Минобрнауки России

федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный технологический институт (технический университет)»

СПбГТИ(ТУ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Направление подготовки | | 09.03.01 | Информатика и вычислительная техника | | | |
| Направленность программы | |  | Автоматизированные системы обработки информации и управления  Очная | | | |
|  | |  |  | | | |
| Факультет | |  | Информационных технологий и управления | | | |
| Кафедра | |  | Систем автоматизированного  проектирования и управления | | | |
| Учебная дисциплина | |  | Интернет-технологии | | | |
|  |  | | |  |  |

Курс 3 Группа 404

**Отчет по контрольной работе № 1**

|  |  |
| --- | --- |
| **Тема:** | **Создание интерфейса веб-сайта**  **с использованием HTML и CSS** |

Студент гр. 404 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. К. Азаров

(подпись, дата) (инициалы, фамилия)

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Иванов

(подпись, дата) (инициалы, фамилия)

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись преподавателя)

Санкт-Петербург

2023

#### **1. Описание идеи проекта. Формирование функциональных требований к проекту**

Шифрование – обратимое преобразование [информации](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F) в целях сокрытия от [неавторизованных](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) лиц с предоставлением в это же время [авторизованным](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) пользователям доступа к ней. Главным образом, шифрование служит для соблюдения [конфиденциальности](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%84%D0%B8%D0%B4%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C) передаваемой информации. Важной особенностью любого алгоритма шифрования является использование [ключа](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F)), который утверждает выбор конкретного преобразования из совокупности возможных для данного алгоритма.

Пользователи являются авторизованными, если они обладают определённым [аутентичным](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) [ключом](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F)). Вся сложность и, собственно, задача шифрования состоит в том, как именно реализован этот процесс. Шифрование состоит из двух составляющих: зашифрование и расшифрование.

С помощью шифрования обеспечиваются три состояния безопасности информации:

* Конфиденциальность – шифрование используется для скрытия информации от неавторизованных пользователей при передаче или при хранении.
* Целостность – шифрование используется для предотвращения изменения информации при передаче или хранении.
* Идентифицируемость – шифрование используется для аутентификации источника информации и предотвращения отказа отправителя информации от того факта, что данные были отправлены именно им.

Для того, чтобы прочитать зашифрованную информацию, принимающей стороне необходимы [ключ](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F)) и [дешифратор](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%88%D0%B8%D1%84%D1%80%D0%B0%D1%82%D0%BE%D1%80) (устройство, реализующее алгоритм расшифровывания). Идея шифрования состоит в том, что злоумышленник, перехватив зашифрованные данные и не имея к ним ключа, не может ни прочитать, ни изменить передаваемую информацию. Кроме того, в современных [криптосистемах (с открытым ключом)](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81_%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%BC_%D0%BA%D0%BB%D1%8E%D1%87%D0%BE%D0%BC) для шифрования, расшифрования данных могут использоваться разные ключи. Однако, с развитием криптоанализа, появились методики, позволяющие дешифровать закрытый текст без ключа. Они основаны на математическом анализе переданных данных [1].

**Цель проекта:** разработать интерфейса веб-приложения для шифрования текста.

**Функциональные требования к разрабатываемому веб-приложению:**

1) Отображать алгоритмы.

2) Возможность ввода ключа.

3) Возможность Ввода исходного текста.

4) Отображать описание алгоритма.

#### **2 Обзор и обоснование выбора технологий вёрстки макетов сайта**

**Верстание** — формирование структуры будущего сайта в HTML-документе и перенос дизайнерского наброска в такой документ. Перенос макета в веб-страницу происходит благодаря использованию HTML, CSS, JavaScript и клиентских сценариев. Цель верстки сайта — сделать так, чтобы макет был перенесен в реальный сайт максимально точно и близко дизайнерской задумке, с сохранением расположения текста, заголовков, изображений, других медиафайлов, таблиц [[2]](#_СПИСОК_ЛИТЕРАТУРЫ).

