جامـــــعة المـــــستقبل
AL MUSTAQBAL UNIVERSITY

قســـم الامـــــن الـــــــــسيبرانـــــــي

Department of Cyber Security

Subject:

Key of DES

Class:

Second

Lecturer:

Asst. lect. Mustafa Ameer Awadh

Lecture: (4pr)

Console Application in Visual Basic (.NET) for DES Key Generation

This **Visual Basic .NET** console application **generates a DES key** and prints its binary and hexadecimal representations.

📌 Features of the Program

✅ Generates a **56-bit DES key**
✅ Expands the key to **64 bits** by adding parity bits
✅ Prints the **binary and hexadecimal** representations of the key
✅ Uses **System.Security. Cryptography** for randomness

```
Imports System
Imports System.Security.Cryptography

Module DESKeyGenerator
Sub Main()
        ' Generate a random 56-bit key
Dim keyBytes As Byte() = GenerateRandom56BitKey()

        ' Expand to 64 bits by adding parity bits
Dim finalKey As Byte() = AddParityBits(keyBytes)

        ' Print the key in binary and hexadecimal format
                Console.WriteLine("Generated DES Key:")
        Console.WriteLine("Hex: " &
BitConverter.ToString(finalKey).Replace("-", ""))
        Console.WriteLine("Binary: " & ByteArrayToBinary(finalKey))

        Console.WriteLine(vbCrLf & "Press any key to exit...")
        Console.ReadKey()
End Sub

    ' Function to generate a 56-bit random key (7 bytes)
Function GenerateRandom56BitKey() As Byte()
Dim keyBytes(6) As Byte ' 7 bytes = 56 bits
Using rng As New RNGCryptoServiceProvider()
            rng.GetBytes(keyBytes)
End Using
Return keyBytes
End Function

    ' Function to add parity bits to form a 64-bit key
Function AddParityBits(ByVal keyBytes As Byte()) As Byte()
Dim finalKey(7) As Byte ' 8 bytes = 64 bits

For i As Integer = 0 To 6
Dim byteWithParity As Byte = keyBytes(i)
```

```
Dim parityBit As Byte = ComputeParity(byteWithParity)

            ' Place the 7-bit key and parity bit into 8-bit format
finalKey(i) = (byteWithParity And &HFE) Or parityBit
Next

        ' Last byte is generated similarly for full 64-bit key
Dim lastParity As Byte = ComputeParity(finalKey(0) Xor finalKey(1) Xor
finalKey(2) Xor finalKey(3) Xor finalKey(4) Xor finalKey(5) Xor finalKey(6))
finalKey(7) = lastParity

Return finalKey
End Function

    ' Function to compute odd parity for a byte
Function ComputeParity(ByVal value As Byte) As Byte
Dim count As Integer = 0
For i As Integer = 0 To 6 ' Only 7 bits
If ((value >> i) And 1) = 1 Then
count += 1
End If
Next
Return If((count Mod 2) = 0, 1, 0) ' Ensure odd parity
End Function

    ' Function to convert a byte array to a binary string
Function ByteArrayToBinary(ByVal byteArray As Byte()) As String
Dim binaryStr As String = ""
For Each b As Byte In byteArray
binaryStr &= Convert.ToString(b, 2).PadLeft(8, "0"c) & " "
Next
Return binaryStr.Trim()
End Function

End Module
```

📌 How It Works

1. **Generates a 56-bit random key** using `RNGCryptoServiceProvider()`.
2. **Expands it to 64 bits** by adding a **parity bit** to each byte.
3. **Prints the key** in **hexadecimal and binary format**.

Generated
DES Key: Hex: A5B3C9D8E7104F29 Binary: 10100101 10110011 1100100
1 11011000 11100111 0001000001001111 00101001

Press any key to exit...