Міністерство освіти та науки України

Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

Звіт

До практичної роботи №2

З дисципліни: «Безпека програм та даних»

Виконав:

ст. гр. ПЗПІ-19-3

Селевич О.В.

Перевірив:

асистент кафедри ПІ

Олійник О.О.

Харків 2022

Тема роботи: Знайомство з шифром “Одноразовий блокнот”.

Мета роботи: Ознайомити студентів з шифром “Одноразовий блокнот”, відпрацювати навички використовування цього шифру для кодування та декодування тексту.

Хід роботи:

Під час практичного заняття була створена програмні реалізації шифру “Одноразовий блокнот”за допомогою мови програмування JavaScript.
Користувацький інтерфейс для демострації роботи шифру був створений за допогою JavaScript-фреймворку Vue.js.

1) **Створення шифру “Одноразовий блокнот” (One-Time-Pad)**

Перед тим як винати кодування за допомогою шифру “Одноразовий блокнот” потрібно перетворити вихідний текст в цифри. Створений програмний застосунок для демонстрації роботи алгоритму підтримує дві мови: російську та англійську. Тому для виконання перетворення вихідного тексту в цифри було створено дві таблиці стиснення, що знаходяться у директорії jsonStorage та мають назви englishStraddlingCheckerboard.json та russianStraddlingCheckerboard.json. Вони зберігаються у форматі json. Вони містять по 37 символів доступних для стиснення. В більшості цими символами є літери відповідних алфавітів, але вони містят ще й додаткові символи, що можуть бути стиснені, такі, як розділові знаки та додаткові символи. Гарним прикладом додаткового символу є “FIG”, що є необхідним для стиснення чисел, та “#”, котрий є будь-яким іншим символом, котрий не є у таблиці стиснення. Вміст файлу russianStraddlingCheckerboard.json (таблиця стиснення російського алфавіту):

```
1. {
```

```
2.     "1":"а",
3.     "2":"и",
4.     "3":"т",
5.     "4":"е",
6.     "5":"с",
7.     "6":"н",
8.     "7":"о",
9.     "8":"",
10.      "9":"",
11.      "0":"",
12.      "81":"б",
13.      "82":"в",
14.      "83":"г",
15.      "84":"д",
16.      "85":"ж",
17.      "86":"з",
18.      "87":"к",
19.      "88":"л",
20.      "89":"м",
21.      "80":"п",
22.      "91":"р",
23.      "92":"у",
24.      "93":"ф",
25.      "94":"х",
26.      "95":"ц",
27.      "96":"ч",
28.      "97":"ш",
29.      "98":"щ",
30.      "99":"ъ",
31.      "90":"ы",
32.      "01":"ь",
33.      "02":"э",
34.      "03":"ю",
35.      "04":"я",
36.      "05":".",
37.      "06":"/",
38.      "07":"FIG",
39.      "08":"й",
40.      "09":"#",
41.      "00":" "
42.  }
```

Вміст файлу englishStraddlingCheckerboard.json (таблиця стиснення англійського алфавіту):

```
{
   "1":"e",
   "2":"s",
   "3":"t",
   "4":"o",
   "5":"n",
   "6":"i",
   "7":"a",
   "8":"",
     "9":"",
     "0":"",
     "81":"b",
     "82":"c",
     "83":"d",
     "84":"f",
     "85":"g",
     "86":"h",
     "87":"j",
     "88":"k",
     "89":"l",
     "80":"m",
     "91":"p",
     "92":"q",
     "93":"r",
     "94":"u",
     "95":"v",
     "96":"w",
     "97":"x",
     "98":"y",
     "99":"z",
     "90":".",
     "01":"/",
     "02":"FIG",
     "03":",",
     "04":"?",
     "05":"-",
     "06":":",
     "07":"|",
     "08":"!",
     "09":"#",
     "00":" "
 }
```

