Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**«Отчёт по лабораторной работе №6»**

“Исследование потоковых шифров”

**Выполнил:** студент 4 курса

1 группы специальности ИСИт

Халалеенко Андрей Николаевич

**Проверил:** преподаватель

Сазонова Дарья Владимировна

Минск 2024

**Разработать авторские многооконные приложения в соответствии с целью лабораторной работы. При этом можно воспользоваться готовыми библиотеками либо программными кодами, реализующими заданные алгоритмы (BBS)**

Реализация приложения:

|  |
| --- |
| package main  import (  "fmt"  "fyne.io/fyne/v2"  "math/big"  "strconv"  "strings"  "fyne.io/fyne/v2/app"  "fyne.io/fyne/v2/container"  "fyne.io/fyne/v2/layout"  "fyne.io/fyne/v2/widget"  )  func main() {  a := app.New()  win := a.NewWindow("BBS Pseudo-Random Sequence Generator with RC4")  pEntry := widget.NewEntry()  qEntry := widget.NewEntry()  x0Entry := widget.NewEntry()  keyEntry := widget.NewEntry()  pLabel := widget.NewLabel("p:")  qLabel := widget.NewLabel("q:")  x0Label := widget.NewLabel("X0:")  keyLabel := widget.NewLabel("Key (comma separated decimal numbers):")  resultTextView := widget.NewMultiLineEntry()  resultTextView.MultiLine = true  generateButton := widget.NewButton("Generate Sequence", func() {  pStr := pEntry.Text  qStr := qEntry.Text  x0Str := x0Entry.Text  keyStr := keyEntry.Text  p, \_ := new(big.Int).SetString(pStr, 10)  q, \_ := new(big.Int).SetString(qStr, 10)  x0, \_ := new(big.Int).SetString(x0Str, 10)  n := new(big.Int).Mul(p, q)  if !p.ProbablyPrime(0) || !q.ProbablyPrime(0) {  resultTextView.SetText("p and q must be prime numbers")  return  }  key := make([]byte, 0)  for \_, numStr := range splitByComma(keyStr) {  num, \_ := strconv.Atoi(numStr)  key = append(key, byte(num))  }  seq := rc4(key, bbs(n, x0))  resultStr := ""  for \_, num := range seq {  resultStr += fmt.Sprintf("%d", num)  }  resultTextView.SetText(resultStr)  })  content := container.New(layout.NewVBoxLayout(),  container.New(layout.NewGridLayoutWithColumns(2),  pLabel,  pEntry,  qLabel,  qEntry,  x0Label,  x0Entry,  keyLabel,  keyEntry,  ),  generateButton,  resultTextView,  )  win.SetContent(content)  win.Resize(fyne.NewSize(300, 500))  win.ShowAndRun()  }  func splitByComma(s string) []string {  return strings.Split(strings.ReplaceAll(s, " ", ""), ",")  }  func bbs(n, x0 \*big.Int) []int {  var seq []int  for i := 0; i < 256; i++ {  y := new(big.Int).Exp(x0, big.NewInt(2), n)  bit := int(y.Bit(0))  seq = append(seq, bit)  x0 = y  }  return seq  }  func rc4(key []byte, seq []int) []int {  s := make([]int, 256)  for i := 0; i < 256; i++ {  s[i] = i  }  j := 0  for i := 0; i < 256; i++ {  j = (j + s[i] + int(key[i%len(key)])) % 256  s[i], s[j] = s[j], s[i]  }  i := 0  j = 0  result := make([]int, len(seq))  for n := 0; n < len(seq); n++ {  i = (i + 1) % 256  j = (j + s[i]) % 256  s[i], s[j] = s[j], s[i]  result[n] = seq[n] ^ s[(s[i]+s[j])%256]  }  return result  } |

Результат выполнения:

