For Each x As Integer In enc
                out.Append(key(x))
            Next
        Loop While i < input.Length
        Return out.ToString
    End Function
    Public Function ESAB46_Decode(ByVal input As String) As String
        Dim key As String =
"ABCDqrs456tuvNOPwxyz012KLM3789=+QRSTUVWXYZabcdefghijklmnopEFGHIJ/"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs() As Integer = {0, 0, 0}
            For b As Integer = 0 To 3
                enc(b) = key.IndexOf(input(i))
                i = i + 1
            Next
            chrs(0) = (enc(0) << 2) Or (enc(1) >> 4)
            chrs(1) = (enc(1) And 15) << 4 Or (enc(2) >> 2)

```

```

        chrs(2) = (enc(2) And 3) << 6 Or enc(3)
        out.Append(Chr(chrs(0)))
        If enc(2) <> 64 Then out.Append(Chr(chrs(1)))
        If enc(3) <> 64 Then out.Append(Chr(chrs(2)))
    Loop While i < input.Length
    Return out.ToString
End Function
#End Region
MEGAN 35:
Code #Region "Megan 35"
    Public Function MEGAN35_Encode(ByVal input As String) As String
        input = Uri.EscapeDataString(input)
        Dim key As String =
"3GHIJKLMNOPQRSTUbcdefghijklmnopWXYZ/12+406789VqrstuvwxyzABCDEF5"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs As Integer() = {0, 0, 0}
            For b As Integer = 0 To 2
                If i < input.Length Then chrs(b) = Asc(input(i))
                i += 1
            Next
            enc(0) = chrs(0) >> 2
            enc(1) = ((chrs(0) And 3) << 4) Or (chrs(1) >> 4)
            enc(2) = ((chrs(1) And 15) << 2) Or (chrs(2) >> 6)
            enc(3) = chrs(2) And 63
            If chrs(1) = 0 Then
                enc(2) = 64
                enc(3) = 64
            End If
            If chrs(2) = 0 Then
                enc(3) = 64
            End If
            For Each x As Integer In enc
                out.Append(key(x))
            Next
        Loop While i < input.Length
        Return out.ToString
    End Function
    Public Function MEGAN35_Decode(ByVal input As String) As String
        Dim key As String =
"3GHIJKLMNOPQRSTUbcdefghijklmnopWXYZ/12+406789VqrstuvwxyzABCDEF5"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs() As Integer = {0, 0, 0}
            For b As Integer = 0 To 3
                enc(b) = key.IndexOf(input(i))
                i = i + 1
            Next
            chrs(0) = (enc(0) << 2) Or (enc(1) >> 4)

```

```

    chrs(1) = (enc(1) And 15) << 4 Or (enc(2) >> 2)
    chrs(2) = (enc(2) And 3) << 6 Or enc(3)
    out.Append(Chr(chrs(0)))
    If enc(2) <> 64 Then out.Append(Chr(chrs(1)))
    If enc(3) <> 64 Then out.Append(Chr(chrs(2)))
Loop While i < input.Length
Return out.ToString
End Function
#End Region
ZONG 22:
Code #Region "Zong 22"
    Public Function ZONG22_Encode(ByVal input As String) As String
        input = Uri.EscapeDataString(input)
        Dim key As String =
"ZKj9n+yf0wDVX1s/5YbdxSo=ILaUpPBCHg8uvNO4klm6iJGhQ7eFrWczAMEq3RTt2"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs As Integer() = {0, 0, 0}
            For b As Integer = 0 To 2
                If i < input.Length Then chrs(b) = Asc(input(i))
                i += 1
            Next
            enc(0) = chrs(0) >> 2
            enc(1) = ((chrs(0) And 3) << 4) Or (chrs(1) >> 4)
            enc(2) = ((chrs(1) And 15) << 2) Or (chrs(2) >> 6)
            enc(3) = chrs(2) And 63
            If chrs(1) = 0 Then
                enc(2) = 64
                enc(3) = 64
            End If
            If chrs(2) = 0 Then
                enc(3) = 64
            End If
            For Each x As Integer In enc
                out.Append(key(x))
            Next
            Loop While i < input.Length
            Return out.ToString
        End Function
        Public Function ZONG22_Decode(ByVal input As String) As String
            Dim key As String =
"ZKj9n+yf0wDVX1s/5YbdxSo=ILaUpPBCHg8uvNO4klm6iJGhQ7eFrWczAMEq3RTt2"
            Dim out As New System.Text.StringBuilder
            Dim i As Integer
            Do
                Dim enc(3) As Integer
                Dim chrs() As Integer = {0, 0, 0}
                For b As Integer = 0 To 3
                    enc(b) = key.IndexOf(input(i))
                    i = i + 1
                Next

```

```

    chrs(0) = (enc(0) << 2) Or (enc(1) >> 4)
    chrs(1) = (enc(1) And 15) << 4 Or (enc(2) >> 2)
    chrs(2) = (enc(2) And 3) << 6 Or enc(3)
    out.Append(Chr(chrs(0)))
    If enc(2) <> 64 Then out.Append(Chr(chrs(1)))
    If enc(3) <> 64 Then out.Append(Chr(chrs(2)))
    Loop While i < input.Length
    Return out.ToString
End Function

```

#End Region

Tripo 5:

Code #Region "Tripo 5"