Головними функціями для виконання кодування/декодування за допомогою шифру “Одноразовий блокнот” є encryptDecimal та decryptDecimal, що знаходяться у файлі oneTimePad.js. Програмний код вищезазначених функцій:

```
export const encryptDecimal = (key, plainText, language) => {
  const straddlingTable =
getStraddingCheckerboardFromJson(language);
  const plainCode = convertPlainTextToPlainCode(plainText,
straddlingTable);
  const constructedData = constructKey(key, plainCode,
straddlingTable);

  const encryptedData  =
  performDecimalOperation(constructedData.plainCode,
  constructedData.contructedKey, decimalAddition);
    return encryptedData;
  };

  export const decryptDecimal = (key, cipherText, language) => {
    const straddlingTable =
  getStraddingCheckerboardFromJson(language);
    const constructedData = constructKey(key, cipherText,
  straddlingTable);

    const decryptedSequence = performDecimalOperation(
      constructedData.plainCode,
      constructedData.contructedKey,
      decimalSubstraction
    );


    const plainText =
  convertPlainCodeToPlainText(decryptedSequence,
  straddlingTable);
    return plainText;
  };
```

Ці функції приймають на вхід три параметри: ключ, мова повідомлення та шифр/вихідне повідомлення. Спочатку у обох функціях виконується виклик функції getStraddlingCheckerboardFromJson, що повертає таблицю стиснення згідно з параметром мови, що була передана у функцію. Код функції getStraddlingCheckerboardFromJson:

```
const getStraddingCheckerboardFromJson = (language) => {
  const straddlingMap = new Map();
  let json = "";

  switch (language) {
```

```
    case "Ru":
      json = russianStraddlingCheckerboard;
      break;
    case "En":
        json = englishStraddlingCheckerboard;
        break;
    }

    for (let number in json) {
      straddlingMap.set(number, json[number]);
    }

    return straddlingMap;
  };
```

Далі були реалізовані функції convertPlainCodeToPlainText та convertPlainTextToPlainCode для виконання преобразування і зворотнього преобразування вихідного тексту у числовий вираз. Програмний код вищезазначених функцій:

```
const convertPlainCodeToPlainText = (plainCode, straddlingTable)
=> {
  const numerics = plainCode.split("");
  let plainText = "";
  let numberNoteWasUsed = false;

  for (let i = 0; i < numerics.length; i++) {
    if (numerics[i] === "0" && i + 1 === numerics.length) break;
    else if (straddlingTable.has(numerics[i]) &&
  !numberNoteWasUsed && straddlingTable.get(numerics[i]) !== '')
        plainText += straddlingTable.get(numerics[i]);
      else if(straddlingTable.has(numerics[i] + numerics[i + 1])
  && straddlingTable.get(numerics[i] + numerics[i + 1]) ===
  "FIG"){
          numberNoteWasUsed = !numberNoteWasUsed;
          i += 1;
      }
      else if(straddlingTable.has(numerics[i] + numerics[i + 1])
  && !numberNoteWasUsed){
          plainText += straddlingTable.get(numerics[i] +
  numerics[i + 1])
          i += 1;
      } else {
          plainText += numerics[i];
          i += 2;
```

```
      }
    }

    return plainText;
  };

  const convertPlainTextToPlainCode = (plainText,
  straddlingTable) => {
    const characters = plainText.split("");
    let plainCode = "";

    const reversedStraddlingTable =
  reverseStraddlingTable(straddlingTable);
    let numberWasUsed = false;

    characters.forEach((ch) => {
      const lowerCaseCharacter = ch.toLowerCase();

      if (reversedStraddlingTable.has(lowerCaseCharacter)) {
        plainCode +=
  reversedStraddlingTable.get(lowerCaseCharacter);
        numberWasUsed = false;
      } else if (parseInt(lowerCaseCharacter).toString() !==
  'NaN') {
        plainCode += reversedStraddlingTable.get("FIG");
        plainCode += lowerCaseCharacter + lowerCaseCharacter +
  lowerCaseCharacter;
        plainCode += reversedStraddlingTable.get("FIG");
        numberWasUsed = false;
      } else {
        plainCode += reversedStraddlingTable.get("#");
      }
    });

    return plainCode;
  };
```

