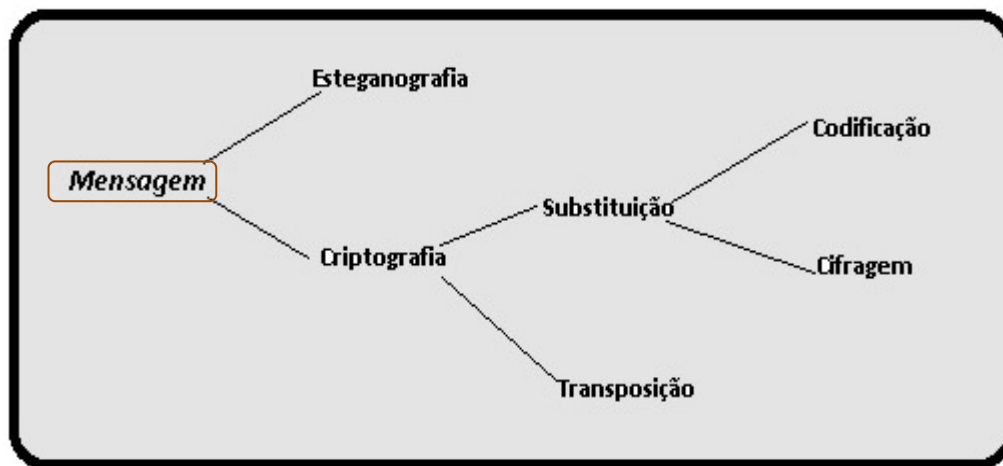


...

Conforme é do conhecimento de todos da área de TI, a proteção da informação é fundamental nos dias de hoje. Existem vários tipos de proteção: desde uma simples mudança de posição nas letras de uma mensagem até a encriptação com caracteres ilegíveis para os não autorizados a conhecer o conteúdo real da mensagem. De um modo mais abrangente, os métodos de proteção de uma mensagem podem ser classificados em “Estenografia” e “Criptografia”; veja a **figura 1**.



**Figura 1 - Métodos de proteção de uma mensagem**

Na literatura sobre o assunto existem muitos algoritmos e métodos mostrando como encriptar textos de modo quase impossível de serem decryptados; como o RSA que usa grandes números primos, tornando o método bastante seguro, mas o algoritmo bastante difícil de compreender.

Por outro lado, é possível desenvolver e escrever algoritmos bem simples que possam proteger as mensagens de maneira bem segura. O programa modular “**CriptoMorseBinario**”, codificado em Visualg, é uma solução simples para proteger uma mensagem; ele recebe o texto, converte-o em código Morse e finalmente o codifica em Binário. Neste caso, embora seja uma solução bem simples, a codificação/decodificação da mensagem é bastante sofisticada; o que o torna num programa bem eficiente. A **figura 2** mostra uma entrada/saída na codificação de uma mensagem digitada pelo usuário.

```

Digite a mensagem [máx. 1000 caracteres]: O alvo foi avistado

Mensagem em Binário
00000000000000000000000000000000 11111111000000000111111100000000
11111111111111111111111111111111 100000000000000000000000000000
00000000 111111111111110000000001111111000000000000000000000
0000000011111111111111 11111110000000001111111111111111
11111110000000011111111111111111111111111111111111100000000
111111100000000000000001111111111111111000000000000000000000

Mensagem original
O ALVO FOI AVISTADO

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg. |
```

