Министерство образования Республики Беларусь

Учреждение образования

«Полоцкий государственный университет»

ПАСТУХОВ Д.Ф. (Полоцкий университет);
ВОЛОСОВА Н.К. (Московский государственный технический университет им. Н.Э. Баумана – Национальный исследовательский университет);
ПАСТУХОВ Ю.Ф.;СЕРЫЙ Т.А.;БАТАЛКО И.И.; ВАСИЛЕВИЧ В.В; СМОЛЯК А.И. (Полоцкий университет)

АЛГЕБРАИЧЕСКИЕ МЕТОДЫ ШИФРОВАНИЯ

Учебное пособие к лекционным и практическим занятиям
для студентов специальности

1-98 01 01 Компьютерная безопасность

Новополоцк
ПГУ
2020

УДК 519.66

Рецензенты:
А.А. Козлов, кандидат физико-математических наук, доцент, Заведующий кафедрой высшей математики и дифференциальных уравнений Полоцкого государственного университета

Пастухов Д.Ф., Волосова Н.К., Пастухов Ю.Ф., Серый Т.А., Баталко И.И., Василевич В.В, Смоляк А.И.
Алгебраические методы шифрования:
Учебное пособие. – Новополоцк: 2020.-17с.

В учебном пособии рассмотрено два примера преобразований $Z_p \to Z_p \times Z_p$ при шифровании и $Z_p \leftarrow Z_p \times Z_p$ при дешифровании. А также пример взаимно-однозначного отображения $Z_p \times Z_p \times Z_p \to Z_p \times Z_p \times Z_p$ при шифровании и обратно $Z_p \times Z_p \times Z_p \leftarrow Z_p \times Z_p \times Z_p$ при дешифровании с использованием аффинного преобразования. Написанные программы к примерам на языке FORTRAN можно использовать в качестве ядра для других программ.

Для студентов университетов, педагогических вузов, а также для студентов технических вузов, преподавателей, инженеров, программистов использующих в своей практической деятельности математические методы шифрования.

УДК 519.66

Оглавление

ВВЕДЕНИЕ

По словам академика Александра Андреевича Самарского современная математика в настоящее время имеет много общего с экспериментальной наукой. Программистом, как правило, называют специалиста-разработчика использующего известные алгоритмы и создающего новые алгоритмы. Однако теперь область деятельности программиста связана и с прикладной математикой. Сегодня программист не может отказаться от задач прикладной математики и сослаться, например, на занятость в сфере банковских услуг. А областью деятельности программиста являются все известные разделы математики на сегодняшний день! Исторически сложилось так, что в двадцатом веке математики всего мира создали специальный язык FORTRAN и математические библиотеки imsl,msimsl(для потребностей линейной алгебры)[8,9,10]. Можно сказать FORTRAN-математический язык-конструктор и, любую математическую задачу можно программировать на языке FORTRAN за меньшее число строк кода, чем на любом известном языке на сегодняшний день. А, следовательно, быстродействие программ на языке FORTRAN выше быстродействия программ на других языках, поскольку программа затрачивает на чтение и выполнение операций каждой строки кода. Следовательно, программистам, косвенно связанным с математикой полезно познакомиться с мощными методами FORTRAN и его библиотек. Исходя из этого, авторы учебного пособия предлагают студентам три метода шифрования в области целых чисел и с программными кодами на FORTRAN-не, чтобы иметь практику перевода их на все доступные им известные современные языки. Один из авторов пособия бывал на лекциях профессора математики Валентина Федоровича Бутузова и слышал от него такое высказывание: ”Серьезное изучение математики вырабатывает у человека математическую культуру, то есть возможность ориентироваться в математических разделах и литературе, математическую интуицию того какие математические объекты и как они связаны между собой”. Прикладной раздел математики – шифрование связан с ее областью – теорией чисел.

