Государственное образовательное учреждение

«Белорусский государственный технологический университет»

Отчет

*по лабораторной работе №13*

ТЕМА ЗАНЯТИЯ: ИЗУЧЕНИЕ СТАНДАРТНЫХ СРЕДСТВ ДЛЯ РЕАЛИЗАЦИИ СИММЕТРИЧНОГО И АССИМЕТРИЧНОГО ШИФРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ SUBTLECRYPTO В JS

|  |  |
| --- | --- |
|  | **Исполнитель:**  Студента (-ка) группы 2  Специальность\_\_\_\_ИСиТ\_\_\_  Гулевич Никита Максимович  (Ф.И.О.) |
|  | **Преподаватель**:  Ржеутская Н. В. |

Минск, 2025

**Цель**: изучить интерфейс SubtleCrypto, основные классы и структуры данных: разработать приложение для шифрования файлов, использующих симметричные и ассиметричные алгоритмы шифрования

ДЛЯ ВАРИАНТОВ 5 и 7

<!DOCTYPE html>

<html lang="ru">

<head>

  <meta charset="UTF-8">

  <title>Crypto Variants</title>

  <style>

    body {

      font-family: 'Segoe UI', sans-serif;

      background: #1e1e2f;

      color: #f0f0f0;

      padding: 40px;

    }

    h1 {

      color: #4fd1c5;

    }

    button {

      padding: 10px 20px;

      margin: 10px 10px 20px 0;

      background-color: #4fd1c5;

      border: none;

      border-radius: 8px;

      color: #1e1e2f;

      font-weight: bold;

      cursor: pointer;

      transition: background 0.3s;

    }

    button:hover {

      background-color: #38b2ac;

    }

    pre {

      background: #2d2d3a;

      padding: 20px;

      border-radius: 10px;

      white-space: pre-wrap;

    }

  </style>

</head>

<body>

  <h1>Шифрование фамилии: Гулевич</h1>

  <button onclick="runVariant7()">Вариант 7 (AES-GCM + SHA384)</button>

  <button onclick="runVariant5()">Вариант 5 (AES-CTR + SHA1)</button>

  <pre id="output"></pre>

  <script>

    const output = document.getElementById('output');

    const surname = "Гулевич";

    // Вариант 7: AES-GCM + SHA-384

    async function runVariant7() {

      output.textContent = "🔐 Вариант 7: AES-GCM + SHA-384\n\n";

      const enc = new TextEncoder();

      const dec = new TextDecoder();

      // Генерация ключа AES-GCM (256 бит)

      const key = await crypto.subtle.generateKey(

        { name: "AES-GCM", length: 256 },

        true,

        ["encrypt", "decrypt"]

      );

      // Случайный вектор инициализации длиной 12 байт

      const iv = crypto.getRandomValues(new Uint8Array(12));

      // Шифруем фамилию

      const ciphertext = await crypto.subtle.encrypt(

        { name: "AES-GCM", iv: iv },

        key,

        enc.encode(surname)

      );

      // Расшифровываем обратно

      const decrypted = await crypto.subtle.decrypt(

        { name: "AES-GCM", iv: iv },

        key,

        ciphertext

      );

      // Хешируем через SHA-384

      const hashBuffer = await crypto.subtle.digest("SHA-384", enc.encode(surname));

      const hashHex = [...new Uint8Array(hashBuffer)].map(b => b.toString(16).padStart(2, '0')).join('');

      // Выводим результат

      output.textContent += `Исходная фамилия: ${surname}\n`;

      output.textContent += `Случайный IV (hex): ${Array.from(iv).map(b => b.toString(16).padStart(2, '0')).join('')}\n`;

      output.textContent += `Зашифровано (Base64): ${btoa(String.fromCharCode(...new Uint8Array(ciphertext)))}\n`;

      output.textContent += `Расшифровано: ${dec.decode(decrypted)}\n`;

      output.textContent += `SHA-384 хеш: ${hashHex}\n`;

    }

    // Вариант 5: AES-CTR + SHA-1

    async function runVariant5() {

      output.textContent = "🔐 Вариант 5: AES-CTR + SHA-1\n\n";

      const enc = new TextEncoder();

      const dec = new TextDecoder();

      // Генерация ключа AES-CTR (256 бит)

      const key = await crypto.subtle.generateKey(

        { name: "AES-CTR", length: 256 },

        true,

        ["encrypt", "decrypt"]

      );

      // Счетчик (nonce), обязательно 16 байт

      const counter = crypto.getRandomValues(new Uint8Array(16));

      // Шифруем фамилию

      const ciphertext = await crypto.subtle.encrypt(

        { name: "AES-CTR", counter: counter, length: 64 },

        key,

        enc.encode(surname)

      );

      // Расшифровка

      const decrypted = await crypto.subtle.decrypt(

        { name: "AES-CTR", counter: counter, length: 64 },

        key,

        ciphertext

      );

