Федеральное государственное бюджетное образовательное учреждение

высшего образования

**Национальный исследовательский университет «МЭИ»**

**Кафедра вычислительных машин, систем и сетей**

**КУРСОВАЯ РАБОТА**

На тему: «Разработка биометрического идентификатора с

электронно-цифровой подписью»

по дисциплине «Цифровые технологии защиты информации»

|  |  |
| --- | --- |
| Выполнил |  |
| Студент | Синегубкин А.С. |
| Группа | А-12м-19 |

|  |  |
| --- | --- |
| Принял |  |
| Преподаватель | Рытов А.А. |

Москва, 2019 г.

Оглавление

[Введение 3](#_Toc27555689)

[Рабочее задание 4](#_Toc27555690)

[Часть 1. 4](#_Toc27555691)

[Часть 2. 4](#_Toc27555692)

[Выполнение работы 6](#_Toc27555693)

[Исходные данные. 6](#_Toc27555694)

[Часть 1. Поиск точки, удовлетворяющей условию. 6](#_Toc27555695)

[Проверка эллиптической кривой на гладкость: 9](#_Toc27555696)

[Часть 2. Реализация модуля электронно-цифровой подписи. 10](#_Toc27555697)

[2.1 Встраивание идентификационной информации 12](#_Toc27555698)

[Заключение 17](#_Toc27555699)

[СПИСОК ЛИТЕРАТУРЫ 18](#_Toc27555700)

# Введение

При выполнении данной курсовой работы необходимо было выполнить две задачи:

1. С помощью стеганографических методов встроить данные, содержащиеся в студенческом билете, в фотографию.
2. Создать электронно-цифровую подпись (ЭЦП) цифрового объекта.

В первой части работы выполнена разработка программного модуля ЭЦП. На основе данных задания была найдена базовая точка, принадлежащая данной эллиптической кривой, соответствующая указанным параметрам. Был разработан программный модуль реализующий алгоритм ЭЦП ECDSA и хэш-функцию MD5. В заключении первой части модуль прошел верификацию на текстовом файле, содержащем данные из студенческого билета.

Во второй части работы осуществлено встраивание идентификационной информации в изображение. Был подготовлен контейнер-оригинал –цифровая фотография с заданными параметрами. Была проведена предварительная подготовка встраиваемой информации (содержание студенческого билета). Был разработан программный модуль для встраивания ЦВЗ в соответствии с методом блочного сокрытия и указанным стего-путем. В заключении второй части модуль прошел верификацию на основе подготовленных данных.

# Рабочее задание

## Часть 1.

Разработка программного модуля ЭЦП на основе эллиптической кривой (ЭК).

Задана эллиптическая кривая y2 = x3 + ax2 + bx + (c + i) над полем GF(p). Коэффициент а=0 для всех вариантов, переменная i – целое 0,

стартовое значение i=0.

Остальные параметры кривой приведены в Табл. 1.

Номер варианта определяется следующим образом: N=9 + Nгр=19,

где Nгр – номер по списку в группе.

1. Найти базовую точку P(x,y), принадлежащую ЭК, со следующими параметрами:
   * x – координата задается из условия x Floor[p/2];
   * порядок точки должен быть простым числом, б*о*льшим 2p/3;
   * поиск точки может быть проведен с помощью изменения (инкремента) переменной "i";
   * эллиптическая кривая должна быть гладкой.
2. Разработать программный модуль, реализующий алгоритм ЭЦП и хэш- функцию, согласно варианту в Табл. 1.
3. Провести верификацию программного модуля ЭЦП на текстовом файле, содержащем данные из студенческого билета.

Таблица 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | b | c + i | p | Алгоритм | Хэш-функция |
| 19 | 23 | 7+i |  | ECDSA | "MD5" |

## Часть 2.

Встраивание идентификационной информации (основных элементов содержимого студенческого билета) в фиксированный контейнер – фотографию студента.