Наиболее простой метод шифрования - применение монотонных функций действительного переменного на интервале или прямой требует, чтобы прямая и обратная к ней функции были взаимно однозначны. Например, отображение синуса на арксинус в соответствующих интервалах[5,7] $\sin\left[-\frac{\pi}{2},\frac{\pi}{2}\right] \leftrightarrow \arcsin[-1,1]$. Наиболее часто при шифровании текстовой информации применяют множество целых положительных остатков группы Z_p, так как мощность любого алфавита конечна[1,2,3,4,6]. Множество остатков Z_p образует поле с двумя операциями сложения и умножения, если число p простое. То есть операции сложения (вычитания), умножения (деления - нахождение числа обратному к делимому по модулю p) являются замкнутыми над полем Z_p. Дополнительная групповая операция сложения целочисленных точек алгебраических кривых приводит, например, к эллиптическому кодированию[6].

Под алгебраическим кодированием будем понимать применение алгебры многочленов (кольца многочленов) нескольких переменных с коэффициентами и значениями переменных из множества целочисленных остатков по модулю простого числа p Z_p. Под руководством академика Андрея Николаевича Колмогорова его ученик Алексей Михайлович Матиясевич решил девятую проблему Гильберта. То есть показал, что не существует конечного алгоритма для решения алгебраического полинома нескольких переменных в целых числах. В учебном пособии мы рассмотрим три примера алгебраического шифрования.

При шифровании над полем целых остатков Z_p с фиксированными ключами обычно используют отображение множества в себя $Z_p \to Z_p$, а при дешифровании обратное отображение $Z_p \leftarrow Z_p$. В учебном пособии мы рассмотрим два примера преобразований $Z_p \to Z_p \times Z_p$ при шифровании и обратно $Z_p \leftarrow Z_p \times Z_p$ при дешифровании. А также пример взаимно-однозначного отображения $Z_p \times Z_p \times Z_p \to Z_p \times Z_p \times Z_p$ при шифровании и обратно $Z_p \times Z_p \times Z_p \leftarrow Z_p \times Z_p \times Z_p$ при дешифровании. Идея третьего примера принадлежит Волосовой Наталье Константиновне.

Пример 1(Algebra 1).

Запишем известную алгебраическую формулу

$$y^n - x^n = (y-x)\left(y^{n-1} + y^{n-2}x + \ldots + yx^{n-2} + x^{n-1}\right) = (y-x)\sum_{i=0}^{n-1} x^i y^{n-1-i} \quad (1)$$

Пусть целые числа x,n ключи шифрования, y – номер текущего символа в сообщении по таблице ASCII. Введем обозначения

$$R(x,y,n) = y^n - x^n, Q(x,y,n) = \sum_{i=0}^{n-1} x^i y^{n-1-i} \Leftrightarrow (y-x) \overset{(1)}{=} \frac{R(x,y,n)}{Q(x,y,n)} \Leftrightarrow y = x + \frac{R(x,y,n)}{Q(x,y,n)}, Q(x,y,n) \neq 0 \quad (2)$$

В формуле(2) исключаются все нули - корни уравнения $y : Q(x,y,n) = 0$ при заданных целых ключах x,n. Рассмотрим алгебраические преобразования(2) на множестве целых положительных остатков Z_p:

Шифром формулой(1) с ключами x,n символа y назовем пару целых чисел $\left(R(x,y,n)(\bmod p); Q(x,y,n)(\bmod p)\right)$

$$\left(R(x,y,n)(\bmod p); Q(x,y,n)(\bmod p)\right) = \left(\left(y^n - x^n\right)(\bmod p), \sum_{i=0}^{n-1} x^i y^{n-1-i} (\bmod p)\right), y \notin Y_0 : Q(x,y,n) = 0 \quad (3)$$

$$y \equiv x(\bmod p) + R(x,y,n)(\bmod p) \cdot Q^{-1}(x,y,n)(\bmod p) \mid Q^{-1}(x,y,n)(\bmod p) \cdot Q(x,y,n) \equiv 1(\bmod p) \quad (4)$$

Итак, алгоритм шифрования-дешифрования формулой (1) можно разбить на два случая:

1) $y \notin Y_0 : Q(x,y,n) = 0$. Шифрование по формуле(3), дешифрование по формуле(4).

