Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Национальный исследовательский университет «Московский энергетический институт»

Кафедра Вычислительных машин, систем и сетей.

КУРСОВОЙ ПРОЕКТ

по дисциплине: Цифровые технологии защиты информации.

на тему: Биометрический идентификатор с электронной цифровой подписью.

Выполнила:

Федосенко С.И.

А-12м-19

Проверил:

Рытов А.А.

Москва 2019

Оглавление

[Цель работы 2](#_Toc26816070)

[Рабочее задание 2](#_Toc26816071)

[Часть 1. 2](#_Toc26816072)

[Часть 2. 3](#_Toc26816073)

[Выполнение работы 4](#_Toc26816074)

[Исходные данные. 4](#_Toc26816075)

[1.1 Поиск точки, удовлетворяющей условию. 4](#_Toc26816076)

[1.1.1. Проверка ЭК на гладкость (D != 0): 7](#_Toc26816077)

[1.2 Реализация модуля электронно- цифровой подписи. 7](#_Toc26816078)

[2.1 Встраивание идентификационной информации 10](#_Toc26816079)

[Заключение 15](#_Toc26816080)

[СПИСОК ЛИТЕРАТУРЫ 16](#_Toc26816081)

# Цель работы

В фотографию с помощью стеганографических методов встроить данные, содержащиеся в студенческом билете, и создать ЭЦП этого цифрового объекта.

# 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,

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 |
| 20 | 25 | 25+i |  | ГОСТ Р 3410-2001 | "SHA512" |

## 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=2.
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 | 1 |
| Размер фотографии,  ширина х высота | 240х320 |
| Размер блока, точек | - |
| Метод встраивания  ЦВЗ | Метод замены наименьшего значащего бита |
| Стего путь | Последовательно, непрерывно,  по строкам. |

Таблица 3. ЦВЗ – идентификационные данные.

|  |  |
| --- | --- |
| Наименование документа | Студенческий |
| Билет |
| Номер документа | 0020195826 |
| Фамилия | Федосенко |
| Имя | Софья |
| Отчество | Игоревна |
| Группа | А-12м-19 |
| Номер по списку в группе | 11 |
| Дата выдачи | 30.08.2019 |
| курс | первый |

# Выполнение работы

## Исходные данные.

a = 0;

b = 25;

c = 25;

p = 31313267;

Алгоритм: ГОСТ Р 3410-2001;

Хэш- функция: "SHA512";

Размер изображения: 240х320;

Метод встраивания: Метод замены наименьшего значащего бита;

Канал: синий;

Стего- путь: по строкам 1-2-3-2;

## 1.1 Поиск точки, удовлетворяющей условию.

Модуль сложения точек эллиптической кривой:

EllipticAdd[p\_, a\_, b\_, c\_, P\_List, Q\_List] :=

Module[{lam, x3, y3, P3},

Which[

P == {O}, Q,

Q == {O}, P,

P[[1]] != Q[[1]],

lam =

Mod[(Q[[2]] - P[[2]]) PowerMod[Q[[1]] - P[[1]], p - 2, p], p];

x3 = Mod[lam^2 - a - P[[1]] - Q[[1]], p];

y3 = Mod[-(lam (x3 - P[[1]]) + P[[2]]), p];

{x3, y3},

(P == Q) \[And] (P[[2]] == 0), {O},

(P == Q) \[And] (P != {O}),

lam =

Mod[ (3\*P[[1]]^2 + 2 a\*P[[1]] + b) PowerMod[2 P[[2]], p - 2, p],

p];

x3 = Mod[lam^2 - a - P[[1]] - Q[[1]], p];

y3 = Mod[-(lam (x3 - P[[1]]) + P[[2]]), p];

{x3, y3},

(P[[1]] == Q[[1]]) \[And] (P[[2]] != Q[[2]]), {O}]]

Описание функции PoiskFSI:

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.