1. Подготовить контейнер-оригинал, цифровую фотографию с размерами, указанными в Табл. 2.
2. Провести предварительную подготовку встраиваемой информации из Табл. 3, для чего перевести все символы ЦВЗ в двоичную форму и выровнять цифры до размера букв – 11бит/символ с использованием функции PadLeft[\*\*\*,11], сформировать битовую строку предназначенную для встраивания в фотографию.
3. Определить цветовой канал для встраивания информации по следующему правилу: Nmod3= 0 - > R; 1 -> G; 2 -> B. Nmod3=1.
4. Разработать программный модуль для встраивания ЦВЗ в соответствии с методом и стего-путем, приведенными в Табл. 2.
5. Провести верификацию ЦВЗ путем восстановления встроенных данных по соответствующему алгоритму.
6. Создать ЭЦП для модифицированного контейнера со встроенным ЦВЗ с использованием функции FileHash[].

Таблица 2. Варианты реализации методов встраивания.

|  |  |
| --- | --- |
| Вариант, Nmod4+1 | 4 |
| Размер фотографии,  ширина х высота | 240х320 |
| Размер блока, точек | 4х4 |
| Метод встраивания  ЦВЗ | Метод блочного сокрытия |
| Стего путь | Последовательно, непрерывно, по столбцам |

Таблица 3. ЦВЗ – идентификационные данные.

|  |  |
| --- | --- |
| Наименование документа | Студенческий |
| Билет |
| Номер документа | 0020195843 |
| Фамилия | Синегубкин |
| Имя | Александр |
| Отчество | Сергеевич |
| Группа | А-12м-19 |
| Номер по списку в группе | 10 |
| Дата выдачи | 30.08.2019 |
| курс | пятый |

# Выполнение работы

## Исходные данные.

a = 0;

b = 23;

c = 7;

p = 13642163;

Алгоритм: ECDSA;

Хэш- функция: " MD5";

Размер изображения: 240х320;

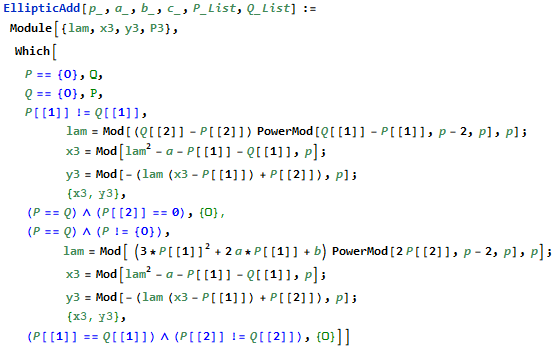
Метод встраивания: Метод блочного сокрытия;

Канал: зеленый;

Стего- путь: Последовательно, непрерывно, по столбцам;

## Часть 1. Поиск точки, удовлетворяющей условию.

Модуль сложения точек эллиптической кривой:



Описание функции SearchPoint:

Входные данные:

* **p** – порядок поля;
* **a** – коэффициент при x2;
* **b** – коэффициент при х;
* **с** – свободный член;
* **XMin** – минимальное значение переменной x;
* **RankMin** – минимальное значение порядка точки ЭК.

Выходные данные*:*

* P {x, y} – базовая точка ЭК, с координатами (x, y);
* rank – порядок точки;
* i – инкремент для свободного члена c.



Описание блоков:

1. Задаем коэффициенты эллиптической кривой согласно ТЗ:

a=0;b=23;c=7;p=13642163;i=0. (y2 = x3 + a\*x + (c + i)).

Задаем вспомогательные переменные:

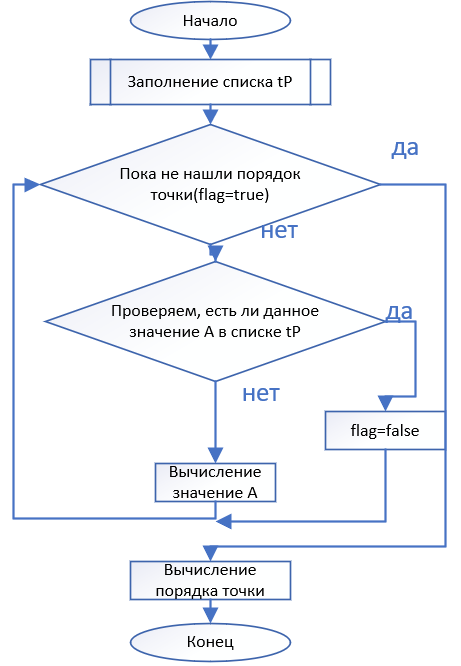
* found – флаг для поиска точка (flag1=False – точка найдена);
* flag – флаг для поиска порядка точки (flag2=False – порядок найден);
* r – размер вспомогательной списка для поиска порядка точки;
* tP – вспомогательная таблица для поиска порядки точки;
* RankMin = 2\*p/3 – задается для проверки порядка точки.

1. Цикл с предусловием While (пока точка не найдена found = False).
2. Гладкость эллиптической кривой определяется неравенством:

Если прямая гладкая, то:

* задаем начальное значение x1 = XMin (согласно ТЗ);
* обнуляем счетчик точек try.

1. Цикл с предусловием While (пока проверено точек < 40 или found = False)
2. В данном блоке для значения x вычисляется значение y2.
3. Алгоритм вычисления порядка точки эллиптической кривой (алгоритм больших и малых шагов):



1. Вычислить *r*  – округление с избытком (полученное из теоремы Хассе).

2. Построить список пар (t, tP) для t =1, 2, ..., r.

3. Вычислить *P1 = −r P*.

4. Положить *A = O* – бесконечно удаленная точка.

5. Для i r = − 1,2,..., 1 выполнить следующие действия:

5.1. проверить, будет ли точка A содержаться в таблице, построенной на шаге 1;

5.2. если *A = tP*, то закончить поиск;

5.3. положить *A = A + P1 .*

5.4. вычислить *rank = rs + t*

1. Сравнение порядка точки с заданным в ТЗ и проверка, что данный порядок является простым числом.



Результат действия модуля:

Найдена базовая точка P={6821082,5569902}.

Координата x=6821082 >= Floor[p/2]=6821081.

Порядок точки=13645001 >= 2p/3=9094775 является простым числом.

Эллиптическая кривая ^2=x^3+0\*x^2+23\*x+66 - гладкая. i=59

### Проверка эллиптической кривой на гладкость:

y = 5569902; x1 = 6821082; i = 59;

Mod[y ^2, p] = Mod[x1^3 + a\*x1^2 + b\*x1 + c + i, p]

PrimeQ[13645001] = True

## Часть 2. Реализация модуля электронно-цифровой подписи.

Реализацией алгоритма ЭЦП ECDSA являются функции ECDSAGenerateSign (генерация) и ECDSAVerificateSign (верификация). Для работы алгоритма также используются функции EllipticAdd (сложение двух точек ЭК), код которой приведен выше и Nsum (умножение точки ЭК на целое число).



Описание функции ECDSAGenerateSign:

Входные данные:

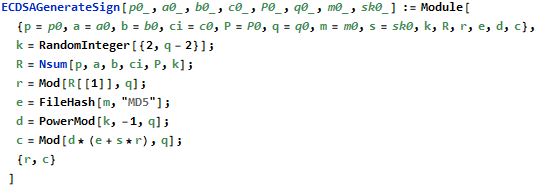
* **p** – порядок поля;
* **a** – коэффициент при x2;
* **b** – коэффициент при х;
* **c+i** – свободный член;
* **P** – базовая точка открытого ключа;
* **q** – порядок точки P;
* **m** – сообщение для которого формируется ЭЦП;
* **s** – секретный ключ подписывающего сообщение участника.

Выходные данные:

* **sign – {r, c}** – цифровая подпись.

Работа функции.

