

Міністерство освіти та науки України  
Харківський національний університет радіоелектроніки  
Кафедра програмної інженерії

Звіт

До практичної роботи №1

З дисципліни: «Безпека програм та даних»

Виконав:

ст. гр. ПЗП-19-3

Селевич О.В.

Перевірив:

асистент кафедри ПІ

Олійник О.О.

Харків 2022

**Тема роботи:** Шифр Цезаря.

**Мета роботи:** Ознайомити студентів з шифром Цезаря, відпрацювати навички використання цього шифру для кодування та декодування тексту.

**Хід роботи:**

Під час практичного заняття була створені програмні реалізації шифрів Цезаря та Віженера за допомогою мови програмування JavaScript. Користувацький інтерфейс для демонстрації роботи шифрів був створений за допомогою JavaScript-фреймворку Vue.js.

## 1) Шифр Цезаря

Шифр Цезаря – тип шифру підстановки (моноалфавітної заміни), в якому кожен символ у відкритому тексті повинен замінюватися символом, розташованим на певній постійній кількості позицій лівіше або правіше за нього в алфавіті.

Для виконання кодування та декодування за допомогою шифру Цезаря були створені наступні функції `encrypt` та `decrypt`, що знаходяться у файлі `caesarsCipher.js`:

```
1. export const decrypt = (shift, cipher, language) => {
2.   const languageArray = getLanguageArray(language);
3.   let encryptionArray = getLanguageArray(language);
4.
5.   if (shift < 0) shift = languageArray.length + shift;
6.
7.   shiftArray(shift, encryptionArray);
8.
9.   return mapText(cipher, encryptionArray, languageArray, false);
10. };
11.
12. export const encrypt = (shift, message, language) => {
13.   const languageArray = getLanguageArray(language);
14.   let encryptionArray = getLanguageArray(language);
15.
16.   if (shift < 0) shift = languageArray.length + shift;
17.
18.   shiftArray(shift, encryptionArray);
```

```

19.
20.     return  mapText(message,  languageArray,  encryptionArray,
    false);
21. };

```

Ці функції приймають по три параметри: вихідне повідомлення/шифр, мова тексту та ключ. Функція `encrypt` відповідає за кодування, а `decrypt` відповідно для декодування. Спочатку виконується виклик функції `getLanguageArray`, що повертає масив з алфавітом. Далі виконується перевірка на наявність ключа з негативним зсувом. Функція `shiftArray` відповідає за виконання зсуву масиву для кодування/декодування. Код функції `shiftArray`:

```

1. export const shiftArray = (shift, encryptionArray) => {
2.   const n = encryptionArray.length;
3.
4.   const shiftedArray = [...Array(n)];
5.   for (let i = 0; i < n; i++) {
6.     shiftedArray[i] = encryptionArray[(i + shift) % n];
7.   }
8.   encryptionArray.splice(0, n, ...shiftedArray);
9. };

```

Далі виконується виклик функції `mapText` що повертає результуючий шифр/вихідне повідомлення та зіставляє символи з двох масивів (масиву алфавіту та масиву кодування/декодування) для виконання моноалфавітної заміни. Код функції `mapText`:

```

1. export const mapText = (
2.   text,
3.   textLanguageArray,
4.   textEncryptionArray,
5.   makeLowerCase
6. ) => {
7.   let result = [];
8.   let textCharacters = [...text.split(")];
9.
10.   textCharacters.forEach((elem) => {
11.     const lowerCaseElem = elem.toLowerCase();
12.     const index = textLanguageArray.indexOf(lowerCaseElem);
13.

```

```

14.     if (index === -1) result.push(elem);
15.     else {
16.         let resultCharacter = "";
17.         if (lowerCaseElem === elem || (lowerCaseElem !== elem &&
18. makeLowerCase)) {
19.             resultCharacter = textEncryptionArray[index];
20.         } else {
21.                                     resultCharacter =
textEncryptionArray[index].toUpperCase();
22.         }
23.         result.push(resultCharacter);
24.     }
25.     });
26.
27.     return result.join("");
28. };

```

## 2) Шифр Віженера

Шифр Віженера – метод поліалфавітного шифрування буквеного тексту з використанням ключового слова. Шифр Віженера складається з послідовності кількох шифрів Цезаря з різними значеннями зсуву. Для шифрування може використовуватися таблиця алфавітів – квадрат (таблиця) Віженера. Стосовно до російського алфавіту таблиця Віженера складається з рядків по 33 символи, причому кожен наступний рядок зсувається на кілька позицій. Таким чином, у таблиці виходить 33 різних шифрів Цезаря. На кожному етапі шифрування використовуються різні алфавіти, які обирають залежно від символу ключового слова.

