UDC 517.948

S.A. YEPISKOPOSYAN, A.S. YEPISKOPOSYAN

ENCRYPTING INFORMATION USING WALSH SYSTEMS

This article discusses a method for encrypting the text information using the Walsh system and generating a secret key.

***Keywords*:** encrypting, Walsh matrix, secret key.

Introduction

With the development of modern technologies and the expansion of information exchange, ensuring data security becomes a priority. In [1], an encryption algorithm based on the Walsh system is presented, which makes it possible to improve the security of text information transmission. However, nowadays, a more reliable and advanced encryption method is required to protect confidential data from prying eyes.

The purpose of this article is to present a study to improve the encryption method using the Walsh system. We will present a more complex and secure method, including the generation of a secret key. This key will serve as an additional layer of protection, making the encryption process more secure and resistant to hacking.

1. Walsh matrix

Walsh functions were introduced into mathematics by J. Walsh in 1923 [2].

Definitions: Let's put $W_0(x) = 0$. Next for anyone $n = \sum_{s=1}^{k} 2^{m_s}$ $m_1 > m_2 > \ldots > m_k$ we bet $W_n(x) = \prod_{s=1}^{n} r_{m_s}(x)$, where $\{r_k(x)\}_{k=0}^{\infty}$ is the Rademacher system:

$$r_0(x) = \begin{cases} 1, x \in \left[0, \frac{1}{2}\right), \\ -1, x \in \left(\frac{1}{2}, 1\right] \end{cases}, \quad r_0(x+1) = r_0(x), \; r_k(x) = r_0(2^k x), k = 1,2,\ldots.$$

They represent a complete set of orthogonal functions that take only the values +1 and –1.

Walsh matrix. In mathematics, the Walsh matrix is a specific square matrix dimensions 2^n, Where n is some natural number. The elements of the matrix are either +1 or −1, and its rows and columns are orthogonal, i.e. the scalar product is equal to zero. It should be noted that each row of the Walsh matrix corresponds to

the value of the Walsh function. Walsh matrices are a special case Hadamard matrices. The naturally ordered Adam-ra matrix is defined by the formula recursive the formula is below, and the ordered Hadamard matrix is formed by rearranging the rows so that the number of sign changes in the row is in ascending order [3]. Walsh-Hadamard matrices can be determined by the following simple rule:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},\ H_4 = \left[\begin{array}{c|c} H_2 & H_2 \\ \hline H_2 & -H_2 \end{array}\right] = \left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right], H_8 = \left[\begin{array}{c|c} H_4 & H_4 \\ \hline H_4 & -H_4 \end{array}\right] \text{etc.}$$

2. Key encryption algorithm

In this article we will look at a specific example of the application of an encryption method using the Walsh system and a secret key. Specifically, we use the key to generate a random permutation of the rows and columns of the Walsh transform matrix.

Below is a detailed description of how this can be done:

1. **Key generation:** first, let's create a secret key. The secret key is a sequence of bits that determines how the rows and columns of the Walsh matrix will be rearranged. This allows you to further complicate the data structure, making it less predictable for potential attackers.

2. **Applying the key to the Walsh matrix:** now we can use the secret key to set up the Walsh transform. One way to do this is to generate a random permutation of the rows and columns of the Walsh transform matrix based on the key. Each bit of the key can determine which rows and columns of the matrix should be rearranged.

3. **Encryption:** during the data encryption process, use the transformed Walsh matrix (taking into account the key) to transform the input data. This ensures that each time you use different keys you will get a different Walsh matrix, making it harder to crack.

4. **Decryption:** when decrypting data, also use the key to generate the same permutation of rows and columns to return the data to its original state.

5. **Key storage:** be sure to securely store the private key. Access to the key must be limited and controlled to prevent data leaks.

This key addition method will strengthen the strength of the presented Walsh transform algorithm, since even if an attacker manages to learn the Walsh transform algorithm, he will also need to know the secret key to correctly interpret the transformed data.