Говоря о верстании веб-страниц, нельзя не упомянуть конкретные методы, которые используются для переноса дизайн-макета в документ. Не стоит путать методы с видами — это разные термины. Итак, в 2023 году активно используются четыре метода верстания страниц: фреймы, табличный способ, блочный способ (div-контейнерами) и слои. Каждая техника имеет свой набор достоинств / недостатков. Давайте посмотрим, какие именно.

1. **Верстание фреймами** — убирает повторы в коде и делает итоговый файл гораздо легче по размеру, но такие страницы плохо индексируется ПС (ведь они не имеют уникального адреса) [[2]](#_СПИСОК_ЛИТЕРАТУРЫ).

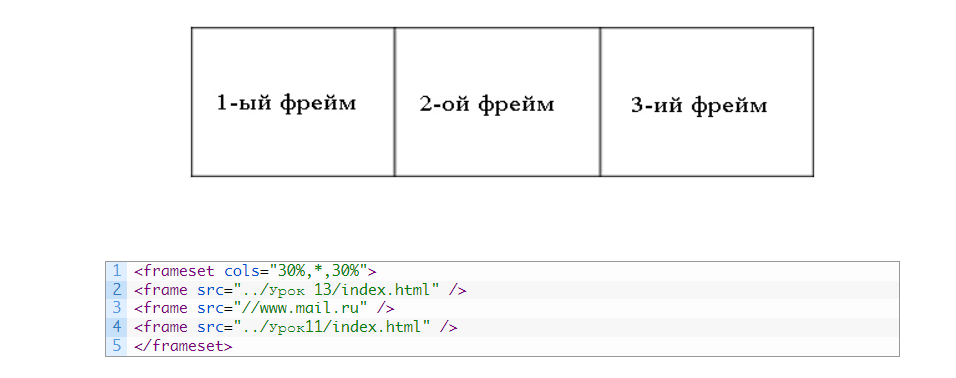


Рисунок 2 – Фреймовая верстка

1. **Табличный метод** верстания долгое время был стандартом в вебе, но сегодня — он активно сдает позиции (ведь этот метод понижает скорость загрузки сайта, ухудшает индексацию, да и сам код, в итоге, выглядит не оптимизированным, так как предусматривает большое количество пустых таблиц).

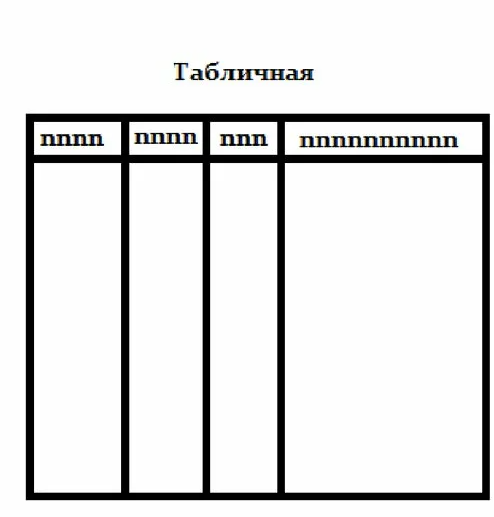


Рисунок 3 – Табличная верстка

1. **Блочный метод верстания** — считается одним из лучших, однако, он ухудшает фактор кроссбраузерности. Зато именно блочный метод предусматривает минимальный размер кода, его высокую читабельности. Именно блочный метод верстания страниц развязывает руки дизайнерам — можно внедрить любое, даже сложное решение. Также сайты с блочной версткой лучше всего индексируется, по сравнению с теми, которые используют иные методы верстания страниц [[2]](#_СПИСОК_ЛИТЕРАТУРЫ).

