**EP 13.10.2023 CRIPTOGRAFIE**

**1.** **Алгоритмы шифрования можно классифицировать по различным критериям, таким как метод шифрования, ключевые длины, алгоритмы и протоколы::**

***1. Симметричное шифрование:***

Преимущества:

* - Быстрое шифрование и дешифрование.
* - Простота реализации.
* - Эффективно для больших объемов данных.

Недостатки:

* - Проблемы с обменом ключами (необходимо безопасно передавать ключи другой стороне).
* - Один ключ используется для шифрования и дешифрования, что делает его уязвимым для атак, если ключ скомпрометирован.

***2. Асимметричное шифровани***:

Преимущества:

* - Безопасный обмен ключами, так как существует открытый и закрытый ключ.
* - Более безопасное, чем симметричное шифрование.

Недостатки:

* - Медленнее, чем симметричное шифрование.
* - Требует больше ресурсов для выполнения.
* - Неэффективно для шифрования больших объемов данных.

***3. Гибридное шифрование:***

Преимущества:

* - Комбинирует преимущества симметричного и асимметричного шифрования.
* - Обеспечивает безопасный обмен ключами, используя асимметричное шифрование, а затем шифрует данные симметричным шифрованием.

Недостатки:

* - Более сложно реализовать.

***4. Шифрование с открытым ключом (Public Key Infrastructure, PKI):***

Преимущества:

* - Обеспечивает безопасный обмен ключами с помощью асимметричного шифрования.
* - Широко используется для безопасной аутентификации и обмена данных в интернете.

Недостатки:

* - Требует инфраструктуры для управления ключами (сертификационные центры, цифровые сертификаты).

***5. Шифрование с открытым текстом (Plain Text Encryption):***

Преимущества:

* - Обеспечивает конфиденциальность данных.
* - Подходит для коротких сообщений и данных, где скорость менее важна.

Недостатки:

* - Не предоставляет целостности данных.
* - Не подходит для больших объемов данных.

Каждый тип шифрования имеет свои особенности и подходит для разных сценариев. Выбор конкретного типа зависит от требований безопасности, производительности и контекста использования. Важно правильно настраивать и управлять ключами при использовании любого алгоритма шифрования, чтобы предотвратить утечку данных.

**2. Пусть сообщение M=NameFirstName. Используя алгоритмы шифрования Afin, зашифруйте и расшифруйте сообщение M, выбрав правильный ключ шифрования (замените LastName на свои личные).**

<!DOCTYPE html>

<html>

<head>

    <title>Afin Cipher</title>

</head>

<body>

    <h1>Шифр Afin</h1>

    <input type="text" id="message" placeholder="Введите ваше сообщение">

    <input type="number" id="a" placeholder="Введите значение «а»">

    <input type="number" id="b" placeholder="Введите значение «b»">

    <button onclick="encrypt()">Зашифровать</button>

    <button onclick="decrypt()">Расшифровать</button>

    <p id="result"></p>

    <script>

        function modInverse(a, m) {

            a = (a % m + m) % m;

            for (let x = 1; x < m; x++) {

                if ((a \* x) % m == 1) {

                    return x;

                }

            }

            return 1;

        }

        function encrypt() {

            const message = document.getElementById("message").value;

            const a = parseInt(document.getElementById("a").value);

            const b = parseInt(document.getElementById("b").value);

            if (isNaN(a) || isNaN(b)) {

                alert("Введите действительные значения «a» и «b».");

                return;

            }

            let result = "";

            for (let i = 0; i < message.length; i++) {

                const charCode = message.charCodeAt(i);

                if (charCode >= 65 && charCode <= 90) {

                    const encryptedCharCode = (a \* (charCode - 65) + b) % 26 + 65;

                    result += String.fromCharCode(encryptedCharCode);

                } else if (charCode >= 97 && charCode <= 122) {

                    const encryptedCharCode = (a \* (charCode - 97) + b) % 26 + 97;

                    result += String.fromCharCode(encryptedCharCode);