PoiskFSI[p0\_, a0\_, b0\_, c0\_, xm\_, rm\_] := Module[

{p = p0, a = a0, b = b0, c = c0, XMin = xm, RankMin = rm, found, x1,

y1, i, r, try, y, tP, P, P1, A, s, flag, t, rank},

found = False;

x1 = 0;

y1 = 0;

i = 0;

r = Ceiling[Sqrt[Round[p + 1 + 2\*Sqrt[p]]]];

tP = Table[0, r];

While[found == False,

If[Mod[4\*b^3 + 27\*(c + i)^2, p] != 0,

(\*Print["Кривая y^2=x^3+",a,"\*x^2+",b,"\*x+",c,"+",i,

" - гладкая"];\*)

x1 = XMin; try = 0;

While[(try < 50) && (found == False),

If[Solve[y^2 == x1^3 + a\*x1^2 + b\*x1 + c + i, {y},

Modulus -> p] != {},

y1 =

y /. Flatten[

Solve[y^2 == x1^3 + a\*x1^2 + b\*x1 + c + i, {y},

Modulus -> p], 1];

tP[[1]] = {x1, y1};

Do[

tP[[l]] =

EllipticAdd[p, a, b, c + i, tP[[l - 1]], tP[[1]]], {l, 2, r}];

P1 = Mod[tP[[r]]\*{1, -1}, p];

A = {O}; s = 0; flag = True;

While[flag,

A = EllipticAdd[p, a, b, c + i, P1, A]; s++;

For[t = 1, t <= r, t++,

If[tP[[t]] == A,

flag = False; Break[];

];

];

];

rank = r\*s + t;

Which[rank > RankMin && PrimeQ[rank],

{Print["Найдена базовая точка P=", tP[[1]], ".

Координата x=", x1, " \[GreaterEqual] Floor[p/2]=", XMin, ".

Порядок точки=", rank, " \[GreaterEqual] 2p/3=", RankMin,

" и является простым числом.

Эллиптическая кривая y^2=x^3+", a, "\*x^2+", b, "\*x+", c + i,

" - гладкая. i=", i];

found = True;},

rank > RankMin && ! (PrimeQ[rank]), {(\*Print[

"Найдена базовая точка P = ",tP[[1]],". Порядок = ",rank,

" не является простым числом. i = ",i];\*)

x1++;

try++;},

rank <= RankMin, {(\*Print["Найдена базовая точка P = ",

tP[[1]],". Порядок = ",rank," < 2p/3. i = ",i];\*)

x1++;

try++;}];,

(\*Точка не принадлежит кривой.\*)

x1++;

try++;

];

];

(\*,

Print["Кривая y^2=x^3+",a,"\*x^2+",b,"\*x+",c,"+",i,

" - не гладкая"];\*)

];

i++;];

]

Timing[PoiskFSI[p, a, b, c, XMin, RankMin]] // AbsoluteTiming

Входные данные:

a = 0;

b = 25;

c = 25;

p = 31313267;

RankMin = Floor[2\*p/3];

XMin = Floor[p/2];

Результат действия модуля:

Найдена базовая точка P={15656633,14773226}.

Координата x=15656633 >= Floor[p/2]=15656633.

Порядок точки=31310681 >= 2p/3=20875511 и является простым числом.

Эллиптическая кривая y^2=x^3+0\*x^2+25\*x+103 - гладкая. i=78

{58014.5, {57587.6, Null}}

### Проверка ЭК на гладкость (D != 0):

Mod[14773226^2, p] = Mod[15656633^3 + 25\*15656633 + 103, p]

PrimeQ[31310681] = True

## 1.2 Реализация модуля электронно- цифровой подписи.

Реализацией алгоритма ЭЦП ГОСТ Р 3410-2001 являются функции GOST3410SigntureCreationFSI(генерация) и GOST3410SigntureVerificationFSI (верификация). Для работы алгоритма также используются функции EllipticAdd (сложение двух точек ЭК), код которой приведен выше и Nsum (умножение точки ЭК на целое число методом аддитивных цепочек).