2) $y \in Y_0 : Q(x,y,n) = 0$. Шифром назовем пару чисел $\left(R(x,y) \equiv x + y(\bmod p); Q(x,y) \equiv y - x(\bmod p)\right)$ (5)

Дешифрование проводим по формуле(6),отметим, что число $2^{-1}(\bmod p)$ существует, если p простое.

$$y \equiv R(x,y) + Q(x,y) \equiv x + y + y - x(\bmod p) = 2y(\bmod p) \Leftrightarrow y = \left(R(x,y) + Q(x,y)\right)(\bmod p) \cdot 2^{-1}(\bmod p) \quad (6)$$

Усложним формулы (3)-(6), введем дополнительные целые ключи a,b и множитель $a^b \in N$

1) $y \notin Y_0 : Q(x,y,n) = 0$.Шифрование проводим по формуле(7), дешифрование по формуле(8).

$$\left(R(x,y,n)(\bmod p); Q(x,y,n)(\bmod p)\right) = \left(a^b \cdot \left(y^n - x^n\right)(\bmod p), a^b \cdot \sum_{i=0}^{n-1} x^i y^{n-1-i} (\bmod p)\right), y \notin Y_0 : Q(x,y,n) = 0 \quad (7)$$

$$y \equiv x(\bmod p) + R(x,y,n)(\bmod p) \cdot Q^{-1}(x,y,n)(\bmod p) \mid Q^{-1}(x,y,n)(\bmod p) \cdot Q(x,y,n) \equiv 1(\bmod p) \quad (8)$$

2) $y \in Y_0 : Q(x,y,n) = 0$. Шифром назовем пару чисел $\left(R(x,y) \equiv x + y(\bmod p); Q(x,y) \equiv y - x(\bmod p)\right)$ (9)

Дешифрование проводим по формуле(10), p простое.

$$y = \left(R(x,y) + Q(x,y)\right)(\bmod p) \cdot 2^{-1}(\bmod p) \quad (10)$$

Ниже написана программа с использованием алгоритма(7)-(10) на языке FORTRAN, в которой шифруется символьная фраза s="Polotsk State University 1234567890" с ключами a=1119;b=131; n=10000;x=103. Простое число p=257 выбрано ближайшим простым к мощности клавиатуры ASCII равной 256. Допустимы и большие простые числа, чем 257, однако клавиатура ASCII f не сможет в консоли отобразить все символы шифра для произвольного сообщения, однако декодирование все равно производится верно.

```
program algebra1
integer(8),parameter::n=10000,p=257,len=40
integer(8)::x,y,z1,z2,z3,z4,z00,mas1(len+1),mas2(len+1),mas3(len+1)
integer(8)::a,b
character(len+2)::s
integer(8)::mas(len+2)
s="Polotsk State University 1234567890"
a=1119;b=131
x=103;
do i=1,len
mas(i)=ichar(s(i:i))
print*,s(i:i),mas(i)
enddo
do i=1,len
!print*,"i=",i
y=mas(i)
call f1(x,y,n,p,a,b,z2)
if(z2==0)then
z1=mod(x+y,p)
print*,"********************************************************** zero *********"
```

```
z2=mod(y-x,p)
if(z1<0)then
z1=z1+p
elseif(z2<0)then
z2=z2+p

endif
!print*,z1,z2
call f3(2,p,z3)
z4=mod((z1+z2)*z3,p)
mas1(i)=z4
mas2(i)=0
elseif(.not.z2==0)then
call f3(z2,p,z3)
call f2(x,y,n,p,a,b,z00)

z4=mod(z3*z00+x,p)
if(z4<0)then
z4=z4+p
end if
mas1(i)=0
mas2(i)=z4
!print*,z2,z00
endif
mas3(i)=mas1(i)+mas2(i)
print*,"i=",i,mas3(i),char(mas3(i))
enddo
pause

end program algebra1
subroutine f1(x,y,n,p,a,b,z2)
integer(8)::a,b,d
integer(8)::i,j
integer(8)::z1,z,z2,z0,n,p,x,y
z0=0;d=1
do i=1,b
d=mod(d*a,p)
enddo
do i=0,n-1
z1=1;z=1
do j=1,i
z=mod(z*x,p)
enddo
do j=1,n-i-1
z1=mod(z1*y,p)
enddo
z0=mod((z0+z*z1),p)
enddo
z2=mod(z0*d,p)
if(z2<0)then
z2=z2+p
endif
end subroutine
subroutine f2(x,y,n,p,a,b,z00)
integer(8)::a,b,c,d
integer(8)::i
integer(8)::z1,z2,z00,x,y,n,p
d=1
do i=1,b
d=mod(d*a,p)
enddo
z1=1;z2=1;
do i=1,n
```

```
z1=mod(z1*x,p)
z2=mod(z2*y,p)
enddo
z00=mod(z2-z1,p)
if(z00<0)then
z00=z00+p
endif
z00=mod(z00*d,p)
end subroutine
subroutine f3(t,p,z0)
integer(8)::t,p,i,z0
do i=1,p-1
if(mod(i*t,p)==1)then
z0=i
endif
enddo
end subroutine
```