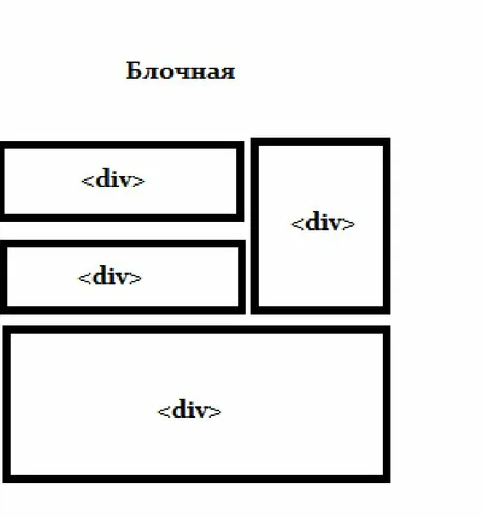


Рисунок 4 – Блочная верстка

Таким образом, для разработки веб-приложения был использован блочный метод верстания.

**3 Разработка макета графического интерфейса сайта**

Разрабатываемое веб-приложение будет состоять из заголовка с алгоритмами шифрования и страницами с исходным текстом, текстом результата, ключем и описание алгоритма. На рисунке 5 представлен схематичный макет страницы веб-приложения пользовательского интерфейса при открытой вкладки страниц.

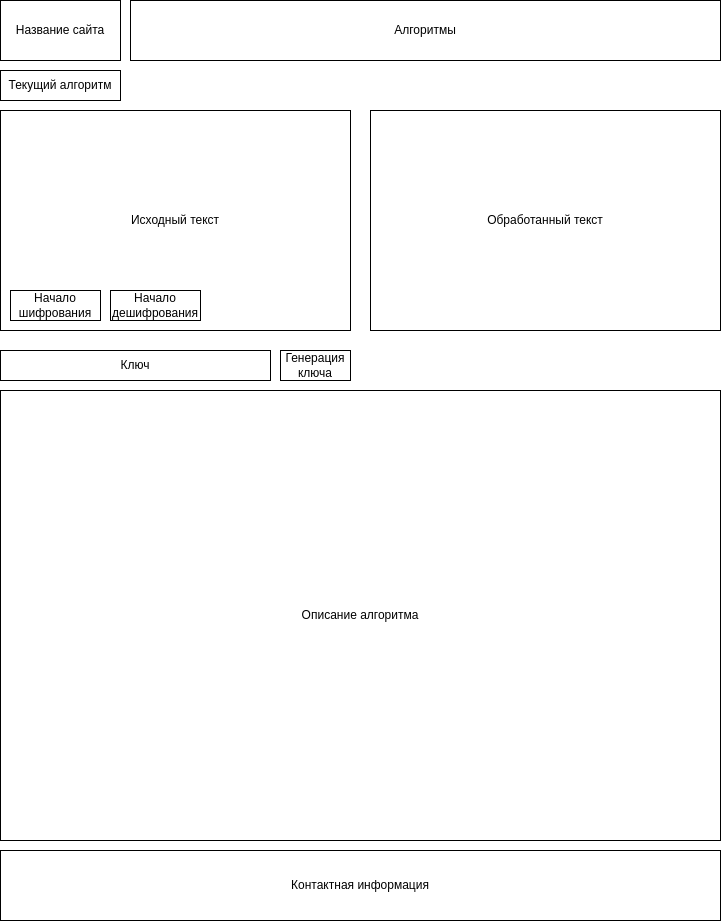


Рисунок 5 – Содержимое вкладки страниц с рецептами

#### **4 Реализация с использованием HTML и CSS макета графического интерфейса сайта**

Ниже представлены детали реализации основного макета страницы, состоящего из заголовка, основного контента и подвала графического интерфейса на HTML и CSS.