Now let's give a specific example.

Let us have:

initial data:[10, -3, 5, 1, 0, -1, 4, 2]

secret key: [1, 0, 0, 0, 1, 0, 0, 1].

1. Creating a Walsh matrix: we start by creating a Walsh matrix for the length of the input data (8 elements). The Walsh matrix will look like Fig.1.

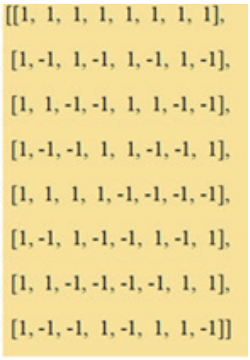

Fig. 1

2. Applying a secret key to the Walsh matrix:

The secret key [1, 0, 0, 0, 1, 0, 0, 1] specifies which rows and columns of the Walsh matrix should be rearranged:

- The first and fifth rows are swapped because the corresponding key bits are 1.
- The same thing happens with the first and fifth columns.

The Walsh matrix after applying the key will look like Fig. 2.

3. Data encryption: now we use the Walsh matrix after applying the key to encrypt the original data. The operation of multiplying the matrix by the input data gives the encrypted data:

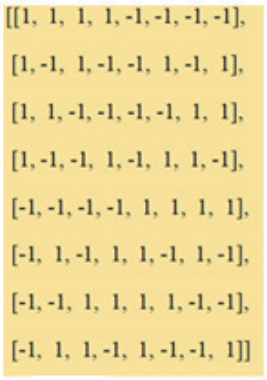

Fig. 2

Encrypted data: [12, 6, 6, 2, 10, 14, 2, -2]

4. Data decryption: to decrypt the data we use the same matrix

Walsh after using the key. We also use the matrix inverse of this matrix to perform the inverse transformation. The operation of multiplication by an inverse matrix returns the original data:

Decrypted data: [10, -3, 5, 1, 0, -1, 4, 2]

Thus, the secret key [1, 0, 0, 0, 1, 0, 0, 1] was used to adjust the Walsh matrix, resulting in the encryption of the original data. After decryption, using the same key, the data was successfully restored to its original state.

3.Python Code

To provide a practical perspective, we offer a sample Python code:

```
import numpy as np
# Generating a secret key
secret_key = [1, 0, 0, 0, 1, 0, 0, 1]
# Function to generate a Walsh matrix of a given size
def generate_walsh_matrix(n):
 if n == 1:
```

```
    return np.array([[1]])
    else:
    upper_left = generate_walsh_matrix(n // 2)
    upper_right = generate_walsh_matrix(n // 2)
    lower_left = generate_walsh_matrix(n // 2)
    lower_right = -generate_walsh_matrix(n // 2)
    top_half = np.concatenate((upper_left, upper_right), axis=1)
    bottom_half = np.concatenate((lower_left, lower_right), axis=1)
    walsh = np.concatenate((top_half, bottom_half), axis=0)
    return walsh
# Function to encrypt data using a key
def encrypt_data(data, key):
    n = len(data)
    walsh_matrix = generate_walsh_matrix(n)
    encrypted_data = np.dot(walsh_matrix, data)
    return encrypted_data
# Function to decrypt data using a key
def decrypt_data(encrypted_data, key):
    n = len(encrypted_data)
    walsh_matrix = generate_walsh_matrix(n)
    decrypted_data = np.dot(walsh_matrix, encrypted_data) / n
    return decrypted_data
# Initial data
original_data = np.array([10, -3, 5, 1, 0, -1, 4, 2])
# Encryption
encrypted_data = encrypt_data(original_data, secret_key)
# Decryption
decrypted_data = decrypt_data(encrypted_data, secret_key)
print("Original data:", original_data)
print("Secret Key:", secret_key)
print("Encrypted data:", encrypted_data)
print("Decrypted data:", decrypted_data)
```

Original data: [10 -3 5 1 0 -1 4 2]
Secret Key: [1, 0, 0, 0, 1, 0, 0, 1]
Encrypted data: [18 20 -6 8 8 14 8 10]
Decrypted data: [10. -3. 5. 1. 0. -1. 4. 2.]