```

    Public Function TRIPO5_Encode(ByVal input As String) As String
        input = Uri.EscapeDataString(input)
        Dim key As String =
"ghijopE+G78lmnIJQRXY=abcS/UVWdefABCS456tDqruvNOPwx2KLyz01M3Hk9ZFT"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs As Integer() = {0, 0, 0}
            For b As Integer = 0 To 2
                If i < input.Length Then chrs(b) = Asc(input(i))
                i += 1
            Next
            enc(0) = chrs(0) >> 2
            enc(1) = ((chrs(0) And 3) << 4) Or (chrs(1) >> 4)
            enc(2) = ((chrs(1) And 15) << 2) Or (chrs(2) >> 6)
            enc(3) = chrs(2) And 63
            If chrs(1) = 0 Then
                enc(2) = 64
                enc(3) = 64
            End If
            If chrs(2) = 0 Then
                enc(3) = 64
            End If
            For Each x As Integer In enc
                out.Append(key(x))
            Next
        Loop While i < input.Length
        Return out.ToString
    End Function

    Public Function TRIPO5_Decode(ByVal input As String) As String
        Dim key As String =
"ghijopE+G78lmnIJQRXY=abcS/UVWdefABCS456tDqruvNOPwx2KLyz01M3Hk9ZFT"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs() As Integer = {0, 0, 0}
            For b As Integer = 0 To 3
                enc(b) = key.IndexOf(input(i))
                i = i + 1
            Next
        Loop While i < input.Length
        Return out.ToString
    End Function

```

```

        Next
        chrs(0) = (enc(0) << 2) Or (enc(1) >> 4)
        chrs(1) = (enc(1) And 15) << 4 Or (enc(2) >> 2)
        chrs(2) = (enc(2) And 3) << 6 Or enc(3)
        out.Append(Chr(chrs(0)))
        If enc(2) <> 64 Then out.Append(Chr(chrs(1)))
        If enc(3) <> 64 Then out.Append(Chr(chrs(2)))
    Loop While i < input.Length
    Return out.ToString
End Function
#End Region
TIGO3FX:
Code #Region "TIGO3FX"
    Public Function TIGO3FX_Encode(ByVal input As String) As String
        input = Uri.EscapeDataString(input)
        Dim key As String =
"FrsxyzA8VtuwvDEqWZ/1+4klm67=cBCa5Ybdef0g2hij9nopMNO3GHIRSTJKLPQUX"
        Dim out As New System.Text.StringBuilder
        Dim i As Integer
        Do
            Dim enc(3) As Integer
            Dim chrs As Integer() = {0, 0, 0}
            For b As Integer = 0 To 2
                If i < input.Length Then chrs(b) = Asc(input(i))
                i += 1
            Next
            enc(0) = chrs(0) >> 2
            enc(1) = ((chrs(0) And 3) << 4) Or (chrs(1) >> 4)
            enc(2) = ((chrs(1) And 15) << 2) Or (chrs(2) >> 6)
            enc(3) = chrs(2) And 63
            If chrs(1) = 0 Then
                enc(2) = 64
                enc(3) = 64
            End If
            If chrs(2) = 0 Then
                enc(3) = 64
            End If
            For Each x As Integer In enc
                out.Append(key(x))
            Next
            Loop While i < input.Length
            Return out.ToString
        End Function
        Public Function TIGO3FX_Decode(ByVal input As String) As String
            Dim key As String =
"FrsxyzA8VtuwvDEqWZ/1+4klm67=cBCa5Ybdef0g2hij9nopMNO3GHIRSTJKLPQUX"
            Dim out As New System.Text.StringBuilder
            Dim i As Integer
            Do
                Dim enc(3) As Integer
                Dim chrs() As Integer = {0, 0, 0}
                For b As Integer = 0 To 3
                    enc(b) = key.IndexOf(input(i))

```

```

i = i + 1
Next
chrs(0) = (enc(0) << 2) Or (enc(1) >> 4)
chrs(1) = (enc(1) And 15) << 4 Or (enc(2) >> 2)
chrs(2) = (enc(2) And 3) << 6 Or enc(3)
out.Append(Chr(chrs(0)))
If enc(2) <> 64 Then out.Append(Chr(chrs(1)))
If enc(3) <> 64 Then out.Append(Chr(chrs(2)))
Loop While i < input.Length
Return out.ToString
End Function
#End Region
FERON 74:
Code #Region "FERON 74"
Public Function FERON74_Encode(ByVal input As String) As String
    input = Uri.EscapeDataString(input)
    Dim key As String =
"75XYTabcS/UVWdefADqr6RuvN8PBCsQtwx2KLyz+OM3Hk9ghi01ZFlmnjopE=GIJ4"
    Dim out As New System.Text.StringBuilder
    Dim i As Integer
    Do
        Dim enc(3) As Integer
        Dim chrs As Integer() = {0, 0, 0}
        For b As Integer = 0 To 2
            If i < input.Length Then chrs(b) = Asc(input(i))
            i += 1
        Next
        enc(0) = chrs(0) >> 2
        enc(1) = ((chrs(0) And 3) << 4) Or (chrs(1) >> 4)
        enc(2) = ((chrs(1) And 15) << 2) Or (chrs(2) >> 6)
        enc(3) = chrs(2) And 63
        If chrs(1) = 0 Then
            enc(2) = 64
            enc(3) = 64
        End If
        If chrs(2) = 0 Then
            enc(3) = 64
        End If
        For Each x As Integer In enc
            out.Append(key(x))
        Next
        Loop While i < input.Length
        Return out.ToString
    End Function
Public Function FERON74_Decode(ByVal input As String) As String
    Dim key As String =
"75XYTabcS/UVWdefADqr6RuvN8PBCsQtwx2KLyz+OM3Hk9ghi01ZFlmnjopE=GIJ4"
    Dim out As New System.Text.StringBuilder
    Dim i As Integer
    Do
        Dim enc(3) As Integer
        Dim chrs() As Integer = {0, 0, 0}
        For b As Integer = 0 To 3

```