**Figura 2 - Um exemplo de saída do programa “CriptoMorseBinario”**

### Algoritmo "CriptoMorseBinario"

```
//Gera mensagem criptografada em binário, a partir da codificação em Código Morse.
//Autor : Mário Leite
//E-mail : marleite@gmail.com
//-----
//Variáveis globais
Var MatMorse: vetor[1..100,1..4] de caractere
    VetMsg, VetMorse: vetor[1..100] de caractere
    VetBin: vetor[1..10000] de caractere
    i, TamEleVet, TamMsg: inteiro
    nada, Msg: caractere
    Vale: logico
//-----
Funcao FunVerifCarac(car: caractere): logico
//Verifica se um caractere da mensagem original é valido (letra ou espaço)
var Resp, Cond1, Cond2, Cond3: logico
Inicio
    Cond1 <- Asc(car)>=65
    Cond2 <- Asc(car)<=90
    Cond3 <- Asc(car)=32
    Se ((Cond1 e Cond2) ou Cond3) Entao
        Resp <- Verdadeiro
    Senao
        Resp <- Falso
    FimSe
    Retorne Resp
FimFuncao //fim da função "FunVerifCarac"
//-----
Procedimento ProCriaVetMorse(Msg: caractere; TamMsg: inteiro)
//Cria um vetor composto por elementos do Código Morse
var i: inteiro
Inicio
    Para i De 1 Ate TamMsg Faca //varre toda a mensagem original
        Escolha VetMsg[i]
        Caso "A"
            TamEleVet <- 2
            MatMorse[i,1] <- "."
            MatMorse[i,2] <- "-"
            VetMorse[i] <- "-."
        Caso "B"
            TamEleVet <- 4
            MatMorse[i,1] <- "-"
            MatMorse[i,2] <- "."
            MatMorse[i,3] <- "."
            MatMorse[i,4] <- "."
            VetMorse[i] <- "-..."
        Caso "C"
            TamEleVet <- 4
            MatMorse[i,1] <- "-"
            MatMorse[i,2] <- "."
            MatMorse[i,3] <- "-"
            MatMorse[i,4] <- "."
            VetMorse[i] <- "-.-."
        Caso "D"
            TamEleVet <- 3
            MatMorse[i,1] <- "-"
            MatMorse[i,2] <- "."
            MatMorse[i,3] <- "."
            VetMorse[i] <- "-.."
        Caso "E"
            TamEleVet <- 1
            MatMorse[i,1] <- "."
```

```

VetMorse[i] <- "."
Caso "F"
TamEleVet <- 4
MatMorse[i,1] <- "."
MatMorse[i,2] <- "."
MatMorse[i,3] <- "-"
MatMorse[i,4] <- "."
VetMorse[i] <- "..-."
Caso "G"
TamEleVet <- 3
MatMorse[i,1] <- "-"
MatMorse[i,2] <- "-"
MatMorse[i,3] <- "."
VetMorse[i] <- "--."
Caso "H"
TamEleVet <- 4
MatMorse[i,1] <- "."
MatMorse[i,2] <- "."
MatMorse[i,3] <- "."
MatMorse[i,4] <- "."
VetMorse[i] <- "...."
Caso "I"
TamEleVet <- 2
MatMorse[i,1] <- "."
MatMorse[i,2] <- "."
VetMorse[i] <- ".."
Caso "J"
TamEleVet <- 4
MatMorse[i,1] <- "."
MatMorse[i,2] <- "-"
MatMorse[i,3] <- "-"
MatMorse[i,4] <- "-"
VetMorse[i] <- ".---"
Caso "K"
TamEleVet <- 3
MatMorse[i,1] <- "-"
MatMorse[i,2] <- "."
MatMorse[i,3] <- "-"
VetMorse[i] <- "-.-"
Caso "L"
TamEleVet <- 4
MatMorse[i,1] <- "."
MatMorse[i,2] <- "-"
MatMorse[i,3] <- "."
MatMorse[i,4] <- "."
VetMorse[i] <- ".-.."
Caso "M"
TamEleVet <- 2
MatMorse[i,1] <- "-"
MatMorse[i,2] <- "-"
VetMorse[i] <- "--"
Caso "N"
TamEleVet <- 2
MatMorse[i,1] <- "-"
MatMorse[i,2] <- "."
VetMorse[i] <- "-."
Caso "O"
TamEleVet <- 3
MatMorse[i,1] <- "-"
MatMorse[i,2] <- "-"
MatMorse[i,3] <- "-"
VetMorse[i] <- "---"
Caso "P"
TamEleVet <- 4
MatMorse[i,1] <- "."
MatMorse[i,2] <- "-"
MatMorse[i,3] <- "-"

