Politechnika Białostocka

# Sprawozdanie z pracowni specjalistycznej

## *Bezpieczeństwo Sieci Komputerowych*

Temat: *IMPLEMENTACJA PODSTAWOWYCH MODUŁÓW KRYPTOGRAFICZNYCH*

Wykonujący ćwiczenie: **Kamil Karwowski**

Studia dzienne

Kierunek: Kierunek

Semestr: VI

Grupa zajęciowa: Grupa PS6

Prowadzący ćwiczenie: Mgr. Inż. Dariusz Jankowski

Data wykonania ćwiczenia:
30.05.2023r.

## Treść zadania 2a:

Zaimplementuj system kryptograficzny oparty o przestawienie macierzowe pokazane w przykładzie 2a dla d = 5 oraz klucza key = 3-4-1-5-2 (1 punkt).

## Testy:


## Kod:

Metoda na szyfrowanie:

```
private void button3_Click(object sender, RoutedEventArgs e)
{
    string szyfrowanko = null;
    int[] key = new[] { 3, 4, 1, 5, 2 };
    string slowo = textBox6.Text.ToString();

    char[] encodeslowo = new char[slowo.Length];
    int[] pointers = (int[])key.Clone();

    int mainPointer = 0;

    for (int i = 0; i < Math.Ceiling(slowo.Length / (float)key.Length); i++)
    {
        foreach (int pointer in pointers)
        {
            if (pointer <= slowo.Length)
            {
                encodeslowo[mainPointer++] = slowo[pointer - 1];
            }
        }
        for (int j = 0; j < pointers.Length; j++)
        {
            pointers[j] += pointers.Length;
        }


    }
    foreach (var item in encodeslowo)
    {
        szyfrowanko += item;
    }



    textBox6_Copy.Text = szyfrowanko;
}
```

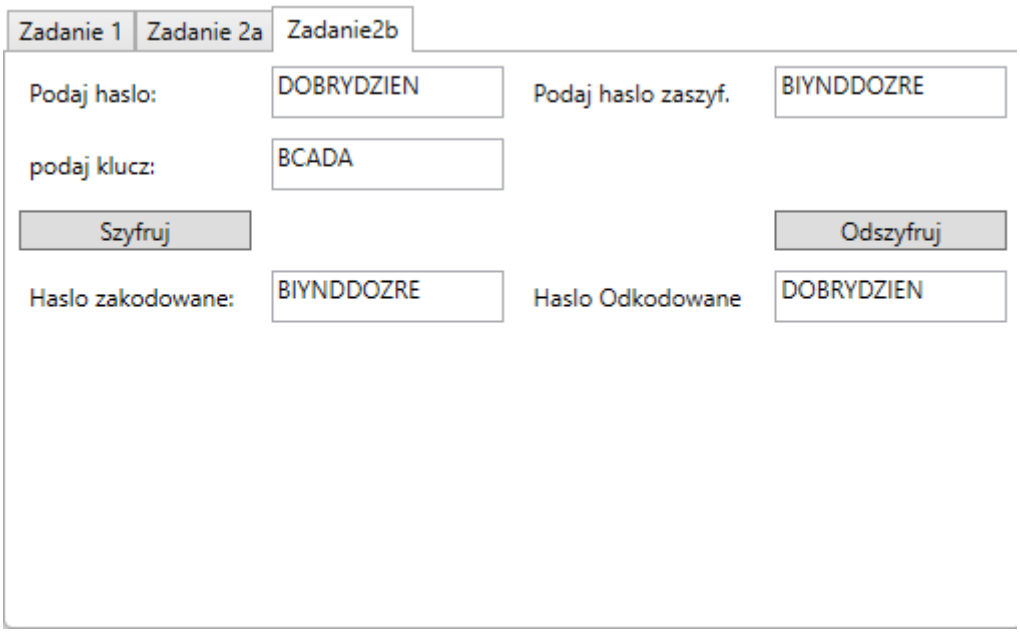
Odszyfrowanie:

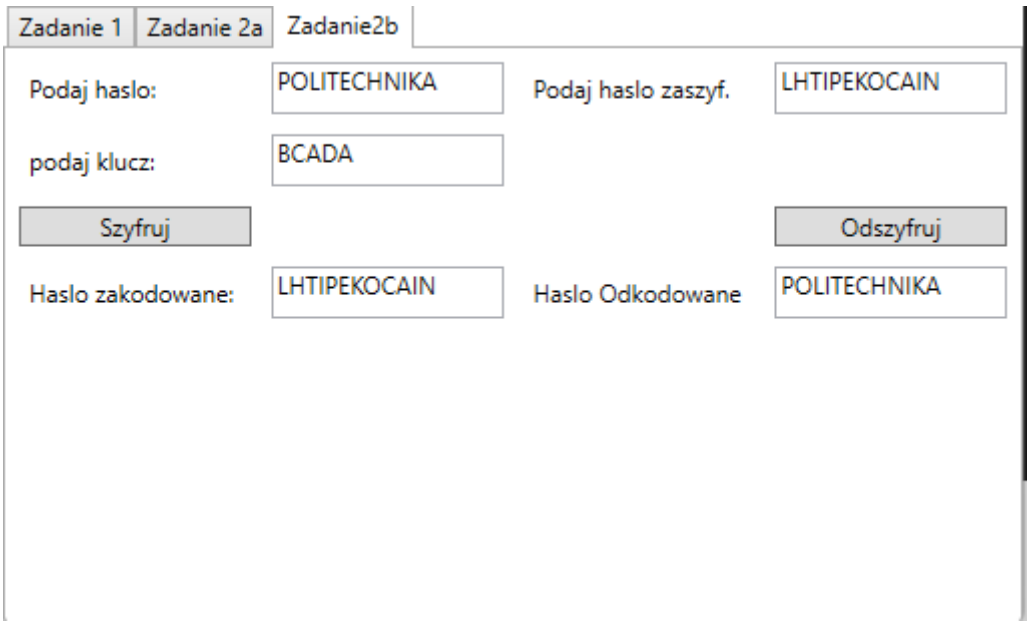
```
int[] key = new[] { 3, 4, 1, 5, 2 };
string slowo = textBox6_Copy1.Text.ToString();
char[] decodedWord = new char[slowo.Length];
string decodeslowo = null;
int[] pointers = (int[])key.Clone();

int mainPointer = 0;

for (int i = 0; i < Math.Ceiling(slowo.Length / (float)key.Length); i++)
{
    foreach (int pointer in pointers)
    {
        if (pointer <= slowo.Length)
        {
            decodedWord[pointer - 1] = slowo[mainPointer++];

        }

    }
    for (int j = 0; j < pointers.Length; j++)
    {
        pointers[j] += pointers.Length;
    }

}
foreach (var item in decodedWord)
{
    decodeslowo += item;
}



textBox6_Copy2.Text = decodeslowo;
```

## Treść zadania 2b:

Zaimplementuj system kryptograficzny oparty o przestawienie macierzowe pokazane w przykładzie 2a dla d = 5 oraz klucza key = 3-4-1-5-2 (1 punkt).

## Testy:



## Kod:

Szyfrowanie:

```
string keyString = text2bklucz.Text;
string text = text2bhaslo.Text;
var key = PrepareKey(keyString);


var map = CalculateMap(key);

StringBuilder builder = new StringBuilder(text.Length);

for (int i = map.MinKey; i <= map.MaxKey; ++i)
    for (int textIndex = map[i]; textIndex < text.Length; textIndex += key.Length)
        builder.Append(text[textIndex]); //wpisujemy odpowiednie wartości do string


text2bzaszyfrowane.Text = builder.ToString();
```

Odszyfrowywanie:

```
private void Odszyfrowanie2b_Click(object sender, RoutedEventArgs e)
{
    string keyString = text2bklucz.Text;
    string text = text2bhaslodoodszyfrowania.Text;
    var key = PrepareKey(keyString);


    var rows = text.Length / key.Length; //ilość pełnych wierszy
    var remainder = text.Length % key.Length;  //ilość liter w ostatnim, niepełnym wierszu
    var jaggedMatrix = CreateJaggedMatrix(key, rows, remainder);
    var map = CalculateMap(key);
    int textI = 0;

    for (int i = map.MinKey; i <= map.MaxKey; ++i)...


    StringBuilder builder = new StringBuilder(text.Length);

    for (int y = 0; y < rows; ++y)
        for (int x = 0; x < key.Length; ++x) // pętla dla pełnych wierszy
            builder.Append(jaggedMatrix[x][y]);

    for (int x = 0; x < remainder; ++x) // pętla dla ostatniego, niepełnego wiersza (o ile istnieje)
        builder.Append(jaggedMatrix[x][rows]);

    text2bpoodszyfrowaniu.Text = builder.ToString();
}
```