```

enc(b) = key.IndexOf(input(i))
i = i + 1
Next
chrs(0) = (enc(0) << 2) Or (enc(1) >> 4)
chrs(1) = (enc(1) And 15) << 4 Or (enc(2) >> 2)
chrs(2) = (enc(2) And 3) << 6 Or enc(3)
out.Append(Chr(chrs(0)))
If enc(2) <> 64 Then out.Append(Chr(chrs(1)))
If enc(3) <> 64 Then out.Append(Chr(chrs(2)))
Loop While i < input.Length
Return out.ToString
End Function
#End Region
ZARA 128
Code #Region "ZARA 128"
Public Function ZARA128_Encode(ByVal input As String) As String
    Dim out As New System.Text.StringBuilder
    For Each c As Char In input
        Dim temp As Integer = Asc(c) + 312
        out.Append(temp.ToString & " ")
    Next
    Return out.ToString.Substring(0, out.Length - 1)
End Function
Public Function ZARA128_Decode(ByVal input As String) As String
    Dim out As New System.Text.StringBuilder
    Dim data As String() = Split(input, " ")
    For Each s As String In data
        out.Append(Chr(Asc(s) - 312))
    Next
    Return out.ToString
End Function
#End Region
Base 64:
Code #Region "Base 64"
Public Function BASE64_Encode(ByVal input As String) As String
    Return Convert.ToBase64String(System.Text.Encoding.ASCII.GetBytes(input))
End Function
Public Function BASE64_Decode(ByVal input As String) As String
    Return System.Text.Encoding.ASCII.GetString(Convert.FromBase64String(input))
End Function
#End Region
ARMON 64:
Code #Region "ARMON 64"
Public Function ARMON64_Encrypt(ByVal message As String, Optional ByVal key As String =
"ARMON64-CRYPO") As String
    Dim out As New System.Text.StringBuilder
    If key.Length < 3 Then Return message
    Dim intkey() As Byte = System.Text.Encoding.UTF8.GetBytes(key)
    Dim x As Integer
    Do While x < message.Length
        Dim hextemp As String = ""
        Dim y As String = ""
        If x > 0 Then y = "+"

```

```

For i As Integer = x To Math.Round(key.Length / 2)
If i < message.Length Then hextemp += Hex(Asc(message(i)))
Next
Dim thenum As Double = "&H" & hextemp
If Information.IsNumeric(thenum) = False Then Return message
For z As Integer = 0 To key.Length - 1
Dim operation As Integer = z Mod 4
Select Case operation
Case 0
thenum += intkey(z)
Case 1
thenum /= intkey(z)
Case 2
thenum -= intkey(z)
Case 3
thenum *= 0.01 * intkey(z)
End Select
Next
out.Append(y & thenum)
x += Math.Round(key.Length / 2)
Loop
Return out.ToString.Replace(",", ".")
End Function

Public Function ARMON64_Decrypt(ByVal message As String, Optional ByVal key As String =
"ARMON64-CRYPO") As String
Dim out As New System.Text.StringBuilder
If key.Length < 6 Then Return message
Dim intkey() As Byte = System.Text.Encoding.UTF8.GetBytes(key)
message = message.Replace(".", ";")
Dim oOutString As String() = Split(message, "+")
For x As Integer = 0 To oOutString.Length - 1
For z As Integer = key.Length - 1 To 0 Step -1
Dim operation As Integer = z Mod 4
Select Case operation
Case 0
oOutString(x) -= intkey(z)
Case 1
oOutString(x) *= intkey(z)
Case 2
oOutString(x) += intkey(z)
Case 3
oOutString(x) /= 0.01 * intkey(z)
End Select
Next
oOutString(x) = Hex(Math.Round(Double.Parse(oOutString(x))))
Next
For i As Integer = 0 To Join(oOutString).Length - 1 Step +2
out.Append(Chr("&H" & Join(oOutString).Substring(i, 2)))
Next
Return out.ToString
End Function
#End Region
EZIP 64:

```

Code #Region "EZIP 64"

Public Function EZIP64_Encrypt(ByVal message As String, Optional ByVal key As String = "EZIP64-CRYPO") As String

```
Dim out As New System.Text.StringBuilder
If key.Length < 10 Then Return message
Dim intkey() As Byte = System.Text.Encoding.UTF8.GetBytes(key)
Dim x As Integer
Do While x < message.Length
Dim hextemp As String = ""
Dim y As String = ""
If x > 0 Then y = "/"
For i As Integer = x To Math.Round(key.Length / 3)
If i < message.Length Then hextemp += Hex(Asc(message(i)))
Next
Dim thenum As Double = "&H" & hextemp
If Information.IsNumeric(thenum) = False Then Return message
For z As Integer = 0 To key.Length - 1
Dim operation As Integer = z Mod 4
Select Case operation
Case 0
thenum += intkey(z)
Case 1
thenum /= intkey(z)
Case 2
thenum -= intkey(z)
Case 3
thenum *= 0.02 * intkey(z)
End Select
Next
Dim temp As String = thenum.ToString.Replace(",", ".")
out.Append(y & temp)
x += Math.Round(key.Length / 3)
Loop
Return out.ToString
```

End Function

Public Function EZIP64_Decrypt(ByVal message As String, Optional ByVal key As String = "EZIP64-CRYPO") As String

```
Dim out As New System.Text.StringBuilder
If key.Length < 10 Then Return message
Dim intkey() As Byte = System.Text.Encoding.UTF8.GetBytes(key)

Dim oOutString As String() = Split(message, "/")
For i As Integer = 0 To oOutString.Length - 1
oOutString(i) = oOutString(i).Replace(".", ",")
Next
For x As Integer = 0 To oOutString.Length - 1
For z As Integer = key.Length - 1 To 0 Step -1
Dim operation As Integer = z Mod 4
Select Case operation
Case 0
oOutString(x) -= intkey(z)
Case 1
oOutString(x) *= intkey(z)
```