```

```

MatMorse[i,4] <- "."
VetMorse[i] <- ".--."
Caso "Q"
TamEleVet <- 4
MatMorse[i,1] <- "-"
MatMorse[i,2] <- "-"
MatMorse[i,3] <- "."
MatMorse[i,4] <- "-"
VetMorse[i] <- "--.-"
Caso "R"
TamEleVet <- 3
MatMorse[i,1] <- "."
MatMorse[i,2] <- "-"
MatMorse[i,3] <- "."
VetMorse[i] <- "-.-"
Caso "S"
TamEleVet <- 3
MatMorse[i,1] <- "."
MatMorse[i,2] <- "."
MatMorse[i,3] <- "."
VetMorse[i] <- "...
Caso "T"
TamEleVet <- 1
MatMorse[i,1] <- "-"
VetMorse[i] <- "-"
Caso "U"
TamEleVet <- 3
MatMorse[i,1] <- "."
MatMorse[i,2] <- "."
MatMorse[i,3] <- "-"
VetMorse[i] <- "...-
Caso "V"
TamEleVet <- 4
MatMorse[i,1] <- "."
MatMorse[i,2] <- "."
MatMorse[i,3] <- "."
MatMorse[i,4] <- "-"
VetMorse[i] <- "...-
Caso "W"
TamEleVet <- 3
MatMorse[i,1] <- "."
MatMorse[i,2] <- "-"
MatMorse[i,3] <- "-"
VetMorse[i] <- ".--"
Caso "X"
TamEleVet <- 4
MatMorse[i,1] <- "-"
MatMorse[i,2] <- "."
MatMorse[i,3] <- "."
MatMorse[i,4] <- "-"
VetMorse[i] <- "-.-"
Caso "Y"
TamEleVet <- 4
MatMorse[i,1] <- "-"
MatMorse[i,2] <- "."
MatMorse[i,3] <- "-"
MatMorse[i,4] <- "-"
VetMorse[i] <- "-.-"
Caso "Z"
TamEleVet <- 4
MatMorse[i,1] <- "-"
MatMorse[i,2] <- "-"
MatMorse[i,3] <- "."
MatMorse[i,4] <- "."
VetMorse[i] <- "--.."