1. Выбираем случайную битовую строку
2. Вычисляем
3. Вычисляем
4. Вычисляем
5. Формируем цифровую подпись , где



Входные данные:

SeedRandom[16048];a=0;b=23;c=7;p=13642163;i=59;P = {6821082, 5569902}; q = 13645001; s = RandomInteger[{1, q - 1}] = 85308

m = "c:\\Users\\SiniyAl\\Documents\\WolframMathematica\\КР\\SAS.docx";

Выходные данные:

{1044692, 348100}

Описание функции ECDSAVerificateSign:

Входные данные:

* **p** – порядок поля;
* **a** – коэффициент при x2;
* **b** – коэффициент при х;
* **c+i** – свободный член;
* **P** – базовая точка открытого ключа;
* **q** – порядок точки P;
* **m** – проверяемое сообщение;
* **Q –** открытый ключ;
* **Sign –** полученная ЭЦП.

Выходные данные:

* Сообщение о результате проверки.

Работа функции.

1. Получить авторизованную копию открытого ключа **,** а также ***m' , P ,***

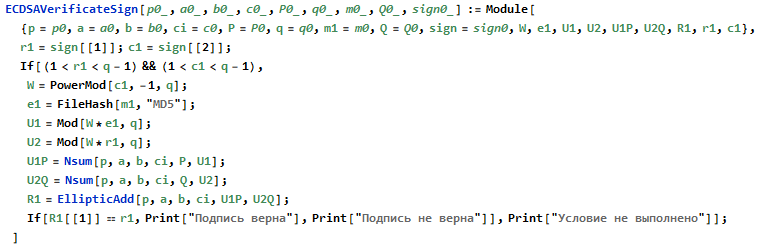
{***r' ,с'***}, ***q..***

2. Проверить, что илежат в интервале .

3. Вычислить и **.**

4. Определить и **.**

5. Вычислить и сравнить – компоненту точки с полученной по открытому каналу составляющей электронной подписи .



Входные данные:

SeedRandom[16048];a=0;b=23;c=7;p=13642163;i=59;P={6821082, 5569902};

q = 13645001; s = RandomInteger[{1, q - 1}] = 853083

m= "c:\\Users\\SiniyAl\\Documents\\WolframMathematica\\КР\\SAS.docx";

Q = Nsum[p, a, b, c + i, P, s]= {4049495, 12394182} – открытый ключ

S1 = {1044692, 348100}

Выходные данные:

Print["Проверка ЭЦП на правильном файле:"]

ECDSAVerificateSign[p,a,b,c+i,P,q,m,Q,S1]

Print["Проверка ЭЦП на неправильном файле:"]

ECDSAVerificateSign[p,a,b,c+i,P,q,"C:\\Readme.txt",Q,S1]

Проверка ЭЦП на правильном файле:

Подпись верна

Проверка ЭЦП на неправильном файле:

Подпись не верна

## 2.1 Встраивание идентификационной информации

Встраивание идентификационной информации (основных элементов содержимого студенческого билета) в фиксированный контейнер –фотографию студента.

Подготовить контейнер-оригинал – цифровую фотографию с размерами, указанными в Табл. 2.

img = Import["c:\\1n.bmp"];



ImageDimensions[img]

{240,320}

Провести предварительную подготовку встраиваемой информации из Табл. 3, для чего перевести все символы ЦВЗ в двоичную форму и выровнять цифры до размера букв – 11бит/символ с использованием функции PadLeft[\*\*\*,11], сформировать битовую строку предназначенную для встраивания в фотографию.

msg = "Наименование документа Студенческий билет;

Номер документа 0020195843;

Фамилия Синегубкин;

Имя Александр;

Отчество Сергеевич;

Группа А-12м-19;

Номер по списку в группе 10;

Дата выдачи 30.08.2019;

курс пятый"

str1 = ToCharacterCode[msgFSI];

str2 = IntegerDigits[str1, 2, 11];

str3 = Flatten[str2];

end = PadRight[IntegerDigits[16048, 2], 11] ;

dw = Flatten[AppendTo[str3, end]];

s = {};

Do[AppendTo[s,

FromCharacterCode[FromDigits[Partition[str3, 11][[i]], 2]]],

{i, 1, Length[str3]/11}]; StringJoin[s]

"Наименование документа Студенческий билет;

Номер документа 0020195843;

Фамилия Синегубкин;

Имя Александр;

Отчество Сергеевич;

Группа А-12м-19;

Номер по списку в группе 10;

Дата выдачи 30.08.2019;

курс пятый"

Разработать программный модуль для встраивания ЦВЗ в соответствии с методом и стего- путем, приведенными в Табл. 2.