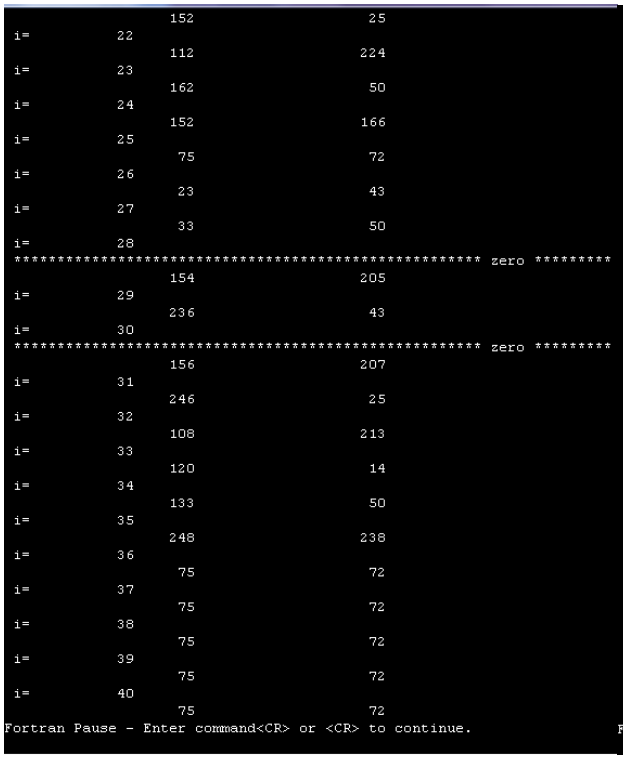

Рис.1

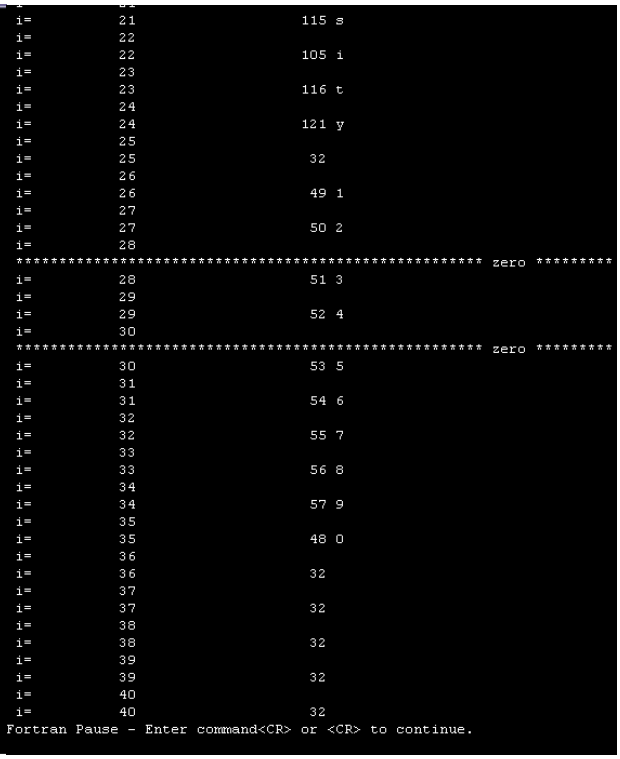

Рис.2

На рисунках 1,2 показан пример шифрования фразы s="Polotsk State University 1234567890" с ключами a=1119;b=131; n=10000;x=103. Рисунок 1 представляет шифр конца фразы в консоли, а рисунок 2 конец дешифрованной фразы. Данные рисунки подтверждают, что оба алгоритма программы работают одновременно. Например, символы “3”,”5” с порядковыми номерами $y=\{51;53\}\in Y_0:\sum_{i=0}^{9999}103^i y^{9999-i}\equiv 0 \pmod{257}$. Поэтому символы “3”,”5” шифруются и дешифруются вторым алгоритмом по формулам(9),(10). Все остальные символы фразы шифруются и дешифруются первым алгоритмом по формулам(7),(8). Отметим, что при больших значениях степени n=10000 программа затрачивает на шифрование время 85 с. Дешифрование происходит быстро за микросекунды. Но чем дольше и сложнее алгоритм шифрования, тем сложнее работа крипто-аналитика!

Сделаем оценку пространства ключей и время его прямого перебора. Ключам a,b,x, сопоставим мощность 256, степени n мощность 10000. Тогда имеем мощность пространства ключей $N\approx 256^3\cdot 10000=1.677\cdot 10^{11}$. Считаем, что в среднем крипто-аналитик затрачивает на подбор одной комбинации ключей время 10^{-3}. Тогда полное время $T\approx 1.677\cdot 10^{11-3}c=1.677\cdot 10^8 c=1941 суток\approx 5{,}3 лет$. При необходимости в алгоритм(7)-(10) можно добавить еще два ключа.

Отметим также, что отдельные массивы – шифры двух алгоритмов на конечном этапе после дешифрования объединяются в один массив (складываются после дешифрования в один массив по правилам арифметики).

Размещая в консоли все записи компактно, из рисунка 3 видно, что все символы фразы s="Polotsk State University 1234567890" с числом знаков 40(лишние символы не входящие во фразу дешифруются пробелами – с порядковым номером символа 32) декодируются взаимно-однозначно.