```

Case 2
oOutString(x) += intkey(z)
Case 3
oOutString(x) /= 0.02 * intkey(z)
End Select
Next
oOutString(x) = Hex(Math.Round(Double.Parse(oOutString(x))))
Next
For i As Integer = 0 To Join(oOutString).Length - 1 Step +2
out.Append(Chr("&H" & Join(oOutString).Substring(i, 2)))
Dim j As String = out.ToString
Next
Return out.ToString
End Function
#End Region
OKTO3:
Code #Region "OKTO3"
Public Function OKTO3_Encrypt(ByVal message As String, Optional ByVal key As String =
"PASS:OKTO3-CRYPO") As String
Dim out As New System.Text.StringBuilder
If key.Length < 10 Then Return message
Dim intkey() As Byte = System.Text.Encoding.UTF8.GetBytes(key)
Dim x As Integer
Do While x < message.Length
Dim hextemp As String = ""
Dim y As String = ""
If x > 0 Then y = ", "
For i As Integer = x To Math.Round(key.Length / 6)
If i < message.Length Then hextemp += Hex(Asc(message(i)))
Next
Dim thenum As Double = "&H" & hextemp
If Information.IsNumeric(thenum) = False Then Return message
For z As Integer = 0 To key.Length - 1
Dim operation As Integer = z Mod 3
Select Case operation
Case 0
thenum += intkey(z)
Case 1
thenum /= intkey(z)
Case 2
thenum -= intkey(z)
Case 3
thenum *= 500.005 * intkey(z)
End Select
Next
Dim temp As String = thenum.ToString.Replace(",", ".")
out.Append(y & temp)
x += Math.Round(key.Length / 6)
Loop
Return out.ToString
End Function
Public Function OKTO3_Decrypt(ByVal message As String, Optional ByVal key As String =
"PASS:OKTO3-CRYPO") As String

```

```

Dim out As New System.Text.StringBuilder
If key.Length < 10 Then Return message
Dim intkey() As Byte = System.Text.Encoding.UTF8.GetBytes(key)

Dim oOutString As String() = Split(message, ", ")
For i As Integer = 0 To oOutString.Length - 1
oOutString(i) = oOutString(i).Replace(".", ",")
Next
For x As Integer = 0 To oOutString.Length - 1
For z As Integer = key.Length - 1 To 0 Step -1
Dim operation As Integer = z Mod 3
Select Case operation
Case 0
oOutString(x) -= intkey(z)
Case 1
oOutString(x) *= intkey(z)
Case 2
oOutString(x) += intkey(z)
Case 3
oOutString(x) /= 0.02 * intkey(z)
End Select
Next
oOutString(x) = Hex(Math.Round(Double.Parse(oOutString(x))))
Next
For i As Integer = 0 To Join(oOutString).Length - 1 Step +2
out.Append(Chr("&H" & Join(oOutString).Substring(i, 2)))
Next
Return out.ToString
End Function
#End Region
RC4:
Code #Region "Rc4"
Private Function Rc4(ByVal message As String, ByVal password As String) As String
Dim s = Enumerable.Range(0, 256).ToArray
Dim i, j As Integer
For i = 0 To s.Length - 1
j = (j + Asc(password(i Mod password.Length)) + s(i)) And 255
Dim temp = s(i)
s(i) = s(j)
s(j) = temp
Next
i = 0
j = 0
Dim sb As New StringBuilder(message.Length)
For Each c As Char In message
i = (i + 1) And 255
j = (j + s(i)) And 255
Dim temp = s(i)
s(i) = s(j)
s(j) = temp
sb.Append(Chr(s((s(i) + s(j)) And 255) Xor Asc(c)))
Next
Return sb.ToString

```

```

End Function
#End Region
Hash's:
Code #Region "Hash's"
Public Function MD5Hash(ByVal input As String) As String
    Dim MD5 As New System.Security.Cryptography.MD5CryptoServiceProvider
    Dim Data As Byte()
    Dim Result As Byte()
    Dim Res As String = ""
    Dim Tmp As String = ""
    Data = System.Text.Encoding.ASCII.GetBytes(input)
    Result = MD5.ComputeHash(Data)
    For i As Integer = 0 To Result.Length - 1
        Tmp = Hex(Result(i))
        If Len(Tmp) = 1 Then Tmp = "0" & Tmp
        Res += Tmp
    Next
    Return Res
End Function
Public Function RIPEMD160Hash(ByVal input As String) As String
    Dim RIPEMD160 As New System.Security.Cryptography.RIPEMD160Managed
    Dim Data As Byte()
    Dim Result As Byte()
    Dim Res As String = ""
    Dim Tmp As String = ""
    Data = System.Text.Encoding.ASCII.GetBytes(input)
    Result = RIPEMD160.ComputeHash(Data)
    For i As Integer = 0 To Result.Length - 1
        Tmp = Hex(Result(i))
        If Len(Tmp) = 1 Then Tmp = "0" & Tmp
        Res += Tmp
    Next
    Return Res
End Function
Public Function SHA1Hash(ByVal input As String) As String
    Dim SHA1 As New System.Security.Cryptography.SHA1CryptoServiceProvider
    Dim Data As Byte()
    Dim Result As Byte()
    Dim Res As String = ""
    Dim Tmp As String = ""
    Data = System.Text.Encoding.ASCII.GetBytes(input)
    Result = SHA1.ComputeHash(Data)
    For i As Integer = 0 To Result.Length - 1
        Tmp = Hex(Result(i))
        If Len(Tmp) = 1 Then Tmp = "0" & Tmp
        Res += Tmp
    Next
    Return Res
End Function
Public Function SHA256Hash(ByVal input As String) As String
    Dim SHA256 As New System.Security.Cryptography.SHA256Managed
    Dim Data As Byte()
    Dim Result As Byte()

```