У обох функціх формується ключ необхідної довжини, котрий потім буде також преобразований в числовий вираз. За формування ключа та його конвертацію у числовй вираз відповідає функція constructKey. У функції constructKey якщо у результаті формування ключа його довжина буде більшою за довжину числового виразу повідомлення/шифру то у числовий вираз повідомлення/шифру дописуються нулі. Код функції constructKey:

```
1. const constructKey = (key, plainCode, straddlingTable) => {
2.   const numericKey = convertPlainTextToPlainCode(key,
3. straddlingTable);
4.   let contructedKey = "";
5.
6.   while (contructedKey.length < plainCode.length) contructedKey
7. += numericKey;
8.
9.   while (contructedKey.length > plainCode.length) plainCode +=
10.  "0";
11.
12.    return { contructedKey, plainCode };
13.  };
```

З метою виконання додавання/віднімання по модулю 10 була реалізована фунція performDecimalOperation. Ця функція приймає на вхід три параметри: перша числова послідовність, друга числова послідовність та callback, що відповідає саме за операцію, що повинна бути виконана с числами у цих послідовностях. Код функції performDecimalOperation:

```
1. const performDecimalOperation = (
2.   numericSequence1,
3.   numericSequence2,
4.   callback
5. ) => {
6.   let result = "";
7.   for (let i = 0; i < numericSequence1.length; i++) {
8.     const num1 = parseInt(numericSequence1[i]);
9.     const num2 = parseInt(numericSequence2[i]);
10.       result += callback(num1, num2);
11.    }
12.
13.    return result;
14.  };
```

2) Демонстрація роботи програми

З метою демонстрації роботи програмної реалізації створеного алгоритму була створена UI сторінка , котру можна побачити на рисунку 1. На цій сторінці є елемент

згортання/розгортання, котрий містить у собі форму для кодування/декодування за допогомою шифру “Одноразовий блокнот”