**HTML:**

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="utf-8">

<title>Ciphers</title>

<link rel="stylesheet" href="styles.css">

<script src="script.js">script.js</script>

</head>

<body>

<header>

<nav>

<img id="logo" src="src/logo.jpg" alt="Шифр">

<ul id="topmenu">

<li>AES</li>

<li>BLOWFISH</li>

<li>CAST</li>

<li>DES</li>

</ul>

</nav>

</header>

<div id="page">

<h2 id="cipher\_name">AES</h2>

<div class="wrapper">

<div>Исходный текст</div>

<div>Преобразованный текст</div>

<div class="text\_box"><textarea id="in\_text"></textarea></div>

<div class="text\_box"><textarea id="out\_text" readonly></textarea></div>

<div id="ciph\_btns\_box">

<button id="ciph\_btn" type="button">Кодировать</button>

<button id="deciph\_btn" type="button">Декодировать</button>

</div>

<div></div>

<div>

<input type="text" size="40" id="key\_box" placeholder="Ключ" autocomplete="off" required>

<button type="button" id="generate\_key">Сгенерировать</button>

</div>

<div></div>

</div>

<article>

<h3>Алгоритм</h3>

<div id="AES-alghoritm" class="show">

<img src="src/aes-alghoritm.jpg" alt="Шифр AES">

<ol>

<li>Сообщение разбивается на блоки одинакового размера. При необходимости последний блок дополняется до стандартной длины 128 бит.</li>

<li>Шифрование очередного блока сообщения выполняется с использованием предыдущего зашифрованного блока.

Для первого блока предыдущего зашифрованного блока не существует, поэтому его шифруют с использованием «вектора инициализации» (<strong>initialization vector</strong>, <strong>IV</strong>).</li>

<li>Вектор инициализации — случайное число. Размер (длина) IV равна размеру блока.</li>

<li>Расшифровка выполняется с использованием тех же ключа k и вектора инициализации IV.</li>

</ol>

<div class="clear\_floating"></div>

</div>

<div id="BLOWFISH-alghoritm" class="hide">

<img src="src/blowfish-alghoritm.png" alt="Шифр BLOWFISH">

<ol>

<li>Исходные данные разбиваются на блоки фиксированной длины (как правило кратно степени двойки — 64 бит, 128 бит). В случае если длина блока исходных данных меньше длины разрядности шифра, то блок дополняется каким-либо заранее известным образом.</li>

<li>Блок делится на два равных подблока — «левый» L<sub>0</sub> и «правый» R<sub>0</sub>. В случае 64-битной разрядности — на два блока с длиной 32 бита каждый.</li>

<li>«Левый подблок» L<sub>0</sub> видоизменяется функцией итерации <i>F</i>(L<sub>0</sub>, P<sub>0</sub>) в зависимости от ключа P<sub>0</sub>, после чего он складывается по модулю 2 (XOR) с «правым подблоком» R<sub>0</sub>.</li>

<li>Результат сложения присваивается новому левому подблоку L<sub>1</sub>, который становится левой половиной входных данных для следующего раунда, а «левый подблок» L<sub>0</sub> присваивается без изменений новому правому подблоку R<sub>1</sub>, который становится правой половиной.</li>

<li>Эта операция повторяется n-1 раз, при этом при переходе от одного этапа к другому меняются раундовые ключи (P<sub>0</sub>, P<sub>1</sub>, P<sub>2</sub> и т.д.), где n — количество раундов для используемого алгоритма.</li>

</ol>

<div class="clear\_floating"></div>

</div>

<div id="CAST-alghoritm" class="hide">

<img src="src/cast-alghoritm.webp" alt="Шифр CAST">

<ol>

<li>(развертка ключа) составляет 16 пар подключей {Kmi, Kri} полученных из K (см. разделы Пары раундовых ключей и Неидентичные раунды).</li>

<li>(L0, R0) <- (m1. .. m64). (Разделяет текст на левую и правую 32-битные половины L0 = m1 ... m32 и R0 = m33 ... m64).</li>

<li>(16 раундов) for i from 1 to 16, вычислить Li и Ri следующим образом: Li = Ri-1; Ri = Li-1 ^ F(Ri-1,Kmi,Kri), где F определена в разделе «Пары раундовых ключей» (F имеет тип 1, тип 2, тип 3 или, в зависимости от i).</li>

<li>c1 ... c64 <- (R16, L16). (Меняем окончательные блоки местами L16, R16 и объединяем, чтобы сформировать зашифрованный текст.)</li>

</ol>

<div class="clear\_floating"></div>

</div>

<div id="DES-alghoritm" class="hide">

<img src="src/des-alghoritm.png" alt="Шифр DES">

<ol>

<li>довести исходное сообщение до такого размера (в битах), чтобы оно нацело делилось на размер блока (sizeOfBlock = 128 бит);</li>

<li>разделить исходное сообщение на блоки;</li>

<li>довести длину ключа до длины половины блока;</li>

<li>перевести ключ в бинарный формат (в нули и единицы);</li>

<li>провести над каждым блоком прямое преобразование сетью Фейстеля в течении 16-ти раундов. После каждого раунда необходимо выполнять циклический сдвиг ключа на заданное количество символов;</li>

<li>соединить все блоки вместе; таким образом получим сообщение, зашифрованное алгоритмом DES.</li>

</ol>

<div class="clear\_floating"></div>

</div>

</article>

</div>

<footer>

<p>Copyright © 2023, Азаров Даниил</p>

</footer>

</body>

</html>

**CSS:**

body {

margin-left: 20%;

margin-right: 20%;

background-color: #111f24;

}

button {

/\* width:90px; \*/

text-align:center;

/\* display:block; \*/

font-family: arial;

text-decoration: none;

font-weight: 300;

font-size: 14px;

border: #0B1F24 1px solid;

color: #0B1F24;

padding: 3px;

padding-left: 5px;

padding-right: 5px;

/\* margin: 20px auto; \*/

transition: .5s;

border-radius: 0px;

}

button:hover {

top: 5px;

transition: .5s;

color: #049bf3;

border: #049bf3 1px solid;

border-radius: 10px;

}

button:active {

color: #0066ff;

border: #0066ff 1px solid;

transition: .07s;

background-color: #FFF;

}

/\* #main { \*/

/\* min-width: max-content; \*/

/\* } \*/

header > nav {

display: flex;

flex-direction: row;

align-items: center;

}

#logo {

display: inline-block;

width: 200px;

}

#topmenu li

{

margin: 10px;

display: inline-block;

text-decoration: none;

color: #fafdff;

font-size: 30px;

}

#page

{

background-color: white;

padding: 20px;

margin-top: 0px;

}

#ciph\_btns\_box

{

margin-top: -30px;

margin-right: 5px;

float: right;

}

.wrapper

{

display: grid;

grid-template-columns: 1fr 1fr;

grid-template-rows: 1fr 15fr;

align-items: center;

gap:5px 20px;

}

.text\_box

{

height: 100%;

width: 100%;

}

textarea

{

height: 100%;

width: 100%;

resize: none;

}

.ciph\_box

{

margin-top: -30px;

}

img

{

float: left;

margin: 20px;

}

.clear\_floating

{

clear: both;

}

footer

{

color: #fafdff;

}

.hide

{

display: none;

}

На рисунке 6 представлено отображение в браузере разработанного макета графического интерфейса.

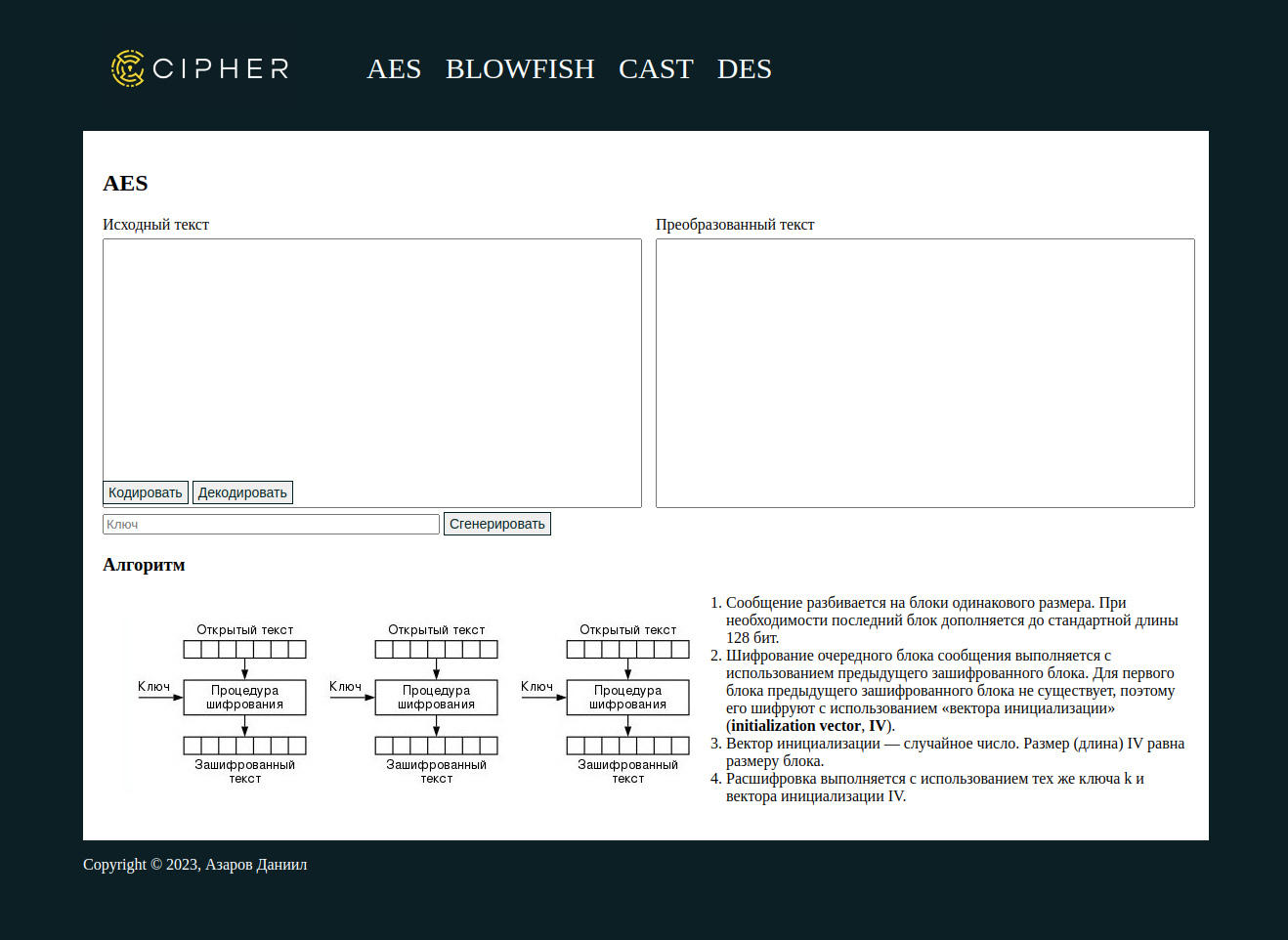


Рисунок 6 - Разработанный графический интерфейс страницы

# **СПИСОК ЛИТЕРАТУРЫ**

* 1. Э. Мэйволд. Безопасность сетей. — 2006. — 528 с. — [ISBN 978-5-9570-0046-9](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/9785957000469)
  2. LoftSchool: Методы верстки сайтов: сайт. – Москва, 2022 –.– URL: <https://loftschool.com/blog/posts/verstka-sajta-chto-eto#verstka4> (Дата обращения: 15.03.2023).