```

Dim Res As String = ""
Dim Tmp As String = ""
Data = System.Text.Encoding.ASCII.GetBytes(input)
Result = SHA256.ComputeHash(Data)
For i As Integer = 0 To Result.Length - 1
    Tmp = Hex(Result(i))
    If Len(Tmp) = 1 Then Tmp = "0" & Tmp
    Res += Tmp
Next
Return Res
End Function

Public Function SHA348Hash(ByVal input As String) As String
    Dim SHA348 As New System.Security.Cryptography.SHA384Managed
    Dim Data As Byte()
    Dim Result As Byte()
    Dim Res As String = ""
    Dim Tmp As String = ""
    Data = System.Text.Encoding.ASCII.GetBytes(input)
    Result = SHA348.ComputeHash(Data)
    For i As Integer = 0 To Result.Length - 1
        Tmp = Hex(Result(i))
        If Len(Tmp) = 1 Then Tmp = "0" & Tmp
        Res += Tmp
    Next
    Return Res
End Function

Public Function SHA512Hash(ByVal input As String) As String
    Dim SHA512 As New System.Security.Cryptography.SHA512Managed
    Dim Data As Byte()
    Dim Result As Byte()
    Dim Res As String = ""
    Dim Tmp As String = ""
    Data = System.Text.Encoding.ASCII.GetBytes(input)
    Result = SHA512.ComputeHash(Data)
    For i As Integer = 0 To Result.Length - 1
        Tmp = Hex(Result(i))
        If Len(Tmp) = 1 Then Tmp = "0" & Tmp
        Res += Tmp
    Next
    Return Res
End Function
#End Region

Ceaser Shift:
Code #Region "Ceaser shift"
Public Function Encrypt(ByVal PlainText As String, ByVal Key As Integer) As String
    Dim PlainChar() As Char = PlainText.ToCharArray()
    Dim Ascii(PlainChar.Length) As Integer
    For Count As Integer = 0 To PlainChar.Length - 1
        Ascii(Count) = Asc(PlainChar(Count))
        If Ascii(Count) >= 65 And Ascii(Count) <= 90 Then
            Ascii(Count) = ((Ascii(Count) - 65 + Key) Mod 26) + 65
        ElseIf Ascii(Count) >= 97 And Ascii(Count) <= 122 Then
            Ascii(Count) = ((Ascii(Count) - 97 + Key) Mod 26) + 97
        End If
    Next
    Return New String(Ascii)
End Function

```

```

End If
PlainChar(Count) = Chr(Ascii(Count))
Next
Return PlainChar
End Function
Public Function Decrypt(ByVal CipherText As String, ByVal Key As Integer) As String
    Dim CipherChar() As Char = CipherText.ToCharArray()
    Dim Ascii(CipherChar.Length) As Integer

    For Count As Integer = 0 To CipherChar.Length - 1
        Ascii(Count) = Asc(CipherChar(Count))
        If Ascii(Count) >= 65 And Ascii(Count) <= 90 Then
            Ascii(Count) = ((Ascii(Count) - 65 - (Key Mod 26) + 26)) Mod 26 + 65
        ElseIf Ascii(Count) >= 97 And Ascii(Count) <= 122 Then
            Ascii(Count) = (((Ascii(Count) - 97 - (Key Mod 26) + 26)) Mod 26) + 97
        End If
        CipherChar(Count) = Chr(Ascii(Count))
    Next
    Return CipherChar
End Function
#End Region
Stairs:
Code #Region "Stairs"
    Public Shared Function Crypt(ByVal Data As String, ByVal key As String) As String
        Return Encoding.Default.GetString(Crypt(Encoding.Default.GetBytes(Data),
Encoding.Default.GetBytes(key)))
    End Function
    Public Shared Function Crypt(ByVal Data() As Byte, ByVal key() As Byte) As Byte()
        For i = 0 To (Data.Length * 2) + key.Length
            Data(i Mod Data.Length) = CByte(CInt((Data(i Mod Data.Length)) + CInt(Data((i + 1) Mod
Data.Length))) Mod 256) Xor key(i Mod key.Length)
        Next
        Return Data
    End Function
    Public Shared Function DeCrypt(ByVal Data As String, ByVal key As String) As String
        Return Encoding.Default.GetString(DeCrypt(Encoding.Default.GetBytes(Data),
Encoding.Default.GetBytes(key)))
    End Function
    Public Shared Function DeCrypt(ByVal Data() As Byte, ByVal key() As Byte) As Byte()
        For i = (Data.Length * 2) + key.Length To 0 Step -1
            Data(i Mod Data.Length) = CByte((CInt(Data(i Mod Data.Length) Xor key(i Mod key.Length)) -
CInt(Data((i + 1) Mod Data.Length)) + 256) Mod 256)
        Next
        Return Data
    End Function
#End Region
ROT13:
Code #Region "ROT13"
    Public Function ROT13(ByVal input As String) As String
        Dim out As New System.Text.StringBuilder
        For i As Integer = 0 To input.Length - 1
            out.Append(Chr(Asc(input(i)) Xor 13))
        Next

```

```

    Return out.ToString
End Function
#End Region
RSA:
Code #Region "RSA"
Public Function RSA_Encrypt(ByVal Input As String) As String
    Dim cp As New Security.Cryptography.CspParameters
    cp.Flags = Security.Cryptography.CspProviderFlags.UseMachineKeyStore
    cp.KeyContainerName = "Keys"
    Dim RSA As New Security.Cryptography.RSACryptoServiceProvider(cp)