```

```

        Caso " ", " ", " ", " " //caso de espaços
        TamEleVet <- 1
        VetMorse[i] <- " "

    FimEscolha
    FimPara //fim da varredura da mensagem original
    FimProcedimento //fim do procedimento "ProCriaVetMorse"
//-----
Procedimento ProCriaEscreveVetBinario (Msg: caractere; TamMsg: inteiro)
//Cria e exibe o vetor de binário
    var j, k, n, Col: inteiro
    Inicio
        Escreval ("Mensagem em Binário")
        Col <- 0 //contador de blocos de bytes
        Para j De 1 Ate TamMsg Faca //varre toda a mensagem original
            Escolha VetMsg[j]
                Caso "A"
                    TamEleVet <- 2
                    MatMorse[j,1] <- "."
                    MatMorse[j,2] <- "-"
                    VetMorse[j] <- "-."
                Caso "B"
                    TamEleVet <- 4
                    MatMorse[j,1] <- "-"
                    MatMorse[j,2] <- "."
                    MatMorse[j,3] <- "."
                    MatMorse[j,4] <- "."
                    VetMorse[j] <- "-..."
                Caso "C"
                    TamEleVet <- 4
                    MatMorse[j,1] <- "-"
                    MatMorse[j,2] <- "."
                    MatMorse[j,3] <- "-"
                    MatMorse[j,4] <- "."
                    VetMorse[j] <- "-.-."
                Caso "D"
                    TamEleVet <- 3
                    MatMorse[j,1] <- "-"
                    MatMorse[j,2] <- "."
                    MatMorse[j,3] <- "."
                    VetMorse[j] <- "-.."
                Caso "E"
                    TamEleVet <- 1
                    MatMorse[j,1] <- "."
                    VetMorse[j] <- "."
                Caso "F"
                    TamEleVet <- 4
                    MatMorse[j,1] <- "."
                    MatMorse[j,2] <- "-"
                    MatMorse[j,3] <- "-"
                    MatMorse[j,4] <- "."
                    VetMorse[j] <- "...-"
                Caso "G"
                    TamEleVet <- 3
                    MatMorse[j,1] <- "-"
                    MatMorse[j,2] <- "-"
                    MatMorse[j,3] <- "."
                    VetMorse[j] <- "--."
                Caso "H"
                    TamEleVet <- 4
                    MatMorse[j,1] <- "."
                    MatMorse[j,2] <- "."
                    MatMorse[j,3] <- "."
                    MatMorse[j,4] <- "."
                    VetMorse[j] <- "...."
            FimEscolha
        FimPara
    FimProcedimento

```

```

Caso "I"
  TamEleVet <- 2
  MatMorse[j,1] <- "."
  MatMorse[j,2] <- "."
  VetMorse[j] <- ".."

Caso "J"
  TamEleVet <- 4
  MatMorse[j,1] <- "."
  MatMorse[j,2] <- "-"
  MatMorse[j,3] <- "-"
  MatMorse[j,4] <- "-"
  VetMorse[j] <- ".----"

Caso "K"
  TamEleVet <- 3
  MatMorse[j,1] <- "-"
  MatMorse[j,2] <- "."
  MatMorse[j,3] <- "-"
  VetMorse[j] <- "-.-"

Caso "L"
  TamEleVet <- 4
  MatMorse[j,1] <- "."
  MatMorse[j,2] <- "-"
  MatMorse[j,3] <- "."
  MatMorse[j,4] <- "."
  VetMorse[j] <- ".-..."

Caso "M"
  TamEleVet <- 2
  MatMorse[j,1] <- "-"
  MatMorse[j,2] <- "-"
  VetMorse[j] <- "--"

Caso "N"
  TamEleVet <- 2
  MatMorse[j,1] <- "-"
  MatMorse[j,2] <- "."
  VetMorse[j] <- "-."

Caso "O"
  TamEleVet <- 3
  MatMorse[j,1] <- "-"
  MatMorse[j,2] <- "-"
  MatMorse[j,3] <- "-"
  VetMorse[j] <- "---"

Caso "P"
  TamEleVet <- 4
  MatMorse[j,1] <- "."
  MatMorse[j,2] <- "-"
  MatMorse[j,3] <- "-"
  MatMorse[j,4] <- "."
  VetMorse[j] <- ".-.-"

Caso "Q"
  TamEleVet <- 4
  MatMorse[j,1] <- "-"
  MatMorse[j,2] <- "-"
  MatMorse[j,3] <- "."
  MatMorse[j,4] <- "-"
  VetMorse[j] <- "--.-"

Caso "R"
  TamEleVet <- 3
  MatMorse[j,1] <- "."
  MatMorse[j,2] <- "-"
  MatMorse[j,3] <- "."
  VetMorse[j] <- ".-."

Caso "S"
  TamEleVet <- 3
  MatMorse[j,1] <- "."
  MatMorse[j,2] <- "."
  MatMorse[j,3] <- "."
  VetMorse[j] <- "...