Описание функции GOST3410SigntureCreationFSI:

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:

* **S1 – {r, c}** – цифровая подпись.

Работа функции.

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

GOST3410SigntureCreationFSI[p0\_, a0\_, b0\_, c0\_, P0\_, q0\_, m0\_, sk0\_] :=

Module[

{p = p0, a = a0, b = b0, ci = c0, P = P0, q = q0, m = m0, s = sk0,

k, R, r, e, c},

k = RandomInteger[{2, q - 2}];

R = Nsum[p, a, b, ci, P, k];

r = Mod[R[[1]], q];

e = FileHash[m, "SHA512"];

c = Mod[r\*s + k\*e, q];

{r, c}

]

Входные данные:

SeedRandom[15948];

a = 0;

b = 25;

c = 25;

p = 31313267;

i = 78;

P = {15656633, 14773226};

q = 31310681;

m = "C:\\Users\\Администратор\\Downloads\\FSI.docx";

s = RandomInteger[{1, q - 1}] =28346083

Выходные данные:

{24521193, 25219420}

Описание функции GOST3410SigntureVerificationFSI:

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;
* **S1 –** полученная ЭЦП.

Выходные данные:

* Сообщение о результате проверки.

Работа функции.

1. Получить авторизованную копию открытого ключа **,** а также
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

GOST3410SigntureVerificationFSI[p\_, a\_, b\_, c\_, P\_, q\_, Q\_, m\_,

Sign\_] := Module[

{f, r, c0, e, V, z1, z2, R},

f = False;

r = Sign[[1]];

c0 = Sign[[2]];

If[(r > 1) && (r < q - 1) && (c0 > 1) && (c0 < q - 1),

e = FileHash[m, "SHA512"];

V = PowerMod[e, -1, q];

z1 = Mod[c0\*V, q];

z2 = Mod[-r\*V, q];

R = EllipticAdd[p, a, b, c, Nsum[p, a, b, c, P, z1],

Nsum[p, a, b, c, Q, z2]];

If[R[[1]] == r,

f = True

];

];

If[f == True,

Print["Подпись верна"],

Print["Подпись не верна"]

];

]

Входные данные:

SeedRandom[15948];

a = 0;

b = 25;

c = 25;

p = 31313267;

i = 78;

P = {15656633, 14773226};

q = 31310681;

m = "C:\\Users\\Администратор\\Downloads\\FSI.docx";

s = RandomInteger[{1, q - 1}] =28346083

Q = Nsum[p, a, b, c + i, P, s]= {27646357, 27248486}

S1 = {24521193, 25219420}

Выходные данные:

Print["Проверка ЭЦП на правильном файле:"]

GOST3410SigntureVerificationFSI[p, a, b, c + i, P, q, Q, m, S1]

Print["Проверка ЭЦП на неправильном файле:"]

GOST3410SigntureVerificationFSI[p, a, b,

c + i, P, q, Q, "C:\\Readme.txt", 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.

imgFSI = Import["E:\\kr\\imgFSI.bmp"]



ImageDimensions[imgFSI]

{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.

msgFSI = "Наименование документа Студенческий билет;

Номер документа 0020195826;

Фамилия Федосенко;

Имя Софья;

Отчество Игоревна;

Группа А-12м-19;

Номер по списку в группе 11;

Дата выдачи 30.08.2019;

курс первый"

str1 = ToCharacterCode[msgFSI];

str2 = IntegerDigits[str1, 2, 11];

str3 = Flatten[str2];

end = PadRight[IntegerDigits[15948, 2], 11]

{1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1}

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 0020195826;

Фамилия Федосенко;

Имя Софья;

Отчество Игоревна;

Группа А-12м-19;

Номер по списку в группе 11;

Дата выдачи 30.08.2019;

курс первый"

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

blCh = ColorSeparate[imgFSI][[3]];