Для виконання кодування та декодування за допомогою шифру Цезаря були створені наступні функції `encrypt` та `decrypt`, що знаходяться у файлі `vigenereCipher.js`:

```

1. export const decrypt = (key, cipher, language) => {
2.   const languageArray = getLanguageArray(language);
3.   const filledKeyArray = getFilledKey(key, cipher,
languageArray).split(
4.     ""

```

```

5.   );
6.   const cipherCharactersArray = cipher.split("");
7.
8.   let plainText = mapVigenereCipher(
9.     cipherCharactersArray,
10.    filledKeyArray,
11.    languageArray,
12.    (cipherCh, keyCh, n) => (cipherCh - keyCh + n) % n
13.  );
14.
15.   return plainText;
16. };
17.
18. export const encrypt = (key, message, language) => {
19.   const languageArray = getLanguageArray(language);
20.   const filledKeyArray = getFilledKey(
21.     key,
22.     message,
23.     languageArray
24.   ).split("");
25.   const messageCharactersArray = message.split("");
26.
27.   let cipherText = mapVigenereCipher(
28.     messageCharactersArray,
29.     filledKeyArray,
30.     languageArray,
31.     (msgCharacter, keyCharacter, n) => (msgCharacter +
    keyCharacter) % n
32.   );
33.
34.   return cipherText;
35. };

```

На вхід ці функції отримують по три параметри ключ, мова та вихідне повідомлення/шифр згідно з виконуваною операцією (кодування/декодування). Далі виконується виклик функції `getLanguageArray` для отримання алфавіту згідно з мовою повідомлення/шифру. Функція `getLanguageArray` імпортується з файлу `caesarsCipher.js`. Далі використовується функція `getFilledKey`, що відповідає за побудову ключа необхідної довжини. Код функції `getFilledKey`:

```

1. const getFilledKey = (key, message, languageArray) => {
2.   const keyLength = key.length;
3.   const messageLength = message.length;
4.
5.   if (keyLength >= messageLength) return key;

```

```

6.
7.   let newKey = "";
8.   newKey += key.toLowerCase();
9.   const messageCharatersArray = message
10.      .split("")
11.      .filter((ch) => languageArray.indexOf(ch.toLowerCase()) !==
12. -1);
13.   const keyCharacters = newKey.split("");
14.   let index = 0;
15.
16.   while (newKey.length < messageCharatersArray.length) {
17.     newKey += keyCharacters[index];
18.     index += 1;
19.     if (index === keyCharacters.length) index = 0;
20.   }
21.
22.   return newKey;
23. };

```

Після побудови ключа виконується виклик функції `mapVigenereCipher`, що відповідає за виконання поліалфавітної заміни. У цю функцію передається `callback`, який визначає саме яким чином будуть замінені літери з вихідного тексту/шифру. Код функції `mapVigenereCipher`:

```

1. const mapVigenereCipher = (
2.   inputCharactersArray,
3.   filledKeyArray,
4.   languageArray,
5.   callback
6. ) => {
7.   let outputText = "";
8.   let keyProgressIndex = 0;
9.
10.   for (let i = 0; i < inputCharactersArray.length; i++) {
11.     const lowerCaseCharacter =
12. inputCharactersArray[i].toLowerCase();
13.     const inputCharacterIndex =
14. languageArray.indexOf(lowerCaseCharacter);
15.
16.     if (inputCharacterIndex !== -1) {
17.       const keyCharacterIndex = languageArray.indexOf(
18.         filledKeyArray[keyProgressIndex]
19.       );
20.       const outputCharacter =
21.         languageArray[
22.           callback(inputCharacterIndex, keyCharacterIndex,

```

```

23.   languageArray.length)
24.       ];
25.       outputText +=
26.           inputCharactersArray[i] === lowerCaseCharacter
27.               ? outputCharacter
28.               : outputCharacter.toUpperCase();
29.       keyProgressIndex += 1;
30.   } else {
31.       outputText += inputCharactersArray[i];
32.   }
33.   }
34.
35.   return outputText;
36. };

```

Після виконання функція `mapVigenereCipher` повертає необхідний результат кодування/декодування.