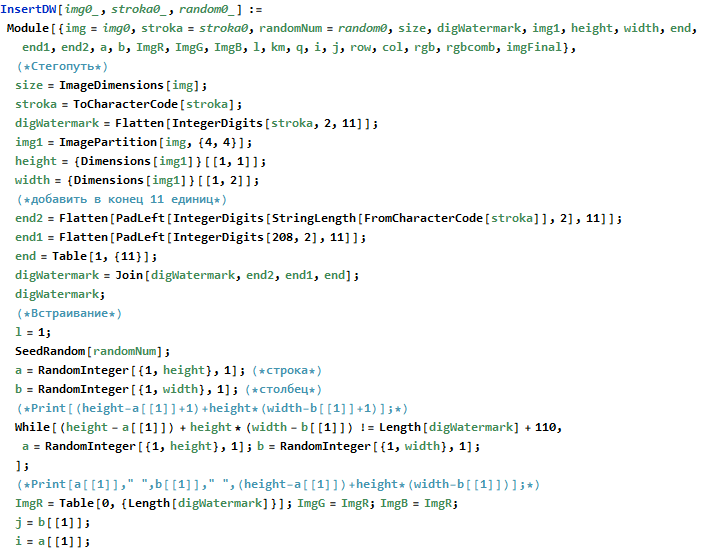
Описание функции InsertDW:

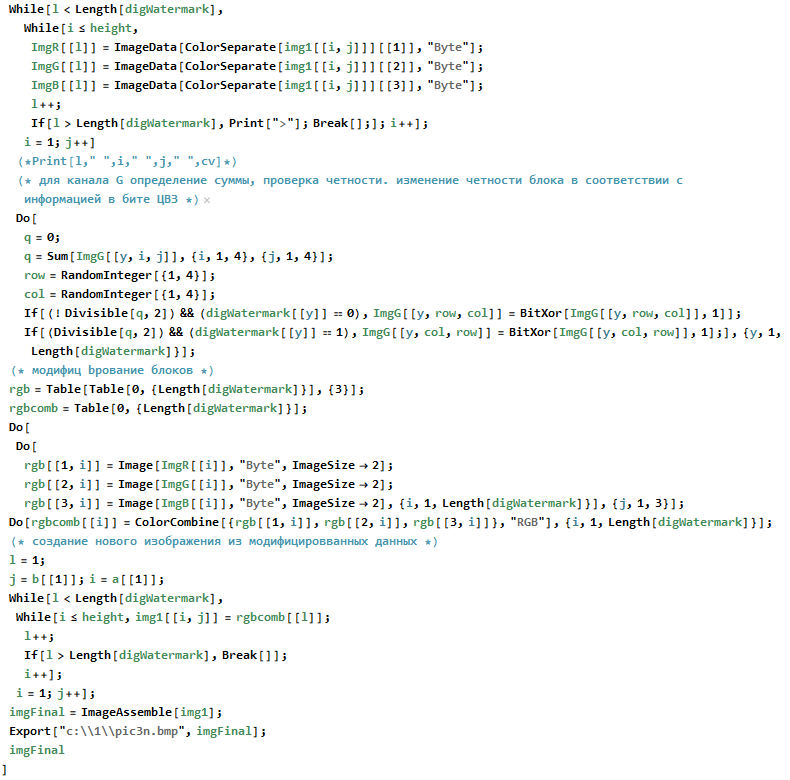
Входные данные:

* **img0** – исходное изображение;
* **stroka0** – информация для встраивания;
* **random0** – значение для генератор случайных чисел;

Выходные данные:

* **imgFinal –** изображение с встроенным ЦВЗ.





iCVZ = InsertDW[img, msg, "16048"]



Провести верификацию ЦВЗ путем восстановления встроенных данных.