      // Хеширование SHA-1

      const hashBuffer = await crypto.subtle.digest("SHA-1", enc.encode(surname));

      const hashHex = [...new Uint8Array(hashBuffer)].map(b => b.toString(16).padStart(2, '0')).join('');

      // Выводим результат

      output.textContent += `Исходная фамилия: ${surname}\n`;

      output.textContent += `Случайный Counter (hex): ${Array.from(counter).map(b => b.toString(16).padStart(2, '0')).join('')}\n`;

      output.textContent += `Зашифровано (Base64): ${btoa(String.fromCharCode(...new Uint8Array(ciphertext)))}\n`;

      output.textContent += `Расшифровано: ${dec.decode(decrypted)}\n`;

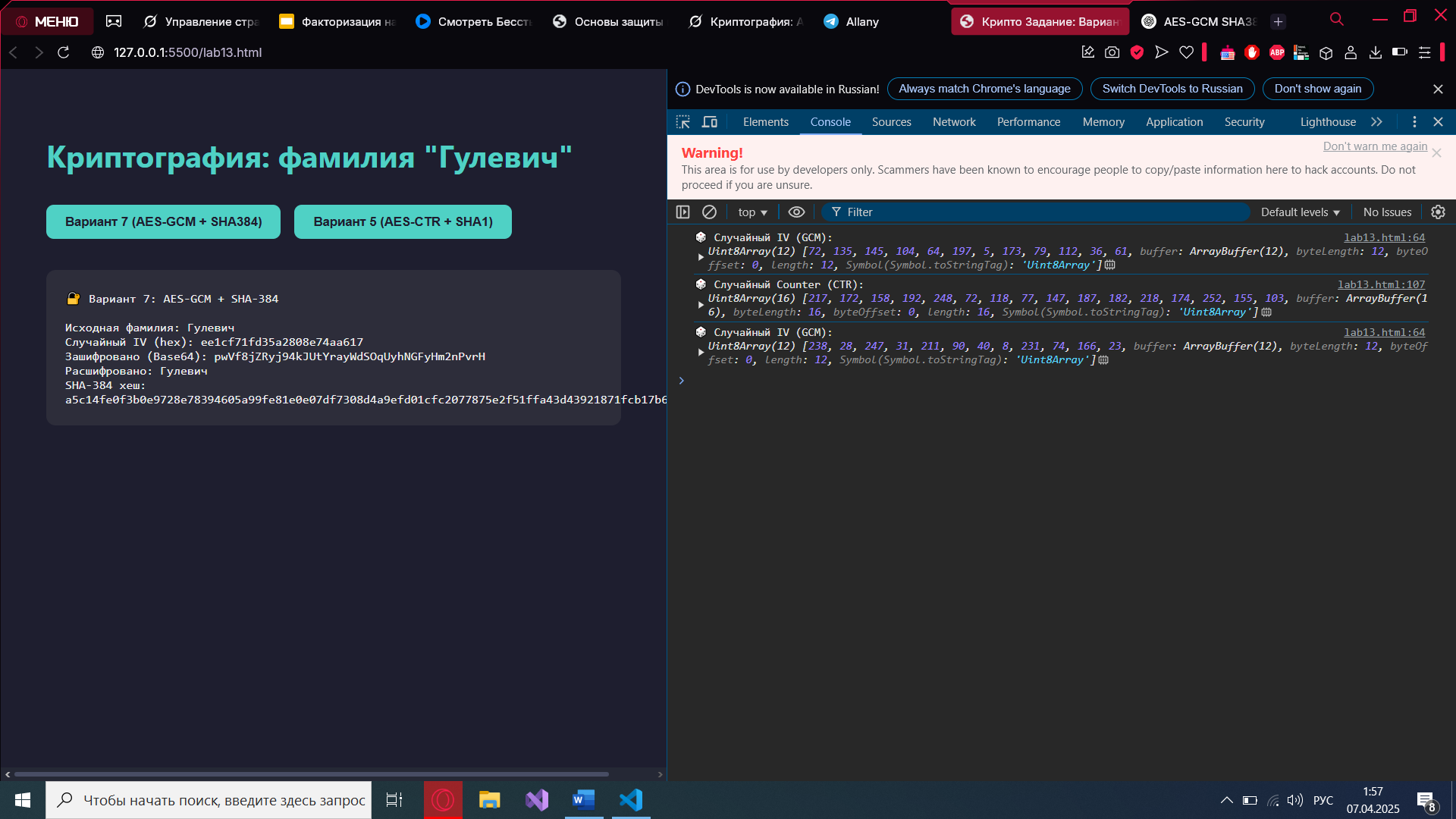
      output.textContent += `SHA-1 хеш: ${hashHex}\n`;

    }

  </script>

</body>

</html>



**Список использованных источников**

1. Ржеутская Н. В., Нистюк О. А., Уласевич Н. И. Основы защиты информации: лабораторный практикум. Минск: Учреждение образования «Белорусский государственный технологический университет», 2024.