### 3) Демонстрація роботи програми

З метою демонстрації роботи програмної реалізації створених алгоритмів була створена UI сторінка, котру можна побачити на рисунку 1. На цій сторінці знаходяться два елементи, що згортаються та розгортаються з відповідними назвами при натисканні на які відкриваються форми для тестування програмної реалізації алгоритмів. Також на цій сторінці є кнопки для перегляду завдання до практичного заняття та створеного звіту.

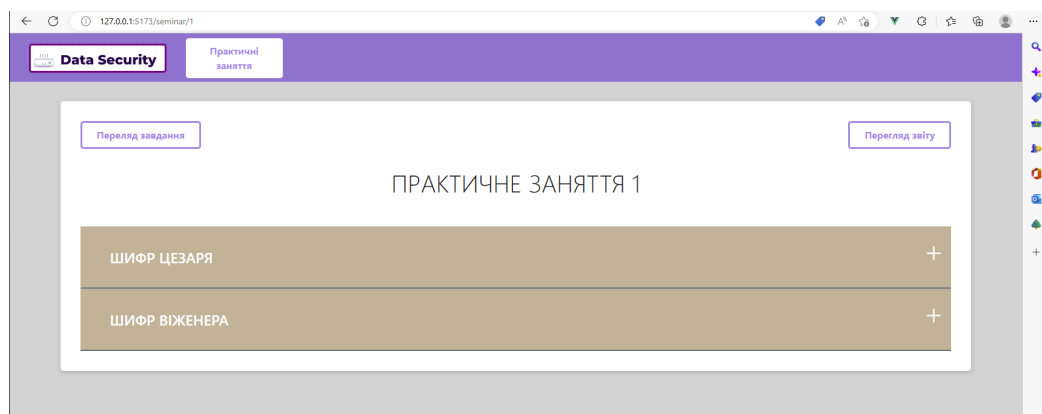


Рисунок 1 – Створена сторінка

Форми для кодування/декодування за допомогою шифрів Цезаря та Віженера виглядають практично ідентично, але головною різницею між ними є те, що у формі для виконання кодування/декодування за допомогою шифру Цезаря ключ є цілочисельним значенням для виконання зсуву, а у формі для кодування/декодування за допомогою шифру Віженера ключ може бути текстовим. Форму для кодування/декодування за допомогою шифру Цезаря можливо побачити на рисунку 2, а форму для кодування/декодування за допомогою шифру Віженера можливо побачити на рисунку 3.

The screenshot shows a web application titled "ШИФР ЦЕЗАРЯ" (Caesar Cipher). It features a light beige background with a darker beige header bar. The interface is divided into two main columns. The left column contains a "Ключ:" (Key) label above a text input field with the value "0", and a "Вихідне повідомлення:" (Original message) label above a large, empty text area. The right column contains a "Мова:" (Language) label above a dropdown menu showing "Англійська" (English), and a "Шифр:" (Cipher) label above another large, empty text area. At the bottom, there are two buttons: "Кодувати" (Encode) in blue and "Декодувати" (Decode) in purple.

Рисунок 2 – Вигляд форми для кодування/декодування за допомогою шифру Цезаря

The screenshot shows a web application titled "ШИФР ВІЖЕНЕРА" (Vigenere Cipher). It has a similar layout to the Caesar cipher interface, with a light beige background and a darker beige header bar. The left column has a "Ключ:" (Key) label above an empty text input field, and a "Вихідне повідомлення:" (Original message) label above a large, empty text area. The right column has a "Мова:" (Language) label above a dropdown menu showing "Англійська" (English), and a "Шифр:" (Cipher) label above another large, empty text area. At the bottom, there are two buttons: "Кодувати" (Encode) in blue and "Декодувати" (Decode) in purple.



Рисунок 3 – Вигляд форми для кодування/декодування за допомогою шифру Віженера

Приклади роботи створеного застосунку з кодуванням/декодуванням за допомогою шифру Цезаря можливо побачити на рисунках 4, 5. Дані, використані для тестування роботи алгоритмів шифрування були взяті з методичних вказівок.