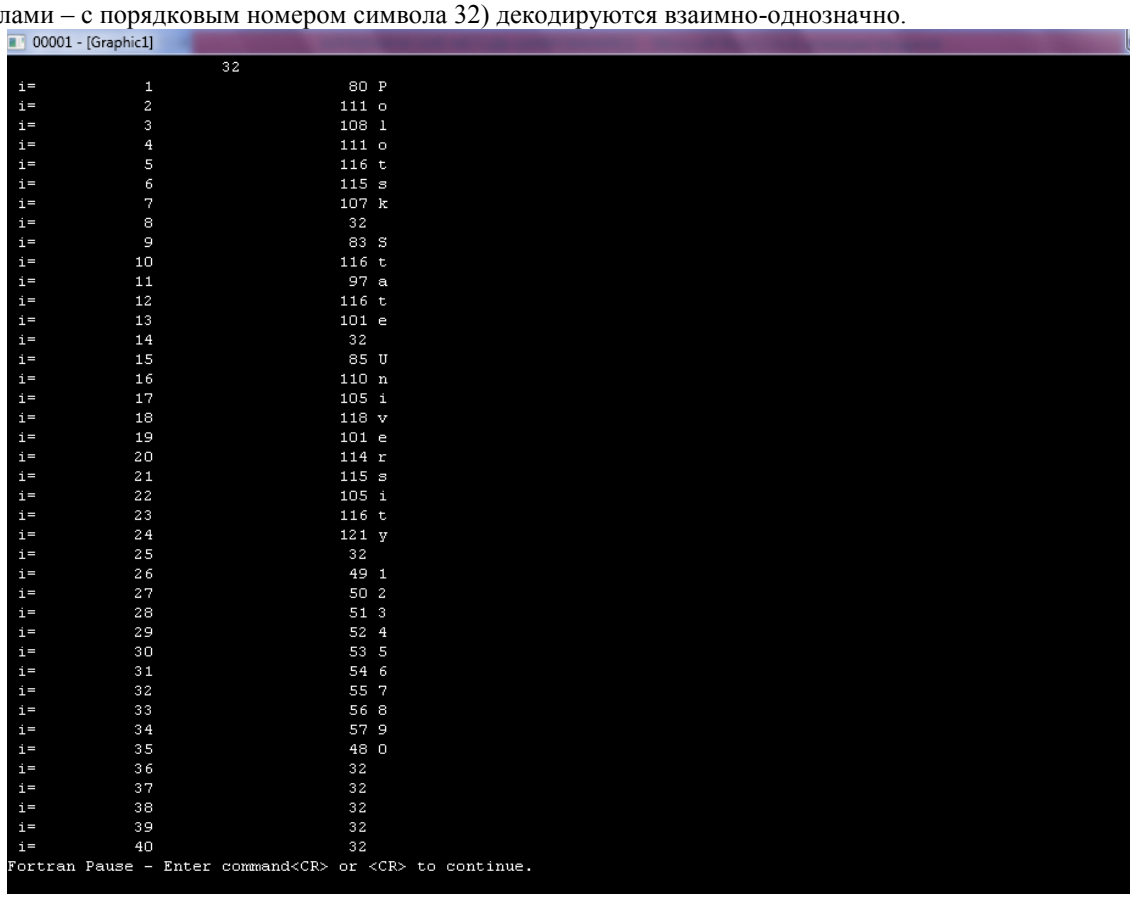

Рис.3. Пример дешифрования программой Algebra 1.

Пример 2(Algebra 2).

Запишем вторую алгебраическую формулу

$$y^{2k+1}+x^{2k+1}=(y+x)\left(y^{2k}-y^{2k-1}x+\ldots+(-1)^{2k-1}yx^{n-2}+(-1)^{2k}x^{2k}\right)=(y+x)\sum_{i=0}^{2k}x^{i}y^{2k-i} \tag{11}$$

Пусть целые числа x,n ключи шифрования, $y-$ номер текущего символа в сообщении по таблице ASCII. Введем обозначения

$$R(x,y,n)=y^{2k+1}+x^{2k+1},Q(x,y,n)=\sum_{i=0}^{2k}x^{i}y^{2k-i}\Leftrightarrow(y+x)\overset{(11)}{=}\frac{R(x,y,n)}{Q(x,y,n)}\Leftrightarrow y=x+\frac{R(x,y,n)}{Q(x,y,n)},Q(x,y,n)\neq 0 \tag{12}$$

В формуле(12) исключаются все нули - корни уравнения $y:Q(x,y,n)=0$ при заданных целых ключах x,n. Рассмотрим алгебраические преобразования(12) на множестве целых положительных остатков Z_p: Шифром формулой(10) с ключами x,n символа y назовем пару целых чисел $\left(R(x,y,n)(\bmod p);Q(x,y,n)(\bmod p)\right)$

$$\left(R(x,y,n)(\bmod p);Q(x,y,n)(\bmod p)\right)=\left(\left(y^{2k+1}+x^{2k+1}\right)(\bmod p),\sum_{i=0}^{2k}x^{i}y^{2k-i}(\bmod p)\right),y\notin Y_0:Q(x,y,n)=0 \tag{13}$$

$$y\equiv x(\bmod p)+R(x,y,n)(\bmod p)\cdot Q^{-1}(x,y,n)(\bmod p)\,|\,Q^{-1}(x,y,n)(\bmod p)\cdot Q(x,y,n)\equiv 1(\bmod p) \tag{14}$$

Итак, алгоритм шифрования-дешифрования формулой (1) можно разбить на два случая:

1) $y\notin Y_0:Q(x,y,n)=0$. Шифрование по формуле(13), дешифрование по формуле(14).