    Dim buffer As Byte() = System.Text.Encoding.UTF8.GetBytes(Input)
    Dim encrypted As Byte() = RSA.Encrypt(buffer, True)
    Return Convert.ToBase64String(encrypted)
End Function
Public Function RSA_Decrypt(ByVal Input As String) As String
    Dim cp As New Security.Cryptography.CspParameters
    cp.Flags = Security.Cryptography.CspProviderFlags.UseMachineKeyStore
    cp.KeyContainerName = "Keys"
    Dim RSA As New Security.Cryptography.RSACryptoServiceProvider(cp)
    Dim buffer As Byte() = Convert.FromBase64String(Input)
    Dim decrypted As Byte() = RSA.Decrypt(buffer, True)
    Return System.Text.Encoding.UTF8.GetString(decrypted)
End Function
#End Region
XOR:
Code #Region "XOR"
#Region "Normal"
Public Function XOR_Encrypt(ByVal Input As String, ByVal pass As String) As String
    Dim out As New System.Text.StringBuilder
    Dim u As Integer
    For i As Integer = 0 To Input.Length - 1
        Dim tmp As String = Hex(Asc(Input(i)) Xor Asc(pass(u)))
        If tmp.Length = 1 Then tmp = "0" & tmp
        out.Append(tmp)
        If u = pass.Length - 1 Then u = 0 Else u = u + 1
    Next
    Return out.ToString
End Function
Public Function XOR_Decrypt(ByVal Input As String, ByVal pass As String) As String
    Dim out As New System.Text.StringBuilder
    Dim u As Integer
    For i As Integer = 0 To Input.Length - 1 Step +2
        Dim tmp As String = Chr("&H" & Input.Substring(i, 2)) Xor Asc(pass(u))
        out.Append(tmp)
        If u = pass.Length - 1 Then u = 0 Else u = u + 1
    Next
    Return out.ToString
End Function
#End Region
#Region "Custom"

Public Function CustomXOR_Encrypt(ByVal Input As String, ByVal pass As String) As String

```

```

Dim out As New System.Text.StringBuilder
Dim Hash As New System.Security.Cryptography.MD5CryptoServiceProvider
Dim XORHash As Byte() = Hash.ComputeHash(System.Text.Encoding.ASCII.GetBytes(pass))
Dim u As Integer
For i As Integer = 0 To Input.Length - 1
Dim tmp As String = Hex(Asc(Input(i)) XOR XORHash(u))
If tmp.Length = 1 Then tmp = "0" & tmp
out.Append(tmp)
If u = pass.Length - 1 Then u = 0 Else u = u + 1
Next
Return out.ToString
End Function

Public Function CustomXOR_Decrypt(ByVal Input As String, ByVal pass As String) As String
Dim out As New System.Text.StringBuilder
Dim Hash As New System.Security.Cryptography.MD5CryptoServiceProvider
Dim XORHash As Byte() = Hash.ComputeHash(System.Text.Encoding.ASCII.GetBytes(pass))
Dim u As Integer
For i As Integer = 0 To Input.Length - 1 Step +2
Dim tmp As String = Chr(("&H" & Input.Substring(i, 2)) XOR XORHash(u))
out.Append(tmp)
If u = pass.Length - 1 Then u = 0 Else u = u + 1
Next
Return out.ToString
End Function

#End Region
#End Region
RC2:
Code #Region "Rc2"
Public Function RC2_Encrypt(ByVal input As String, ByVal pass As String) As String
Dim RC2 As New System.Security.Cryptography.RC2CryptoServiceProvider
Dim Hash_RC2 As New System.Security.Cryptography.MD5CryptoServiceProvider
Dim encrypted As String = ""
Try
Dim hash() As Byte =
Hash_RC2.ComputeHash(System.Text.Encoding.ASCII.GetBytes(pass))

RC2.Key = hash
RC2.Mode = Security.Cryptography.CipherMode.ECB
Dim DESEncrypter As System.Security.Cryptography.ICryptoTransform = RC2.CreateEncryptor
Dim Buffer As Byte() = System.Text.Encoding.ASCII.GetBytes(input)
encrypted = Convert.ToBase64String(DESEncrypter.TransformFinalBlock(Buffer, 0,
Buffer.Length))
Return encrypted
Catch ex As Exception
End Try
End Function
Public Function RC2_Decrypt(ByVal input As String, ByVal pass As String) As String
Dim RC2 As New System.Security.Cryptography.RC2CryptoServiceProvider
Dim Hash_RC2 As New System.Security.Cryptography.MD5CryptoServiceProvider
Dim decrypted As String = ""
Try
Dim hash() As Byte =

```

```

Hash_RC2.ComputeHash(System.Text.ASCIIEncoding.ASCII.GetBytes(pass))
    RC2.Key = hash
    RC2.Mode = Security.Cryptography.CipherMode.ECB
    Dim DESDecrypter As System.Security.Cryptography.ICryptoTransform = RC2.CreateDecryptor
    Dim Buffer As Byte() = Convert.FromBase64String(input)
    decrypted =
System.Text.ASCIIEncoding.ASCII.GetString(DESDecrypter.TransformFinalBlock(Buffer, 0,
Buffer.Length))
    Return decrypted
    Catch ex As Exception
    End Try
End Function
#End Region
DES:
Code #Region "DES"
    Public Function DES_Encrypt(ByVal input As String, ByVal pass As String) As String
        Dim DES As New System.Security.Cryptography.DESCryptoServiceProvider
        Dim Hash_DES As New System.Security.Cryptography.MD5CryptoServiceProvider
        Dim encrypted As String = ""
        Try
            Dim hash(7) As Byte
            Dim temp As Byte() =
Hash_DES.ComputeHash(System.Text.ASCIIEncoding.ASCII.GetBytes(pass))
            Array.Copy(temp, 0, hash, 0, 8)
            DES.Key = hash
            DES.Mode = Security.Cryptography.CipherMode.ECB
            Dim DESEncrypter As System.Security.Cryptography.ICryptoTransform = DES.CreateEncryptor
            Dim Buffer As Byte() = System.Text.ASCIIEncoding.ASCII.GetBytes(input)
            encrypted = Convert.ToBase64String(DESEncrypter.TransformFinalBlock(Buffer, 0,
Buffer.Length))
            Return encrypted
            Catch ex As Exception
            End Try
        End Function
        Public Function DES_Decrypt(ByVal input As String, ByVal pass As String) As String
            Dim DES As New System.Security.Cryptography.DESCryptoServiceProvider
            Dim Hash_DES As New System.Security.Cryptography.MD5CryptoServiceProvider
            Dim decrypted As String = ""
            Try
                Dim hash(7) As Byte
                Dim temp As Byte() =
Hash_DES.ComputeHash(System.Text.ASCIIEncoding.ASCII.GetBytes(pass))
                Array.Copy(temp, 0, hash, 0, 8)
                DES.Key = hash
                DES.Mode = Security.Cryptography.CipherMode.ECB
                Dim DESDecrypter As System.Security.Cryptography.ICryptoTransform = DES.CreateDecryptor
                Dim Buffer As Byte() = Convert.FromBase64String(input)
                decrypted =
System.Text.ASCIIEncoding.ASCII.GetString(DESDecrypter.TransformFinalBlock(Buffer, 0,
Buffer.Length))
                Return decrypted
                Catch ex As Exception
                End Try
            End Try
        End Try
    End Try

```