REFERENCES

1. **Episkoposian S. A.** Application of Walsh system in Data Encryption.- Tuijin Jishu // Journal of Propulsion Technology. -2023.- 44, n. 3.- P. 494 – 502.
2. **Walsh J.L.** A Closed Set of Normal Orthogonal Functions// American Journal of Mathematics.- 1923.- 45, n.1.-P. 5-24.
3. **Horadam K.J.** Hadamard Matrices and Their Applications.- Princeton University Press, Princeton and Oxford, 2007.

Ս.Ա. ԵՊԻՍԿՈՊՈՍՅԱՆ, Ա.Ս. ԵՊԻՍԿՈՊՈՍՅԱՆ

ՏԵՂԵԿՈՒՅԹԻ ԿՈԴԱՎՈՐՈՒՄԸ ՈՒՈԼՇԻ ՀԱՄԱԿԱՐԳԻ ՄԻՋՈՑՈՎ

Ուսումնասիրվել են Ուոլշի համակարգի և գաղտնի բանալու ստեղծման միջոցով տեքստային տեղեկույթի կոդավորման մեթոդները:

Առանցքային բառեր. կոդավորում, Ուոլշի մատրից, գաղտնի բանալի:

С.А. ЕПИСКОПОСЯН, А.С. ЕПИСКОПОСЯН

ШИФРОВАНИЕ ИНФОРМАЦИИ С ПОМОЩЬЮ СИСТЕМЫ УОЛША

Рассмотрен метод шифрования текстовой информации с помощью системы Уолша и генерации секретного ключа.

Ключевые слова: шифрование, матрица Уолша, секретный ключ.

ՀՏԴ 517.518

Ռ.Վ. ԴԱԼԼԱՔՅԱՆ

ԴԻՐԻԽԼԵԻ ՏԻՊԻ ԴԱՍԵՐԻ ԶՐՈՆԵՐԻ ԲԱԶՄՈՒԹՅԱՆ ՄԱՍԻՆ

Աշխատանքում ապացուցվում է, որ Բլյաշկեի պայմանը բավարարող և Շտոլցի անկյան ներսում ընկած ցանկացած հաջորդականություն հանդիսանում է D_s^2 դասերի զրոների բազմություն ցանկացած s -ի համար՝ $s \in (0,1)$: Ապացուցվում են մի քանի այլ պնդումներ ևս:

Առանցքային բառեր. Դիրիխլեի տիպի դասեր, Շտոլցի անկյուն, Կառլեսոնի բազմություն, Բլյաշկեի արտադրյալ, A_α^p – դասեր:

Ընդունենք՝ $-1 < \alpha < +\infty$, $0 < p < +\infty$, D –ն կոմպլեքս հարթության միավոր շրջանն է, և թող $H(D)$ –ն D –ում հոլոմորֆ ֆունկցիաների բազմությունն է: A_α^p ֆունկցիոնալ դասերը սահմանվում են որպես $H(D)$ –ի հետևյալ պայմանը բավարարող ֆունկցիաների բազմություն՝

$$\int_D \left(1-|z|^2\right)^\alpha |f(z)|^p \, dA(z) < +\infty,$$

որտեղ $dA(z) = \frac{1}{\pi} dxdy$:

Միավոր շրջանում հոլոմորֆ այն ֆունկցիաների բազմությունը, որոնց համար $f' \in A_\alpha^p$, նշանակենք D_α^p –ով: Եթե $p > 1+\alpha$, ապա այս դասերը կոչվում են Դիրիխլեի տիպի դասեր: