Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Отчет**

«Исследование стеганографического метода на основе

преобразования наименее значащих бит»

Выполнил:

студент 3 курса 4 группы

специальности ПОИТ

Чернявский А. Л.

Минск 2020

*Определение* **1.** **Стеганографическая система** (stegosystem, стегосистема или стеганосистема – в русскоязычной тематической литературе используются оба сокращения) – совокупность средств и методов, которые используются для формирования скрытого канала передачи (или хранения) информации. При этом скрытый канал организуется на базе и внутри открытого канала с использованием особенностей восприятия информации. «Скрытость» канала передачи тайной информации отличает стеганографии от криптографии: в первом случае тайной является сам факт наличия канала (передачи информации).

*Определение* 2. Абстрактно стеганографическая система обычно определяется, как некоторое множество отображений одного пространства (множества возможных сообщений, М) в другое пространство (множество возможных стеганосообщений, S, и наоборот.

**Основные компоненты стеганосистемы**:

**контейнер, С** (файл-контейнер или электронный документ произвольного формата), в котором размещается (осаждается, скрывается) тайное сообщение, М; именно контейнер является упомянутым скрытым каналом;

**тайное сообщение, М**, осаждаемое в контейнер для передачи или хранения (например, с целью доказательства или защиты авторских прав на документ-контейнер [2, 53-56]; здесь речь может идти о невидимых цифровых водяных знаках, ЦВЗ);

**ключи или ключевая информация, K** системы, выполняющие ту же функцию, что и криптографические ключи; ключей может быть несколько, в соответствии с этим современные стеганосистемы характеризуют как многоключевые: один ключ отождествляется с методом осаждения/извлечения тайной информации, другой – с выбором элементов (например, битов) контейнера для его модификации при осаждении тайной информации, третий (или третьи) – для предварительного (перед осаждением) преобразования тайной информации (например, на основе помехоустойчивого кодирования, сжатия или зашифрования) и т. д. [2, 57, 58]; контейнер с осажденным сообщением или стеганоконтейнер, S, который передается по открытому каналу, также являющемуся важным компонентом анализируемой системы; стеганоконтейнер будем именовать также стеганосообщением;

для полноты упомянем также **субъектов системы: отправителя и получателя**.

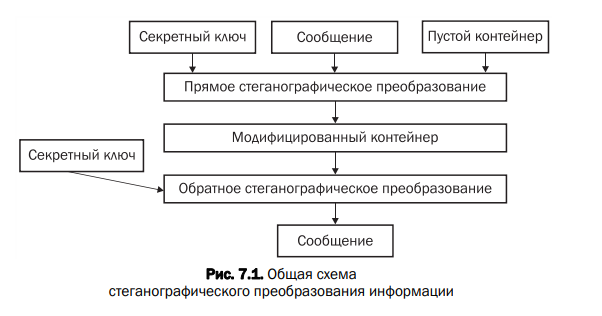
В зависимости от формата документа-контейнера цифровую (или компьютерную) стеганографию подразделяют на классы [2, 52, 59-64]:

* *аудиостеганография,*
* *видеостеганография,*
* *графическая стеганография,*
* *текстовая стеганография*

и др.

*Определение* 3. **Стеганографической** **системой** ∑ будем называть совокупность сообщений M, контейнеров C, ключей K, стеганосообщений (заполненных контейнеров) S и преобразований (прямого F и обратного F

1), которые их связывают: ∑ = (M, C, K, S, F, F)



1. Перед выполнением основного задания целесообразно

познакомится со структурой, интерфейсом и функциональными особенностями доступных и заслуживающих внимания приложений, в

которых реализован метод НЗБ. К ним можно отнести следующие:

* openstego (https://github.com/syvaidya/openstego) – может применяться

не только для осаждения данных, но и для ЦВЗ; использует RandomLSB – псевдослучайный принцип осаждения; поддерживает

шифрование (дополнительный ключ), имеет также GUI; осаждение реализуется командой openstego embed-mf secret.txt -cf cover.png -p password -sfstego.png, извлечение – openstego extract -sf

openstego.png -p abcd -xf output.txt; как видим, работает с PNGконтейнерами;

* Stegano ( https :// github. com / cedricbonhomme / Stegano) – работает не

только с классическим LSB; имеет гибкую настройку, может также использоваться как модуль Python; осаждение реализуется командой stegano-lsb hide --input cover.jpg-f secret.txt -e UTF-8 --output stego.png , извлечение – stegano-lsb reveal -i stego.png -e UTF8 -o output.txt ; работает также с PNG-контейнерами;

* LSB-Steganography ( https :// github. com / RobinDavid / LSB - Steganography) –

приложение написано на Python; работает с PNG- и ВМР-контейнерами.

2. Разработать собственное приложение, в котором должен быть реализован метод НЗБ. При этом:

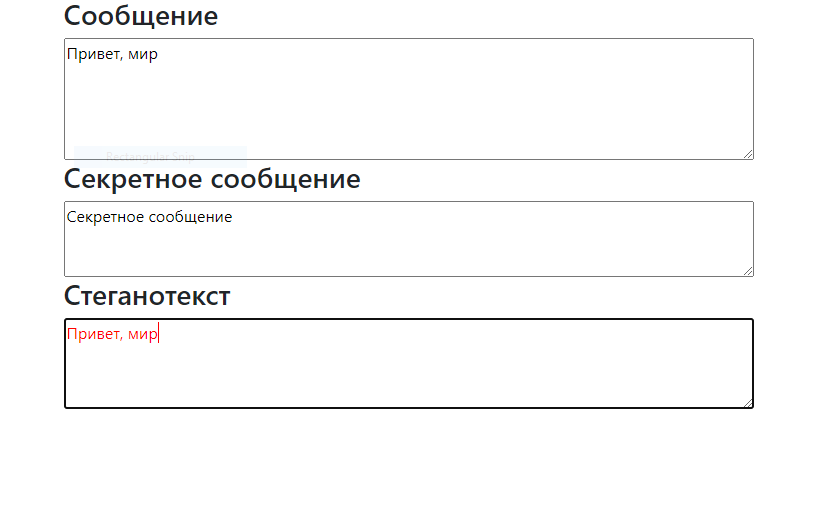
* выбор файла-контейнера – по согласованию с преподавателем;
* реализовать два варианта осаждаемого/извлекаемого сообщения:
* собственные фамилия, имя и отчество,
* текстовая часть отчета по одной из выполненных лабораторных работ;
* реализовать два метода (на собственный выбор) размещения битового потока осаждаемого сообщения по содержимому контейнера;

- сформировать цветовые матрицы (по аналогии с рис. 12.7), отображающие каждый задействованный для осаждения уровень младших значащих бит контейнера;

* выполнить визуальный анализ (с привлечением коллег в качестве экспертов) стеганоконтейнеров с различным внутренним содержанием; сделать выводы на основе выполненного анализа.

3. Результаты выполнения работы оформить в виде отчета по установленным правилам.

**Результат работы программы**:



**Листинги**:

StartFunc**.**js

|  |
| --- |
| function plainChange() {  let plain = document.getElementById("plain");  let message = document.getElementById("message");  let stego = document.getElementById("stego");  stego.value = encode(plain.value, tobits(message.value));  confirm();  }  function messageChange() {  let plain = document.getElementById("plain");  let message = document.getElementById("message");  let stego = document.getElementById("stego");  stego.value = encode(plain.value, tobits(message.value));  confirm();  }  function stegoChange() {  let plain = document.getElementById("plain");  let message = document.getElementById("message");  let stego = document.getElementById("stego");  message.value = frombits(decode(stego.value));  plain.value = sanitize(stego.value);  confirm();  }  function confirm() {  let message = document.getElementById("message");  let stego = document.getElementById("stego");  if (message.value.localeCompare(frombits(decode(stego.value))))  stego.style.color = 'red';  else  stego.style.color = 'black';  } |