```

End Function
#End Region
Triple DES:
Code #Region "Triple DES"
Public Function TripleDES_Encrypt(ByVal input As String, ByVal pass As String) As String
    Dim TripleDES As New System.Security.Cryptography.TripleDESCryptoServiceProvider
    Dim Hash_TripleDES As New System.Security.Cryptography.MD5CryptoServiceProvider
    Dim encrypted As String = ""
    Try
        Dim hash(23) As Byte
        Dim temp As Byte() =
Hash_TripleDES.ComputeHash(System.Text.ASCIIEncoding.ASCII.GetBytes(pass))
        Array.Copy(temp, 0, hash, 0, 16)
        Array.Copy(temp, 0, hash, 15, 8)
        TripleDES.Key = hash
        TripleDES.Mode = Security.Cryptography.CipherMode.ECB
        Dim DESEncrypter As System.Security.Cryptography.ICryptoTransform =
TripleDES.CreateEncryptor
        Dim Buffer As Byte() = System.Text.ASCIIEncoding.ASCII.GetBytes(input)
        encrypted = Convert.ToBase64String(DESEncrypter.TransformFinalBlock(Buffer, 0,
Buffer.Length))
        Return encrypted
    Catch ex As Exception
    End Try
End Function
Public Function TripleDES_Decrypt(ByVal input As String, ByVal pass As String) As String
    Dim TripleDES As New System.Security.Cryptography.TripleDESCryptoServiceProvider
    Dim Hash_TripleDES As New System.Security.Cryptography.MD5CryptoServiceProvider
    Dim decrypted As String = ""
    Try
        Dim hash(23) As Byte
        Dim temp As Byte() =
Hash_TripleDES.ComputeHash(System.Text.ASCIIEncoding.ASCII.GetBytes(pass))
        Array.Copy(temp, 0, hash, 0, 16)
        Array.Copy(temp, 0, hash, 15, 8)
        TripleDES.Key = hash
        TripleDES.Mode = Security.Cryptography.CipherMode.ECB
        Dim DESDecrypter As System.Security.Cryptography.ICryptoTransform =
TripleDES.CreateDecryptor
        Dim Buffer As Byte() = Convert.FromBase64String(input)
        decrypted =
System.Text.ASCIIEncoding.ASCII.GetString(DESDecrypter.TransformFinalBlock(Buffer, 0,
Buffer.Length))
        Return decrypted
    Catch ex As Exception
    End Try
End Function

#End Region
Rijndael:
Code #Region "Rijndael"
Public Shared Function RijndaelDecrypt(ByVal UDecryptU As String, ByVal UKeyU As String)
    Dim XoAesProviderX As New RijndaelManaged

```

```

Dim XbtCipherX() As Byte
Dim XbtSaltX() As Byte = New Byte() {1, 2, 3, 4, 5, 6, 7, 8}
Dim XoKeyGeneratorX As New Rfc2898DeriveBytes(UKeyU, XbtSaltX)
XoAesProviderX.Key = XoKeyGeneratorX.GetBytes(XoAesProviderX.Key.Length)
XoAesProviderX.IV = XoKeyGeneratorX.GetBytes(XoAesProviderX.IV.Length)
Dim XmsX As New IO.MemoryStream
Dim XcsX As New CryptoStream(XmsX, XoAesProviderX.CreateDecryptor(), _
CryptoStreamMode.Write)
Try
XbtCipherX = Convert.FromBase64String(UDecryptU)
XcsX.Write(XbtCipherX, 0, XbtCipherX.Length)
XcsX.Close()
UDecryptU = System.Text.Encoding.UTF8.GetString(XmsX.ToArray)
Catch
End Try
Return UDecryptU
End Function

Public Shared Function Rijndaelcrypt(ByVal File As String, ByVal Key As String)
Dim oAesProvider As New RijndaelManaged
Dim btClear() As Byte
Dim btSalt() As Byte = New Byte() {1, 2, 3, 4, 5, 6, 7, 8}
Dim oKeyGenerator As New Rfc2898DeriveBytes(Key, btSalt)
oAesProvider.Key = oKeyGenerator.GetBytes(oAesProvider.Key.Length)
oAesProvider.IV = oKeyGenerator.GetBytes(oAesProvider.IV.Length)
Dim ms As New IO.MemoryStream
Dim cs As New CryptoStream(ms, _
oAesProvider.CreateEncryptor(), _
CryptoStreamMode.Write)
btClear = System.Text.Encoding.UTF8.GetBytes(File)
cs.Write(btClear, 0, btClear.Length)
cs.Close()
File = Convert.ToBase64String(ms.ToArray)
Return File
End Function
#End Region
HEX:
Code #Region "Hex"
Public Function String2Hex(ByVal input As String) As String
Dim out As New System.Text.StringBuilder
For Each c As String In input
Dim temp As String = Hex(Asc(c))
out.Append(temp & " ")
Next
Return out.ToString.Substring(0, out.Length - 1)
End Function
Public Function Hex2String(ByVal input As String) As String
Dim out As New System.Text.StringBuilder
Dim data As String() = Split(input, " ")
For Each s As String In data
out.Append(Chr("&H" & s))
Next
Return out.ToString
End Function

```

#End Region

Binary:

Code #Region "Binary"