```

```

Caso "T"
    TamEleVet <- 1
    MatMorse[j,1] <- "-"
    VetMorse[j] <- "-"
Caso "U"
    TamEleVet <- 3
    MatMorse[j,1] <- "."
    MatMorse[j,2] <- "."
    MatMorse[j,3] <- "-"
    VetMorse[j] <- "..-"
Caso "V"
    TamEleVet <- 4
    MatMorse[j,1] <- "."
    MatMorse[j,2] <- "."
    MatMorse[j,3] <- "."
    MatMorse[j,4] <- "-"
    VetMorse[j] <- "...-"
Caso "W"
    TamEleVet <- 3
    MatMorse[j,1] <- "."
    MatMorse[j,2] <- "-"
    MatMorse[j,3] <- "-"
    VetMorse[j] <- ".--"
Caso "X"
    TamEleVet <- 4
    MatMorse[j,1] <- "-"
    MatMorse[j,2] <- "."
    MatMorse[j,3] <- "."
    MatMorse[j,4] <- "-"
    VetMorse[j] <- "-..-"
Caso "Y"
    TamEleVet <- 4
    MatMorse[j,1] <- "-"
    MatMorse[j,2] <- "."
    MatMorse[j,3] <- "-"
    MatMorse[j,4] <- "-"
    VetMorse[j] <- "-.-"
Caso "Z"
    TamEleVet <- 4
    MatMorse[j,1] <- "-"
    MatMorse[j,2] <- "-"
    MatMorse[j,3] <- "."
    MatMorse[j,4] <- "."
    VetMorse[j] <- "--.."
Caso " ", " ", " ", " " //caso de espaços
    TamEleVet <- 1
    VetMorse[j] <- " "
FimEscolha
n <- 0 //contador (índice) do vetor de binários
Para k De 1 Ate TamEleVet Faca //varre o caractere em Morse
    n <- n + 1
    Col <- Col + 1
    Se (VetMorse[j]=" ") Entao
        VetBin[n] <- " "
    Senao
        Se (MatMorse[j,k]=".") Entao
            VetBin[n] <- "11111111"
        FimSe
        Se (MatMorse[j,k]="-") Entao
            VetBin[n] <- "00000000"
        FimSe
    FimSe
Se (Col<=8) Entao //limita o número de bytes por linha
    Escreva (VetBin[n])
Senao
    Escreval ("")
    Escreva (VetBin[n])

```

```

        Col <- 1
        FimSe
        FimPara //fim da varredura do caractere em Morse
    FimPara //fim da varredura da mensagem original
    Escreval("")
    FimProcedimento //fim do procedimento "ProCriaEscreveVetBinario"
//=====
//Programa principal
Inicio
//Leitura da mensagem original
Vale <- Falso
Enquanto (Vale=Falso) Faca
    LimpaTela
    Repita
        Escreva("Digite a mensagem [máx. 1000 caracteres]: ")
        Leia(Msg)
        Msg <- Maiusc(Msg) //converte a mensagem original em maiúscula
        TamMsg <- Compr(Msg)
        Ate(TamMsg<=1000)
        Escreval("")
        {Cria um vetor de caracteres da mensagem original}
        Para i De 1 Ate TamMsg Faca
            VetMsg[i] <- Copia(Msg,i,1)
            Vale <- FunVerifCarac(VetMsg[i])
            Se(Nao(Vale)) Entao
                Escreval("Caractere inválido. Digite a mensagem novamente.")
                Leia(nada)
                Interrompa //abandona o loop incondicionalmente
            Senao
                Vale <- Verdadeiro
        FimSe
    FimPara
FimEnquanto
ProCriaVetMorse(Msg,TamMsg) //chama procedimento para criar um vetor em Código Morse
ProCriaEscreveVetBinario(Msg,TamMsg) //chama rotina para criar/escrever em binário
Escreval("")
Escreval("Mensagem original")
Escreval(Msg) //exibe a mensagem originalmente digitada
Escreval("")
FimAlgoritmo //fim do programa "MorseBinario"

```