ПРАКТИЧНЕ ЗАНЯТТЯ 1

ШИФР ЦЕЗАРЯ

Ключ:

11

Мова:

Російська

Вихідне повідомлення:

встрепанные волосы и небрежность в одежде. Представляет, вообще не посмотрел в

Шифр:

мъэыпъжшщѐп мщщщѐ у шплыпсшщѐж м щолсп. ъыпоъэкмщѐп, мщщлп шп ъщщщщѐпщ м

Кодувати

Декодувати

Рисунок 4 – Приклад кодування за допомогою шифру Цезаря

ПРАКТИЧНЕ ЗАНЯТТЯ 1

ШИФР ЦЕЗАРЯ

Ключ:

19

Мова:

Російська

Вихідне повідомлення:

поклонниками, ходила танцевать, один раз даже сбила юбкой бутылку пива ему на колени.

Шифр:

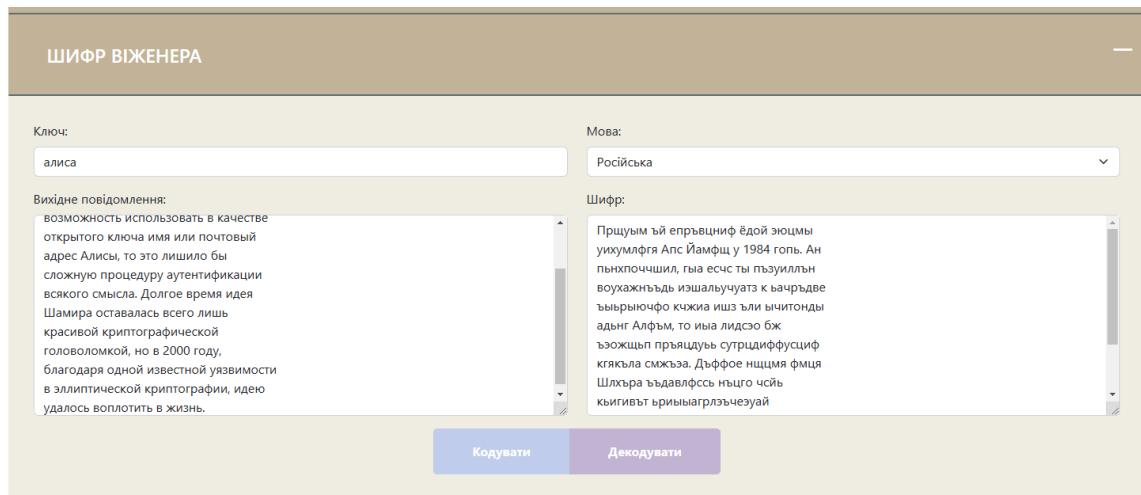
вбэюбаэыэты, эбщѐнот етаищфѐо, бщѐа  
гть цтщц дущѐнот руэбъ уѐѐнѐэѐ выфѐ  
щѐѐ ат эбючаы.

Кодувати

Декодувати

Рисунок 5 – Приклад декодування за допомогою шифру Цезаря

Приклади роботи створеного застосунку з кодуванням/декодуванням за допомогою шифру Віженера можливо побачити на рисунках 6, 7.



**ШИФР ВІЖЕНЕРА**

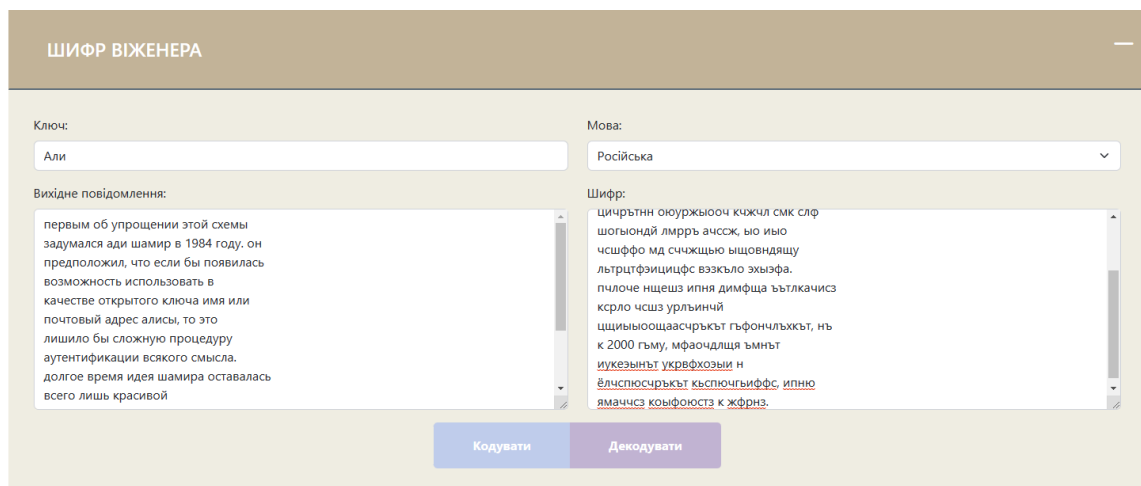
Ключ:

Мова:

Вихідне повідомлення:  
возможность использовать в качестве открытого ключа имя или почтовый адрес Алисы, то это лишило бы сложную процедуру аутентификации всякого смысла. Долгое время идея Шамира оставалась всего лишь красивой криптографической головоломкой, но в 2000 году, благодаря одной известной уязвимости в эллиптической криптографии, идею удалось воплотить в жизнь.

Шифр:  
Прщуым їй епръвщиф ёдой эюцмы уиухмфля Алс Йамфщ у 1984 гопь. Ан пьнхпоччиши, гяа есчс ты пъзуилън воухажнъдь изшальчутатз к ьачръдве ьырыночфо кчжиа ишз ьли ьчитонды адьнг Алфъм, то ина лидсэо бж ьэозщъп пръядцуь сутрицифусциф кгякъла смжъээ. Дъффео нщцмъа фмця Шлхъра ьъдавлфсь нъго чсйь кыгивьт ьриыагрлъэчезуай

Рисунок 6 – Пример кодирования за допомогою шифру Віженера



**ШИФР ВІЖЕНЕРА**

Ключ:

Мова:

Вихідне повідомлення:  
первым об упрощении этой схемы задумался ади шамир в 1984 году. он предположил, что если бы появилась возможность использовать в качестве открытого ключа имя или почтовый адрес алисы, то это лишило бы сложную процедуру аутентификации всякого смысла. долгое время идея шамира оставалась всего лишь красивой

Шифр:  
щичъртънн оюуржъюоч кчжчл смк слф шогъондй лмррь ачсжс, ьо иьо чсшффо мд сччжщъю ьщовндящъ лътрцтфэицицфс възкъло зхъзфа. пчлоче нщещз игня димфща ьътлкачисз ксрло чсшз урльиний щшыноощаасчъркът гъфончлъххът, нъ к 2000 гъму, мфаочдлща ьмнът иукезынът уквфхозыи н ёлчспюсчъркът къспночгыффс. ипню ямаччз кыфноуэз к жфрнз.

Рисунок 7 – Пример декодирования за допомогою шифру Віженера

**Висновки:** Ознайомися з шифром Цезаря, відпрацював навички використання цього шифру для кодування та декодування тексту. Створив програмну реалізацію шифрів Цезаря та Віженера.

## Контрольні питання

### 1. Що таке шифр Цезаря?

Шифр цезаря – один з найдавніших шифрів. При шифруванні кожен символ замінюється іншим, віддаленим від нього в алфавіті на фіксоване число позицій. Шифр Цезаря можна класифікувати як шифр підстановки, при більш вузької класифікації – шифр простої заміни.

### 2. Який ключ використовував Цезар, як він відрізняється від шифру його племінника?

Цезар використовував ключ 3, а його племінник використовував цей шифр, але зі зсувом вправо на один, і він не повторювався до початку алфавіту.

### 3. Яка його математична модель?

Математична модель шифру Цезаря є наступною  $\rightarrow$  якщо зіставити кожному символу алфавіту його порядковий номер (нумеруя з 0), то шифрування і дешифрування можна виразити формулами модульної арифметики:

$$y = (x + k) \bmod n$$

$$x = (y + n) - (k \bmod n) \bmod n, \text{ де:}$$

- $x$  – символ відкритого тексту;
- $y$  – символ шифрованого тексту;
- $n$  – потужність алфавіту;
- $k$  – ключ.

### 4. Де зараз використовується цей шифр?

Шифр Цезаря зі зсувом тринадцять використовується в алгоритмі ROT13, простому методі заплутування тексту, широко застосовується в Usenet, і використовується скоріше як спосіб приховування спойлерів, ніж як метод шифрування.