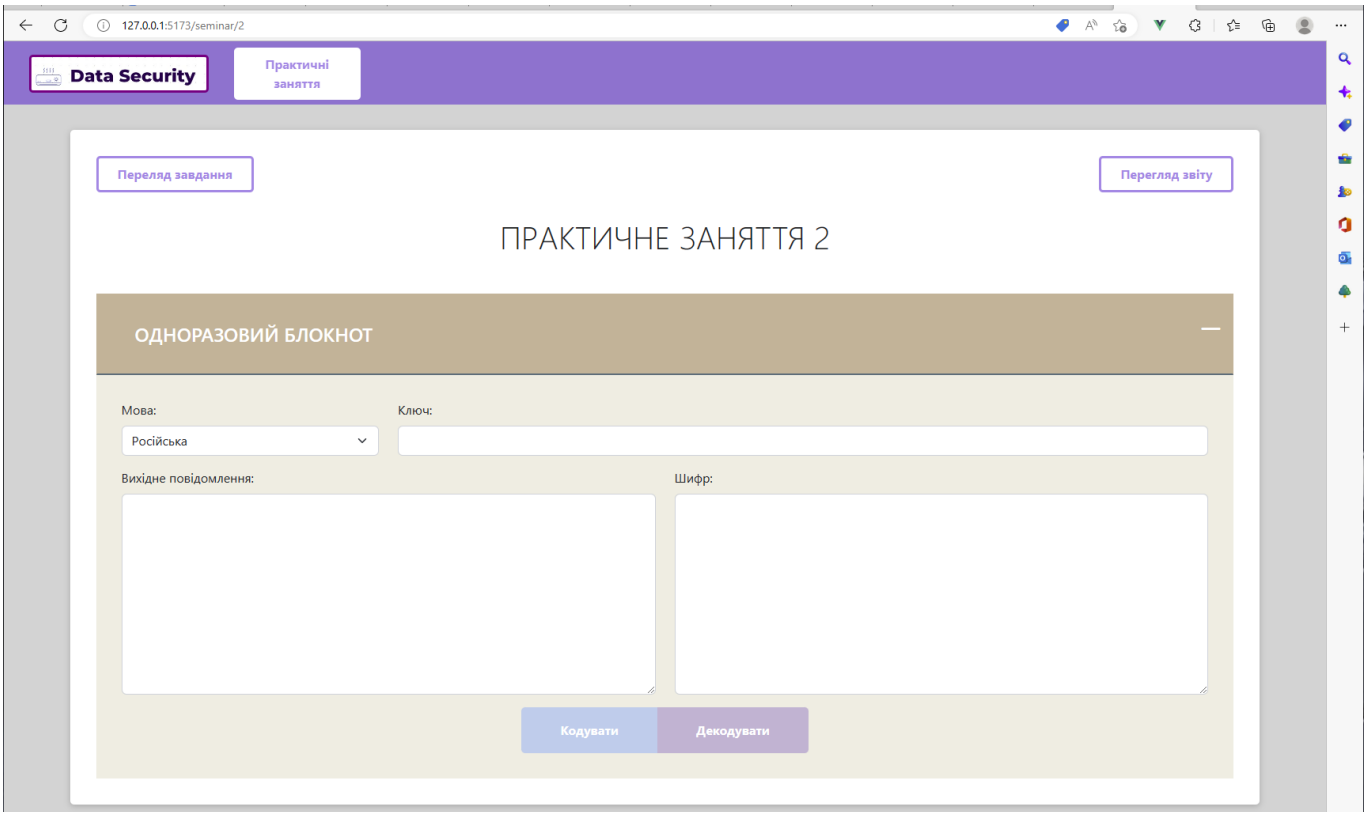

Рисунок 1 – Створена сторінка

Спочатку було розшифроване повідомлення за допомогою прикладного ключа за моїм варіантом у журналі (13). Тобто у даному випадку був обраний варіант завдання номер 3 з вихідним повідомленням → 188639984753915232037213791 2 та прикладним ключем “три”. Результат розшифрування повідомлення можливо побачити на рисунку 2.