StegoFunc.js

|  |
| --- |
| let bit0 = '\u0020' //space  let bit1 = '\u00A0' //NO-BREAK space  // Конвертир. из 8бит в 10бит сроку  function tobits(data) {  let bits = '';  for (let i = 0; i < data.length; i++) {  let c = data.charCodeAt(i);  bits += bit1;  bits += (c & 128) ? bit1 : bit0;  bits += (c & 64) ? bit1 : bit0;  bits += (c & 32) ? bit1 : bit0;  bits += (c & 16) ? bit1 : bit0;  bits += (c & 8) ? bit1 : bit0;  bits += (c & 4) ? bit1 : bit0;  bits += (c & 2) ? bit1 : bit0;  bits += (c & 1) ? bit1 : bit0;  bits += bit0;  }  return bits  }  function checksync(bits, i, n) {  for (let j = i; j < i + 10 \* n; j += 10) {  if (j + 9 < bits.length) {  if (bits[j + 0] != bit1) return false;  if (bits[j + 9] != bit0) return false;  }  }  return true  }  // проверка синхронизации  function frombits(bits) {  let insync = checksync(bits, 0, 1);  let data = '';  console.log(bits.length)  for (let i = 0; i < bits.length; ) {  if (insync) {  if (checksync(bits, i, 1)) {  let c = 0;  if (bits[i + 1] == bit1) c |= 128  if (bits[i + 2] == bit1) c |= 64  if (bits[i + 3] == bit1) c |= 32  if (bits[i + 4] == bit1) c |= 16  if (bits[i + 5] == bit1) c |= 8  if (bits[i + 6] == bit1) c |= 4  if (bits[i + 7] == bit1) c |= 2  if (bits[i + 8] == bit1) c |= 1  data += String.fromCharCode(c);  console.log('|'+data+'|')  i += 10;  }  else {  insync = false;  }  }  else {  if (checksync(bits, i, 4))  insync = true;  else  i += 1;  }  }  return data;  }  function istext(c) {  return (c != '\u0020' && c != '\u00A0' && c != '\t');  }  // проверка на конец строки  function iseol(c) {  return (c == '\n');  }  // Декодирование  function decode(data) {  let intext = false;  let bits = '';  for (let i = 0; i < data.length; i++) {  if (intext) {  if (data[i] == bit0)  bits += bit0;  if (data[i] == bit1)  bits += bit1;  console.log('bit'+bits + '| |' + bit1 + '| |' + bit0)  }  if (istext(data[i])) intext = true;  if (iseol (data[i])) intext = false;  }  return bits;  }  // Зашифрование пустыми пробелами  function encode(text, bits) {  let intext = false;  let data = '';  let j = 0;  for (let i = 0; i < text.length; i++) {  if (intext && j < bits.length && (text[i] == bit0 || text[i] == bit1))  data += bits[j++];  else  data += text[i];  if (istext(text[i])) intext = true;  if (iseol (text[i])) intext = false;  }  return data;  }  // Удаление белых маркеров  function sanitize(data) {  return data.replace(/\u00A0/g, '\u0020');  } |

index.html

|  |
| --- |
| <meta charset="utf-8">  <html>  <head>  <script src="StartFunc.js"></script>  <script src="StegoFunc.js"></script>  <style>  body {  font-family: Calibri, sans-serif;  margin: 3em;  }  textarea {  width: 100%;  }  p.author {  color: #CCC;  text-align: right;  }  </style>  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI" crossorigin="anonymous"></script>  </head>  <body>  <div class="container">  <h3>Сообщение</h3>  <textarea id="plain" rows="10" cols="80" oninput="plainChange()"></textarea>  <h3>Секретное сообщение</h3>  <textarea id="message" rows="3" cols="80" oninput="messageChange()"></textarea>  <h3>Стеганотекст</h3>  <textarea id="stego" rows="10" cols="80" oninput="stegoChange()"></textarea>  </div>  </body>  </html> |