ChData = ImageData[blCh, "Byte"];

wide = 240;

(\*Стего-путь 1-2-3-2 1-2-3-2...\*)

For[i = 1, i <= Length[str3], i++,

If[str3[[i]] == 0,(\*Выбор BitSet или BitClear\*)

If[Mod[i - 1, 4] == 3,(\*Формирование стего-пути\*)

ChData[[Floor[i/wide] + 1]][[Mod[i, wide, 1]]] =

BitClear[ChData[[Floor[i/wide] + 1]][[Mod[i, wide, 1]]], 1],

ChData[[Floor[i/wide] + 1]][[Mod[i, wide, 1]]] =

BitClear[ChData[[Floor[i/wide] + 1]][[Mod[i, wide, 1]]],

Mod[i - 1, 4]]],

If[Mod[i - 1, 4] == 3,

ChData[[Floor[i/wide] + 1]][[Mod[i, wide, 1]]] =

BitSet[ChData[[Floor[i/wide] + 1]][[Mod[i, wide, 1]]], 1],

ChData[[Floor[i/wide] + 1]][[Mod[i, wide, 1]]] =

BitSet[ChData[[Floor[i/wide] + 1]][[Mod[i, wide, 1]]],

Mod[i - 1, 4]]]]]

Провести верификацию ЦВЗ путем восстановления встроенных данных по соответствующему алгоритму.

Полный контейнер:

modifiedFSI =

ColorCombine[{ColorSeparate[imgFSI][[1]], ColorSeparate[imgFSI][[2]],Image[ChData, "Byte"]}, "RGB"]



Export["E:\\kr\\modf.bmp", modifiedFSI]

E:\kr\modf.bmp

Верификация ЦВЗ:

ChDataMod = ImageData[ColorSeparate[modified][[3]], "Byte"];

verString = RandomInteger[{0, 0}, Length[str3]];

For[i = 1, i <= Length[verString], i++,

If[Mod[i - 1, 4] == 3,

verString[[i]] =

BitGet[ChDataMod[[Floor[i/wide] + 1]][[Mod[i, wide, 1]]], 1],

verString[[i]] =

BitGet[ChDataMod[[Floor[i/wide] + 1]][[Mod[i, wide, 1]]],

Mod[i - 1, 4]]]]

secretData = Partition[verString, 11];

FromCharacterCode[FromDigits[#, 2] & /@ secretData]

"Наименование документа Студенческий билет;

Номер документа 0020195826;

Фамилия Федосенко;

Имя Софья;

Отчество Игоревна;

Группа А-12м-19;

Номер по списку в группе 11;

Дата выдачи 30.08.2019;

курс первый"

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

S2 = GOST3410SigntureCreationFSI[p, a, b, c + i, P, q,

"H:\\kursach\\imgFSI.bmp", s]

Print["Проверка ЭЦП на правильном файле:"]

GOST3410SigntureVerificationFSI[p, a, b, c + i, P, q, Q, m, S2]

Проверка ЭЦП на правильном файле:

Подпись верна

Print["Проверка ЭЦП на неправильном файле:"]

GOST3410SigntureVerificationFSI[p, a, b,

c + i, P, q, Q, "H:\\kursach\\1.bmp", 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 ГОСТ Р 3410-2001 и хэш-функцию SHA512, и модуль для встраивания ЦВЗ в соответствии с методом замены наименьшего значащего бита. 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. **Тилборг Ван Х.К.Ф.** Основы криптологии. Профессиональное руководство и интерактивный учебник. – М.:Мир,2006,стр.471,ил.
3. **Конахович Г.Ф., Пузыренко А.Ю.** Компьютерная стеганография. Теория и практика.-К.: "МК-Пресс", 2006.-288с.,ил.
4. **Миано Дж.** Форматы и алгоритмы сжатия изображений в действии. Учеб.пособ.- М.: Триумф, 2003-336 с.: ил.