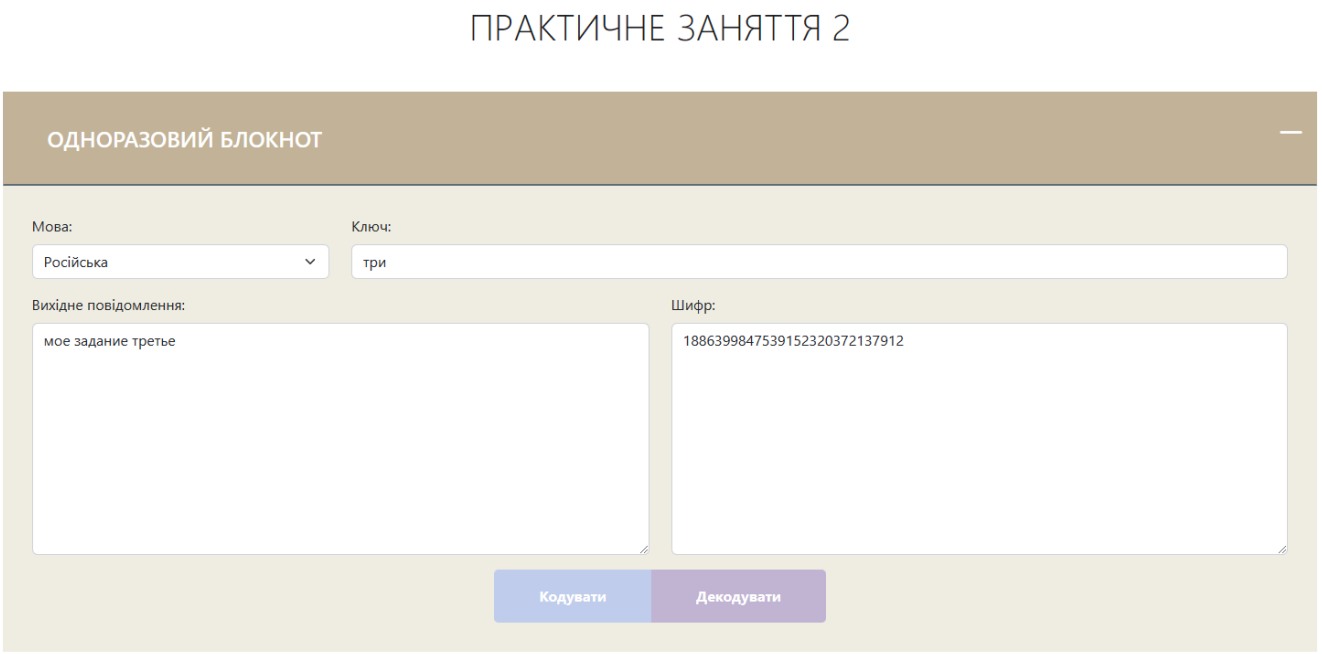

Рисунок 2 – Розшифрування повідомлення

У результаті розширування було отримане вихідне повідомлення “мое задание третье”. Тобто далі потрібно біло зашифрувати назву та даду випадкового свята поточним днем тижня. Результат виконання завдання можливо побачити на рисунку 3.

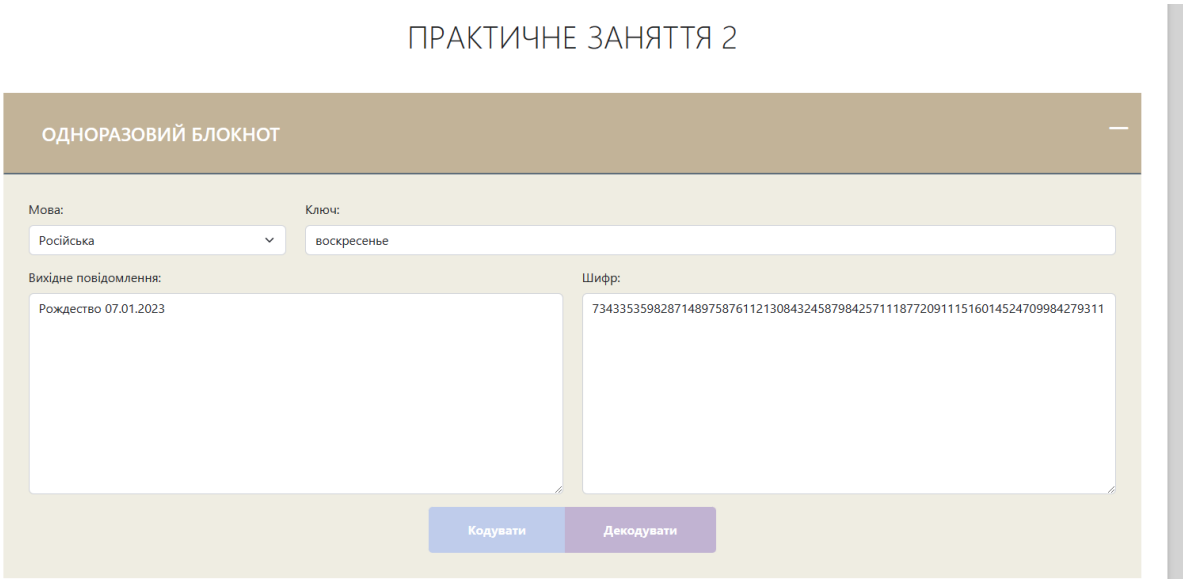

Рисунок 3 – Результат виконання шифрування назви і дати випадкового свята

Далі було отримане зашироване повідомлення від колеги → 845778992348072593517176843498048266766307073638587361040766304965630204888040466604046660404777 04, котре необхідно було розшифрувати по ключу “6” (його порядковий номер у журналі). Результат розшифрування повідомлення від колеги можливо побачити на рисунку 4.