Верификация ЦВЗ:

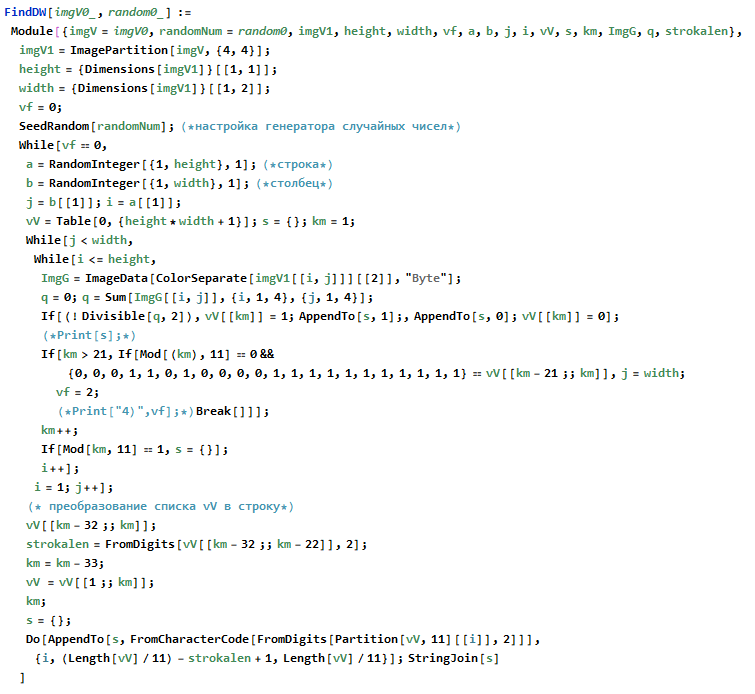
Описание функции FindDW:

Входные данные:

* **imgV0** – изображение с встроенным ЦВЗ;
* **random0** – значение для генератор случайных чисел;

Выходные данные:

* **s –** встроенная информация.

fCVZ = FindDW[iCVZ, "16048"]

"Наименование документа Студенческий билет;

Номер документа 0020195843;

Фамилия Синегубкин;

Имя Александр;

Отчество Сергеевич;

Группа А-12м-19;

Номер по списку в группе 10;

Дата выдачи 30.08.2019;

курс пятый"

Создать ЭЦП для модифицированного контейнера со встроенным ЦВЗ с использованием функции FileHash[].

S1 = ECDSAGenerateSign[p, a, b, c + i, P, q, m, s]

{1044692, 4685811}

Print["Проверка ЭЦП на правильном файле:"]

ECDSAVerificateSign[p, a, b, c + i, P, q, m, Q, S1]

Проверка ЭЦП на правильном файле:

Подпись верна

Print["Проверка ЭЦП на неправильном файле:"]

ECDSAVerificateSign[p, a, b,

c + i, P, q, "c:\\Users\\SiniyAl\\Documents\\Wolfram Mathematica\\КР\

\\1n.bmp", Q, S1]

Проверка ЭЦП на неправильном файле:

Подпись не верна

# Заключение

С помощью системы компьютерной алгебры Wolfram Mathematica были решены поставленные задачи. Создан программный модуль реализующий алгоритм ЭЦП ECDSA и хэш-функцию MD5, и модуль для встраивания ЦВЗ в соответствии с методом блочного сокрытия. Оба модуля прошли успешную верификацию.

# СПИСОК ЛИТЕРАТУРЫ

1. **Болотов А.А., Гашков С.Б., Фролов А.Б., Часовских А.А.** Элементарное введение в эллиптическую криптографию: Алгебраические и алгоритмические основы.- М.: КомКнига, 2006.-328с.
2. **Грибунин В.Г., Оков И.Н., Туринцев И.В.** Цифровая стеганография. –М.: СОЛОН-Пресс,2002.- 272 стр.
3. **Конахович Г.Ф., Пузыренко А.Ю.** Компьютерная стеганография. Теория и практика.-К.: "МК-Пресс", 2006.-288с.,ил.