```
Private Function ConvertToAscii(ByVal str As String) As String
```

```
    Dim chars As String = Regex.Replace(str, "[^01]", "")
```

```
    Dim arr((chars.Length / 8) - 1) As Byte
```

```
    For i As Integer = 0 To arr.Length - 1
```

```
        arr(i) = Convert.ToByte(chars.Substring(i * 8, 8), 2)
```

```
    Next
```

```
    Return ASCIIEncoding.ASCII.GetString(arr)
```

```
End Function
```

```
Private Function ConvertToBinary(ByVal str As String) As String
```

```
    Dim converted As New StringBuilder
```

```
    For Each b As Byte In ASCIIEncoding.ASCII.GetBytes(str)
```

```
        converted.Append(Convert.ToString(b, 2).PadLeft(8, "0"))
```

```
    Next
```

```
    Return converted.ToString()
```

```
End Function
```

#End Region

iSynthesis - June 4, 2012, 11:41 pm

Most of them (if not all) are by Pr0t0typ3,

source:

<http://pastebin.com/wEvjPT4s>

Thanks for sharing though, might help people.

????? - June 4, 2012, 11:42 pm

(Jun 4, 2012 11:41 PM)iSynthesis Wrote: [\[CITE\]](#) Most of them (if not all) are by Pr0t0typ3,

source:

<http://pastebin.com/wEvjPT4s>

Thanks for sharing though, might help people.

Looks like he had about 8-10 there on that Pastebin, there is 28 (if I remember right) here. Credits to Pr0t0typ3 for his however.

iSynthesis - June 4, 2012, 11:43 pm

(Jun 4, 2012 11:42 PM)????? Wrote: [\[CITE\]](#) Looks like he had about 8-10 there on that Pastebin, there is 28 (if I remember right) here. Credits to Pr0t0typ3 for his however.

Found it:

<https://hackforums.net/showthread.php?tid=672570>

yeah :)

Mohatu - June 4, 2012, 11:44 pm

Great post, I'm sure it will help out many people.

????? - June 4, 2012, 11:45 pm

(Jun 4, 2012 11:43 PM)iSynthesis Wrote: [\[CITE\]](#) Found it:

<https://hackforums.net/showthread.php?tid=672570>

yeah :)

I stand corrected, 14. Big credit to him then, I wasn't aware that he made more than half of these.

????? - June 4, 2012, 11:51 pm

(Jun 4, 2012 11:49 PM)PoRt21 Wrote: [\[CITE\]](#) tnx man great share :)

101010000100101001001010100110101001001010010010100101010010010010100

You're welcome. I am sure we will see a lot of these in upcoming crypters. DO NOT FORGET TO CREDIT GUYS! Pr0t0typ3 is responsible for about 14 of these.

?ndromeda - June 5, 2012, 12:28 am

Thank you mate !

I might 'convert' it to an open-source application today or tomorrow.

iSynthesis - June 5, 2012, 12:31 am

Iv'e just made a .Dll for all this, if anyone is interested

Download:

<http://db.tt/3ftSJlHv>

VT:

<http://bit.ly> [WARNING: potentially infected]/M2G6uk

Enjoy

????? - June 5, 2012, 2:32 am

(Jun 5, 2012 12:31 AM)iSynthesis Wrote: [\[CITE\]](#) Iv'e just made a .Dll for all this, if anyone is interested

Download:

<http://db.tt/3ftSJlHv>

VT:

<http://bit.ly> [WARNING: potentially infected]/M2G6uk

Enjoy

Whoa, nice man. Thank you. I would add that to the OP with your quote if I could.

ytlayol - June 5, 2012, 2:36 am

awesome..mind adding them into a class...n posting it on pastebin...i mean would be easy to download n use

pr0t0typ3 - June 5, 2012, 4:55 am

Well most of my original functions I made by reverse engenieering the crypto.com function are poorly coded :D I might recode them next week. Anyway nice find haha

lorens13 - June 27, 2012, 5:31 am

WOW! Thank you very much for this :)

????? - June 27, 2012, 4:05 pm

Make sure to credit pr0t0typ3 if you use his functions.

Horrace456 - August 15, 2012, 7:27 am

Thank you very much for posting this, it certainly helped me!

jessman420 - August 15, 2012, 7:33 am

Thanks for posting these and +1 for your trouble.

Hawk HF - August 15, 2012, 7:52 pm

Great release. I made a text encryptor with most of the functions you posted.

Available for download here:

<https://dl.dropbox.com/u/64850826/HackForums/Encrypter/Text%20Encrypter.exe>

Horrace456 - August 23, 2012, 5:50 am

Could these encryption functions be used to encrypt and decrypt strings or API's to help FUD?.. or do these merely constitute the encryption function thats involved in the main encryption of say a server?... or both.. lol

????? - August 23, 2012, 8:43 pm

(Aug 23, 2012 05:50 AM)Horrace456 Wrote: [\[CITE\]](#) Could these encryption functions be used to encrypt and decrypt strings or API's to help FUD?.. or do these merely constitute the encryption function thats involved in the main encryption of say a server?... or both.. lol
... I want to, but I won't.

Strings, they encrypt strings.

Hawk HF - September 25, 2012, 7:37 pm

Can't believe this thread died so quickly. It's a good post.
GRAVEDIG!