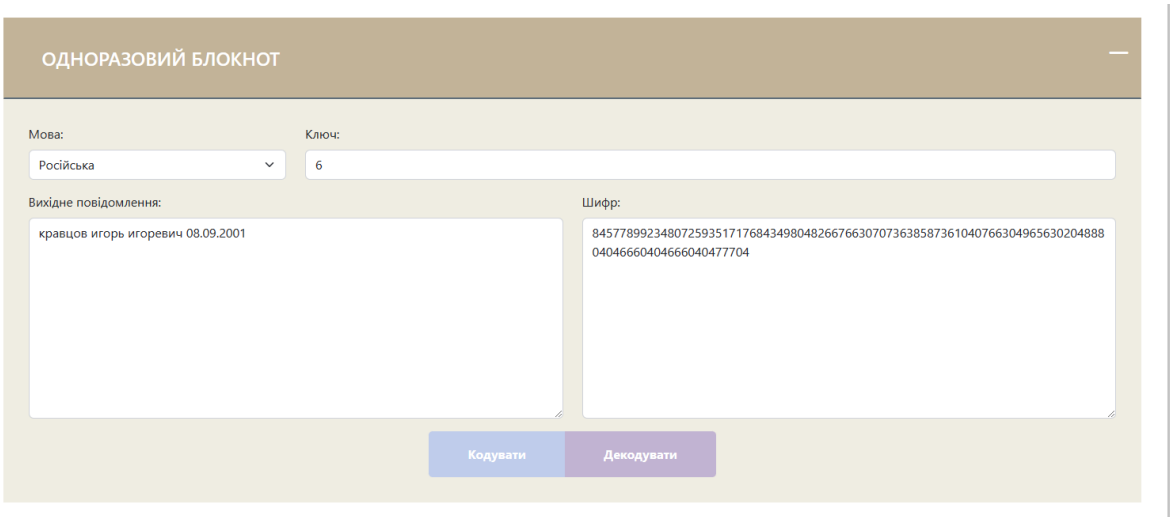

Рисунок 4 – Результат розшифрування повідомлення від колеги

Висновки: Ознайомився з шифром “Одноразовий блокнот”, відпрацював навички використовування цього шифру для кодування та декодування тексту. Створив програмну реалізацію шифру “Одноразовий блокнот”.

Контрольні питання

1. Що таке шифроблокнот?

Шифроблокнот – використовується для шифрування відкритого тексту, у шифроблоноті на кожному аркуші написана послідовність випадкових чисел.

2. Для чого потрібна таблиця стиснення?

Таблиця стиснення потрібна для того, щоб перетворити вихідний текст в цифри.

3. Як створюється таблиця стиснення?

Наприклад для створення таблиці стиснення використовується фраза “А И ТЕСНО”. У цій фразі задіяні сім найчастіше вживаємих літер в російській мові. Далі складається таблиця в першому рядку якої йдуть символи, що найчастіше зустрічаються, а в інших залишки алфавіту та додаткові символи. У першому рядку не задіяні цифри 8, 9 і 0 бо вони необхідні для позначки інших літер, котрі зустрічаються у тесті не часто (позначаються двома цифрами).

4. Чому переведений в цифровий вираз текст ще не є шифром?

Переведений в цифровий вираз текст ще не є шифром, бо такий числовий вираз досить просто зламати за допомогою частотного аналізу.

5. Що таке ключ?

Ключ – секретна інформація, що використовується криптографічним алгоритмом під час зашифрування/розшифрування повідомлень.

6. Що таке циркулянт?

Циркулянт – це матриця, у якій будь-який наступний рядок (стовпчик), починаючи з першого (з першого), отримується циклічною алфавітною перестановкою елементів попереднього рядка (стовпчика).