Элементы криптографии. Однократное гаммирование

Косолапов Степан [1]

[1]Российский Университет Дружбы Народов

21 октября, 2023, Москва, Россия

Цели и задачи работы

Цель лабораторной работы

Освоить на практике применение режима однократного гаммирования.

Процесс выполнения лабораторной работы

Вспомогательные функции

```
const stringToHex = (str) => {
    return str.split('').map(c => c.charCodeAt(0).toString(16)).join(' ');
}

const hexToString = (hexStr) => {
    return hexStr.split(' ').map(c => String.fromCharCode(Number.parseInt(c, 16)
}
```

Функция для генерации ключа

```
const generateKey = (length) => {
    const result = [];

    for (let i = 0; i < length; i++) {
        const asciiCode = Math.floor(Math.random() * 1048);
        result.push(asciiCode.toString(16));
    }

    return result.join(' ');
}
```

Функция шифрования

```
const gammingCipher = (hexText, hexKey) => {
    const textSplit = hexText.split(' ');
    const keySplit = hexKey.split(' ');

    if (textSplit.length !== keySplit.length) {
        throw new Error('Key and message must have equal lengths.');
    }

    return textSplit.map((textCharHex, i) => {
        const keyCharHex = keySplit[i];

        const xorResult = Number.parseInt(textCharHex, 16) ^ Number.parseInt(ke

        return xorResult.toString(16);
    }).join(' ')
```

Генерация ключа

```
const initialTextLength = 'Штирлиц – Вы Герой!!!!'.length;
console.log('initialTextLength:', initialTextLength);
console.log('generatedKey:', generateKey(initialTextLength));
```

```
node index.js
```

```
initialTextLength: 22
generatedKey: 203 3e7 2ea ec 2dc 29 3b4 10b 7f 23 33b 185 1ac 121 26f 97 1d5
```

Шифрование ключом

```
const hexMessage = stringToHex(message);
const encryptedMessage = gammingCipher(hexMessage, hexKey);

console.log('charMessage:', message);
console.log('hexMessage:', hexMessage);
console.log('encryptedMessage:', encryptedMessage);
```

```
node index.js
```

```
charMessage: Штирлиц – Вы Герой!!!!
hexMessage: 428 442 438 440 43b 438 446 20 2013 20 412 44b 20 413 435 440
hexKey: 203 3e7 2ea ec 2dc 29 3b4 10b 7f 23 33b 185 1ac 121 26f 97 1d5 3ad 1
encryptedMessage: 62b 7a5 6d2 4ac 6e7 411 7f2 12b 206c 3 729 5ce 18c 532 6
```

Нахождение ключа

```
const newYearMessage = 'С Новым Годом, друзья!';
const hexNewYearMessage = stringToHex(newYearMessage);
const hexNewYearKey = gammingCipher(encryptedMessage, hexNewYearMessa

console.log('charNewYearMessage:', newYearMessage);
console.log('hexNewYearMessage:', hexNewYearMessage);
console.log('hexNewYearKey:', hexNewYearKey)
```

```
node index.js
```

```
charNewYearMessage: С Новым Годом, друзья!
hexNewYearMessage: 421 20 41d 43e 432 44b 43c 20 413 43e 434 43e 43c 2c 2
hexNewYearKey: 20a 785 2cf 92 2d5 5a 3ce 10b 247f 43d 31d 1f0 5b0 51e 67a e
```

Проверка правильности решения

```
console.log('decrypting initial message with hexNewYearKey...')

console.log('charDecrypted', hexToString(gammingCipher(encryptedMessage, h
console.log('hexDecrypted', gammingCipher(encryptedMessage, hexNewYearKe
```

```
node index.js
```

```
decrypting initial message with hexNewYearKey...
charDecrypted С Новым Годом, друзья!
hexDecrypted 421 20 41d 43e 432 44b 43c 20 413 43e 434 43e 43c 2c 20 434 4
```

Выводы по проделанной работе

Вывод

В данной работе мы освоили на практике применение режима однократного гаммирования.