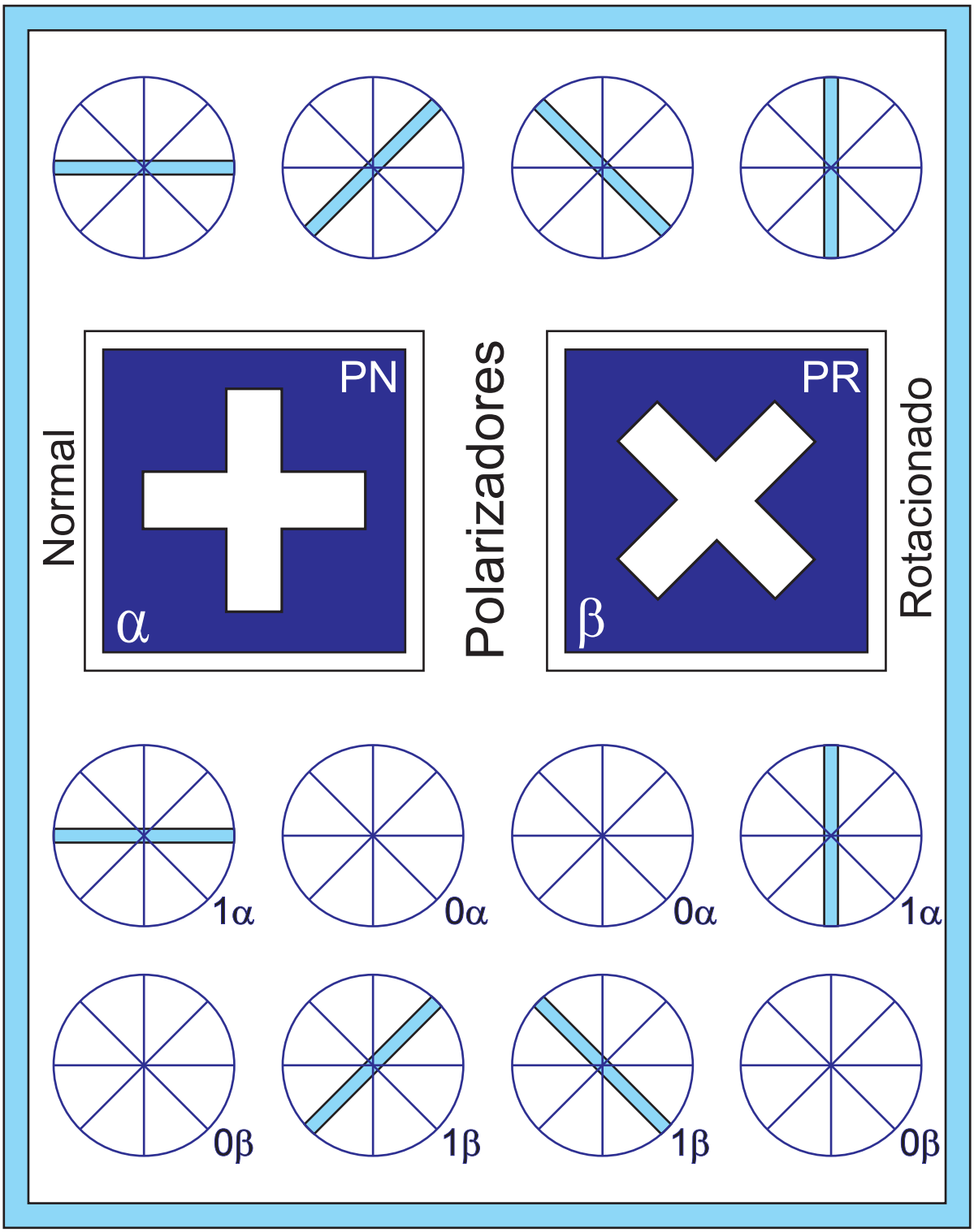
CRIPTOGRAFIA QUÂNTICA

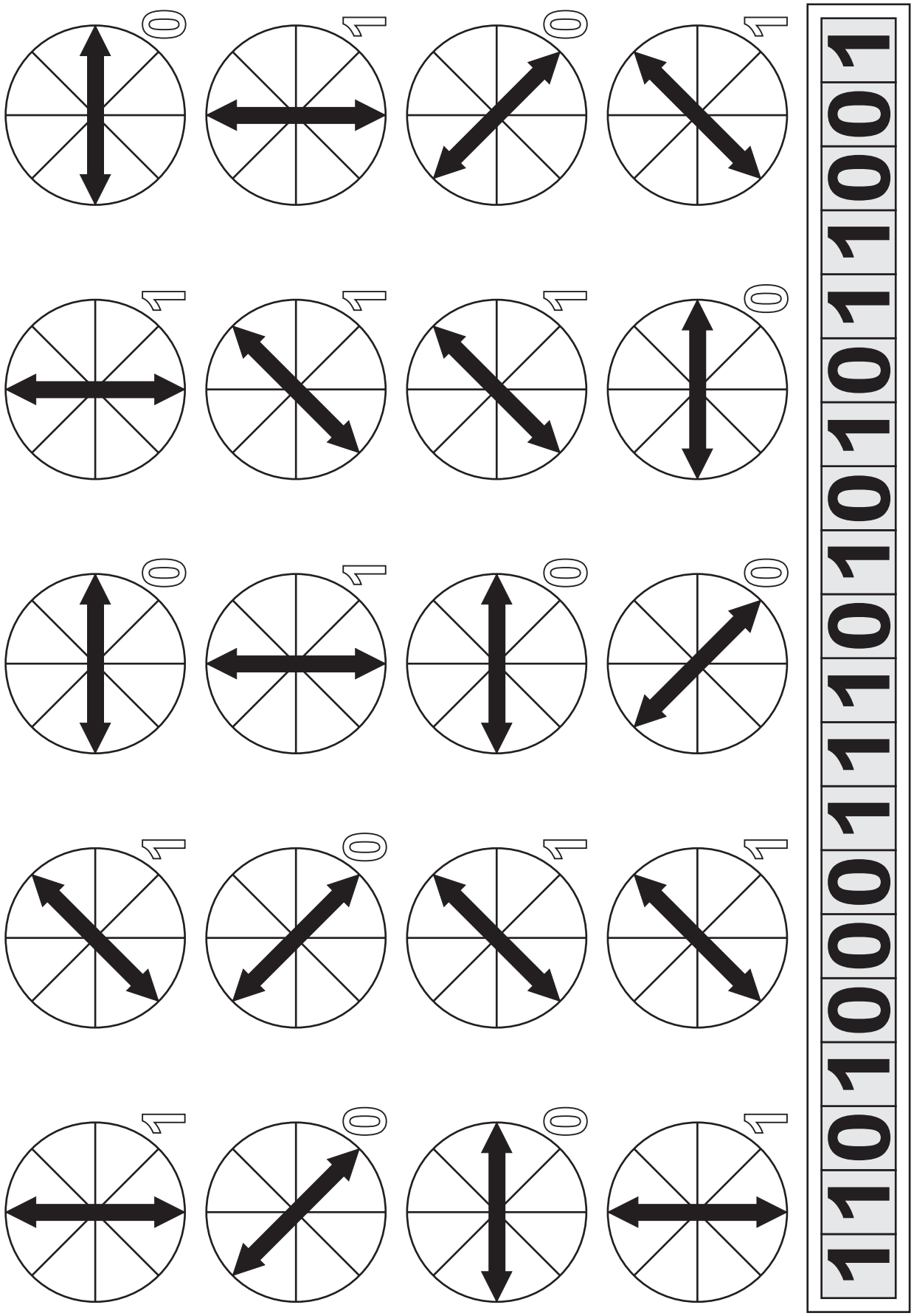


Protocolo BB84

Quantum Key Distribution

Copyright © 2019 Prandiano
Reprodução proibida: Lei 9.610 de 19/02/1998.

4.2 Chave como Sequência de Bits Aleatórios



Copyright © 2019 Prandiano

Reprodução proibida: Lei 9.610 de 19/02/1998.

Copyright © 2019 Prandiano

5.0 Criptografia Bitwise da Lógica XOR

Documento que Será Enviado

Base Decimal
575377
Base Hexadecimal
8c791
Base Binária
10001100011110010001

p

X

Chave Secreta Quântica

Base Decimal
765851
Base Hexadecimal
baf9b
Base Binária
10111010111110011011

q

Criptografia (ADA)

$\wedge$

0	0	0
0	1	0
1	0	0
1	1	1

$\oplus$

10001100011110010001 : Doc.
10111010111110011011 : Chav.

00110110100000001010 : Doc.crip.

$\vee$

0	0	0
0	1	1
1	0	1
1	1	1

Descriptografia (BIA)

$\oplus$

00110110100000001010 : Doc.crip.
10111010111110011011 : Chav.

10001100011110010001 : Doc.

$$p \oplus q = [\![p \wedge (\neg q)]\!] \vee [\![(\neg p) \wedge q]\!]$$

Reprodução proibida: Lei 9.610 de 19/02/1998.

5.1 Solução da Equação da Lógica XOR ⊕

$$\neg$$

0	1
1	0

$$\wedge$$

0	0	0
0	1	0
1	0	0
1	1	1

$$\vee$$

0	0	0
0	1	1
1	0	1
1	1	1

$p = 154 = 9a = 10011010$ | $q = 233 = e9 = 11101001$

$$p \oplus q = [p \wedge (\neg q)] \vee [(\neg p) \wedge q]$$

$$p \oplus q = [p \wedge (\neg q)] \vee [(\neg p) \wedge q]$$

$$p \oplus q = [[10011010 \wedge 00010110] \vee [01100101 \wedge 11101111]]$$

$$p \oplus q = [[00010010] \vee [01100001]]$$

$$p \oplus q = 01110011$$

$\oplus$

0	0	0
0	1	1
1	0	1
1	1	0

$p \oplus q$

$p = 10011010$ $q = 11101001$

01110011

Copyright © 2019 Prandiano

Reprodução proibida: Lei 9.610 de 19/02/1998.

6.4 Código VBA para Executar o Procedimento de Transformação da Mensagem de Entrada para a Forma Binária

```
Function Mensagem_Transformada_Bits(Senha As String) As String


    Dim num_caracts_senha As Integer
    Dim num_bits_senha As Long
    Dim k As Integer
    Dim num_bits_senha_binario As String
    Dim Num_Blocos As Long


    num_caracts_senha = Len(Senha)
    num_bits_senha = 8 * num_caracts_senha


    Dim lenght As Double
    lenght = Int(num_caracts_senha / 4 + 2)
    If lenght Mod 16 = 0 Then
       Num_Blocos = WorksheetFunction.RoundUp(lenght / 16, 0) + 1
    Else
       Num_Blocos = WorksheetFunction.RoundUp(lenght / 16, 0)
    End If


    num_bits_senha_binario = Decimal_para_Binario(num_bits_senha)


    Mensagem_Transformada_Bits = ""
    Dim contador As Integer
    Dim IndiceProcurado As Integer
    Dim BinarioProcurado As String
    For contador = 1 To num_caracts_senha
       Dim LetraProcurada As String
       LetraProcurada = Chr(34) & Mid(Senha, contador, 1) & Chr(34)
```

Bloco A

Copyright © 2019 Prandiano
Reprodução proibida: Lei 9.610 de 19/02/1998.

7.1.2 Código VBA para Executar o Procedimento de Gerar as Words

Algoritmo

```
Public Function GerarWords(Mensagem_em_Bits As String,
NumDeBlocos AsInteger) As Variant
    Dim W(0 To 1000, 0 To 63) As String
    Dim W_Hexa(0 To 1000, 0 To 63) As String
    Dim z As Integer
    c = 0
    Dim i As Integer
    Dim j As Integer
    For i = 0 To NumDeBlocos - 1
        For j = 0 To 15
            W(i, j) = Mid(Mensagem_em_Bits, z * 32 + 1, 32)
            W_Hexa(i, j) = Binario_para_Hexadecimal(CStr(W(i, j)))
            c = c + 1
        Next j
    Next i
    Dim sigma_zero As String
    Dim sigma_um As String
    For i = 0 To NumDeBlocos - 1
        For j = 16 To 63
            sigma_um = sigma_um_Function(CStr(W_Hexa(i, j - 2)))
            sigma_zero = sigma_zero_Function(CStr(W_Hexa(i, j - 15)))
            W_Hexa(i, j) = W_Function(sigma_um, W_Hexa(i, j - 7), _
            sigma_zero, W_Hexa(i, j - 16))
        Next j
    Next i
    GerarWords = W_Hexa


End Function
```

Words

Copyright © 2019 Prandiano
Reprodução proibida: Lei 9.610 de 19/02/1998.