                } else {

                    result += message.charAt(i);

                }

            }

            document.getElementById("result").textContent = "Зашифрованное сообщение: " + result;

        }

        function decrypt() {

            const message = document.getElementById("message").value;

            const a = parseInt(document.getElementById("a").value);

            const b = parseInt(document.getElementById("b").value);

            if (isNaN(a) || isNaN(b)) {

                alert("Введите действительные значения «a» и «b».");

                return;

            }

            const aInverse = modInverse(a, 26);

            let result = "";

            for (let i = 0; i < message.length; i++) {

                const charCode = message.charCodeAt(i);

                if (charCode >= 65 && charCode <= 90) {

                    const decryptedCharCode = (aInverse \* (charCode - 65 - b + 26)) % 26 + 65;

                    result += String.fromCharCode(decryptedCharCode);

                } else if (charCode >= 97 && charCode <= 122) {

                    const decryptedCharCode = (aInverse \* (charCode - 97 - b + 26)) % 26 + 97;

                    result += String.fromCharCode(decryptedCharCode);

                } else {

                    result += message.charAt(i);

                }

            }

            document.getElementById("result").textContent = "Расшифрованное сообщение: " + result;

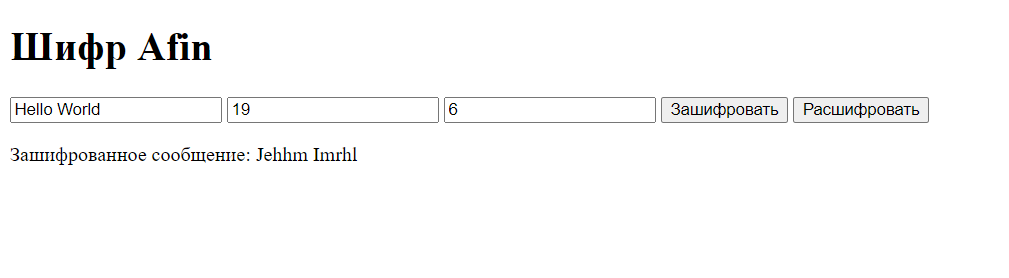
        }

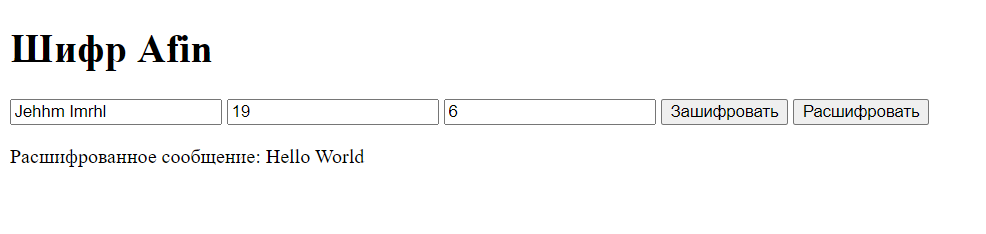
    </script>

</body>

</html>

**2.1 Screenshot**

****



3. **Пусть сообщение M=NameFirstName. Используя алгоритмы транспозиционного шифрования, выполните шифрование и расшифровку сообщения М, выбрав правильный ключ шифрования (замените Фамилию на свои личные).**

<!DOCTYPE html>

<html>

<head>

    <title>Транспозиционное шифрование</title>

</head>

<body>

    <h1>Транспозиционное шифрование</h1>

    <input type="text" id="message" placeholder="Введите ваше сообщение">

    <input type="text" id="key" placeholder="Введите ключ шифрования (например, вашу фамилию)">

    <button onclick="encrypt()">Зашифровать</button>

    <button onclick="decrypt()">Дешифровать</button>

    <p id="result"></p>

    <script>

        function encrypt() {

            const message = document.getElementById("message").value;

            const key = document.getElementById("key").value;

            const encryptedMessage = transposeEncrypt(message, key);

            document.getElementById("result").textContent = "Зашифрованное сообщение: " + encryptedMessage;

        }

        function decrypt() {

            const message = document.getElementById("message").value;

            const key = document.getElementById("key").value;

            const decryptedMessage = transposeDecrypt(message, key);

            document.getElementById("result").textContent = "Дешифрованное сообщение: " + decryptedMessage;

        }

        function transposeEncrypt(message, key) {

            const keyLength = key.length;

            const messageLength = message.length;

            const numRows = Math.ceil(messageLength / keyLength);

            let matrix = new Array(numRows);

            for (let i = 0; i < numRows; i++) {

                matrix[i] = new Array(keyLength);

            }

            let index = 0;

            for (let i = 0; i < numRows; i++) {

                for (let j = 0; j < keyLength; j++) {

                    if (index < messageLength) {

                        matrix[i][j] = message[index];

                        index++;

                    }

                }

            }

            let encryptedMessage = '';

            for (let j = 0; j < keyLength; j++) {

                for (let i = 0; i < numRows; i++) {

                    if (matrix[i][j]) {

                        encryptedMessage += matrix[i][j];

                    }

                }

            }

            return encryptedMessage;

        }

        function transposeDecrypt(encryptedMessage, key) {

            const keyLength = key.length;

            const messageLength = encryptedMessage.length;

            const numRows = Math.ceil(messageLength / keyLength);

            let matrix = new Array(numRows);

            for (let i = 0; i < numRows; i++) {

                matrix[i] = new Array(keyLength);

            }

            let index = 0;

            for (let j = 0; j < keyLength; j++) {

                for (let i = 0; i < numRows; i++) {

                    if (index < messageLength) {

                        matrix[i][j] = encryptedMessage[index];

                        index++;

                    }

                }

            }

            let decryptedMessage = '';

            for (let i = 0; i < numRows; i++) {

                for (let j = 0; j < keyLength; j++) {

                    if (matrix[i][j]) {

                        decryptedMessage += matrix[i][j];

                    }

                }

            }

            return decryptedMessage;

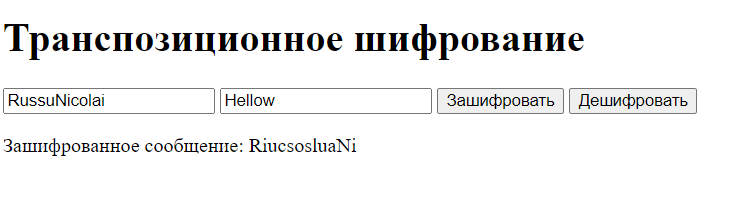
        }

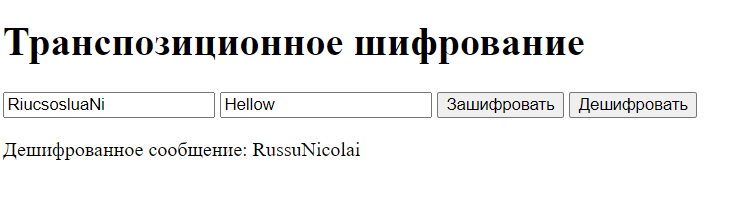
    </script>

</body>

</html>

3.1





**4. Исследование последствий использования машины "Энигма" во Второй мировой войне:**

Машина "Энигма" была шифровальным устройством, которое использовалось нацистами во Второй мировой войне для зашифровывания важных сообщений и коммуникаций. Это было одним из ключевых технических средств, которые позволяли нацистам обеспечивать конфиденциальность связи и секретность операций. Рассмотрим последствия использования машины "Энигма" в контексте войны:

* Обеспечение конфиденциальности важных сообщений: Использование "Энигмы" позволяло нацистам обмениваться информацией без опасения, что их коммуникации будут перехвачены и расшифрованы противниками. Это позволяло им планировать военные операции и передвижения с большей уверенностью.
* Развитие криптоанализа: Попытки альянса, включая Великобританию и Соединенные Штаты, декодировать сообщения, зашифрованные на машинах "Энигма", привели к развитию современной криптоаналитической техники. Это способствовало разработке новых методов и технологий для взлома шифров.
* Помогло в победе альянса: По мере того как союзники смогли расшифровывать зашифрованные сообщения с помощью усовершенствованных методов криптоанализа, они стали получать ценную информацию о планах и действиях нацистов. Это помогло альянсу в планировании контрмер и усилении преимущества в военных операциях.
* Влияние на послевоенную эпоху: Завершение Второй мировой войны не означало, что машина "Энигма" ушла в прошлое. Её технология и методы шифрования продолжили развиваться и оказали влияние на современную криптографию и информационную безопасность.

5. **Пусть p = 37 и α = 9. Смоделируйте обмен ключами с помощью алгоритма Диффи-Хеллмана.**

<!DOCTYPE html>

<html>

<head>

    <title>Диффи-Хеллман Обмен Ключами</title>

</head>

<body>

    <h1>Диффи-Хеллман Обмен Ключами</h1>

    <p>Параметры: p = 37, α = 9</p>

    <button onclick="denisStep1()">Денис: Выбрать секретное число (a)</button>

    <button onclick="nikolaiStep1()">Николай: Выбрать секретное число (b)</button>

    <button onclick="denisStep2()">Денис: Вычислить A и передать Николаю</button>

    <button onclick="nikolaiStep2()">Николай: Вычислить B и передать Денису</button>

    <button onclick="denisStep3()">Денис: Вычислить общий секретный ключ (K)</button>

    <button onclick="nikolaiStep3()">Николай: Вычислить общий секретный ключ (K)</button>

    <p id="denisLog"></p>

    <p id="nikolaiLog"></p>

    <p id="sharedKey"></p>

    <script>

        const p = 37;

        const alpha = 9;

        let a, A, b, B, sharedKey;

        function denisStep1() {

            a = getRandomInt(1, p - 1);

            A = modPow(alpha, a, p);

            addToLog("Денис: Выбрано секретное число a = " + a);

            addToLog("Денис: Вычислено A = α^a mod p = " + A);

        }

        function nikolaiStep1() {

            b = getRandomInt(1, p - 1);

            B = modPow(alpha, b, p);

            addToLog("Николай: Выбрано секретное число b = " + b);

            addToLog("Николай: Вычислено B = α^b mod p = " + B);

        }

        function denisStep2() {

            addToLog("Денис передает B (Николаю): " + B);

        }

        function nikolaiStep2() {

            addToLog("Николай передает A (Денису): " + A);

        }

        function denisStep3() {

            sharedKey = modPow(B, a, p);

            addToLog("Денис: Вычисляет общий секретный ключ K = B^a mod p = " + sharedKey);

            document.getElementById("sharedKey").textContent = "Общий секретный ключ (Денис): " + sharedKey;

        }

        function nikolaiStep3() {

            sharedKey = modPow(A, b, p);

            addToLog("Николай: Вычисляет общий секретный ключ K = A^b mod p = " + sharedKey);

            document.getElementById("sharedKey").textContent = "Общий секретный ключ (Николай): " + sharedKey;

        }

        function addToLog(message) {

            const denisLog = document.getElementById("denisLog");

            const nikolaiLog = document.getElementById("nikolaiLog");

            const logMessage = document.createElement("p");

            logMessage.textContent = message;

            denisLog.appendChild(logMessage);

            nikolaiLog.appendChild(logMessage.cloneNode(true));

        }

        function getRandomInt(min, max) {

            return Math.floor(Math.random() \* (max - min + 1)) + min;

        }

        function modPow(base, exponent, modulus) {

            if (modulus === 1) return 0;

            let result = 1;

            base = base % modulus;

            while (exponent > 0) {

                if (exponent % 2 === 1) {

                    result = (result \* base) % modulus;

                }

                exponent = Math.floor(exponent / 2);

                base = (base \* base) % modulus;

            }

            return result;

        }

    </script>

</body>

</html>