2) $y\in Y_0:Q(x,y,n)=0$. Шифром назовем пару чисел $\left(R(x,y)\equiv x+y(\bmod p);Q(x,y)\equiv y-x(\bmod p)\right)$ (15)

Дешифрование проводим по формуле(16),отметим, что число $2^{-1}(\bmod p)$ существует, если p простое.

$$y\equiv R(x,y)+Q(x,y)\equiv x+y+y-x(\bmod p)=2y(\bmod p)\Leftrightarrow y=\left(R(x,y)+Q(x,y)\right)(\bmod p)\cdot 2^{-1}(\bmod p) \tag{16}$$

Усложним формулы (13)-(16), введем дополнительные целые ключи a,b и множитель $a^b\in N$

1) $y\notin Y_0:Q(x,y,n)=0$.Шифрование проводим по формуле(17), дешифрование по формуле(18).

$$\left(R(x,y,n)\pmod p;Q(x,y,n)\pmod p\right)=\left(a^b\cdot\left(y^{2k+1}+x^{2k+1}\right)\pmod p,a^b\cdot\sum_{i=0}^{n-1}x^iy^{n-1-i}\pmod p\right),y\notin Y_0:Q(x,y,n)=0 \quad (17)$$

$$y\equiv x\pmod p+R(x,y,n)\pmod p\cdot Q^{-1}(x,y,n)\pmod p\,|\,Q^{-1}(x,y,n)\pmod p\cdot Q(x,y,n)\equiv 1\pmod p \quad (18)$$

2) $y\in Y_0:Q(x,y,n)=0$. Шифром назовем пару чисел $\left(R(x,y)\equiv x+y\pmod p;Q(x,y)\equiv y-x\pmod p\right)$ (19)

Дешифрование проводим по формуле(20), p простое.

$$y=\left(R(x,y)+Q(x,y)\right)\pmod p\cdot 2^{-1}\pmod p \quad (20)$$

Ниже написана программа с использованием алгоритма(17)-(20) на языке FORTRAN, в которой шифруется символьная фраза s="Polotsk State University 1234567890" с ключами a=1119;b=131; n=10000;x=103. Простое число p=257 выбрано ближайшим простым к мощности клавиатуры ASCII равной 256. Допустимы и большие простые числа, чем 257, однако клавиатура ASCII f не сможет в консоли отобразить все символы шифра для произвольного сообщения, однако декодирование все равно производится верно.

```
program algebra
integer(8),parameter::n=599,p=257,len=40
integer(8)::x,y,z1,z2,z3,z4,z00,mas1(len+1),mas2(len+1),mas3(len+1)
integer(8)::a,b
character(len+2)::s
integer(8)::mas(len+2)
s="Polotsk State University 1234567890"
a=119;b=131
x=1;
do i=1,len
mas(i)=ichar(s(i:i))
print*,s(i:i),mas(i)
enddo
do i=1,len
y=mas(i)
call f1(x,y,n,p,a,b,z2)
if(z2==0)then
print*,"error"
z1=mod(x+y,p)
z2=mod(y-x,p)
call f3(2,p,z3)
z4=mod((z1+z2)*z3,p)
mas1(i)=z4
mas2(i)=0
elseif(.not.z2==0)then
call f3(z2,p,z3)
call f2(x,y,n,p,a,b,z00)
z4=mod(z3*z00-x,p)
if(z4<0)then
z4=z4+p
end if
mas1(i)=0
mas2(i)=z4
endif
mas3(i)=mas1(i)+mas2(i)
print*,"i=",i,mas3(i),char(mas3(i))
enddo
end program algebra
subroutine f1(x,y,n,p,a,b,z2)
integer(8)::a,b,d
integer(8)::i,j
integer(8)::z1,z,z2,z0,n,p,x,y,zz
z0=0;d=1;zz=-1
do i=1,b
d=mod(d*a,p)
enddo
do i=0,n-1
z1=1;z=1
```

```
do j=1,i
z=mod(z*x,p)
enddo
do j=1,n-i-1
z1=mod(z1*y,p)
enddo
zz=-zz
z0=mod((z0+z*z1*zz),p)
enddo
z2=mod(z0*d,p)
if(z2<0)then
z2=z2+p
endif
end subroutine
subroutine f2(x,y,n,p,a,b,z00)
integer(8)::a,b,c,d
integer(8)::i
integer(8)::z1,z2,z00,x,y,n,p
d=1
do i=1,b
d=mod(d*a,p)
enddo
z1=1;z2=1;
do i=1,n
z1=mod(z1*x,p)
z2=mod(z2*y,p)
enddo
z00=mod(z2+z1,p)
if(z00<0)then
z00=z00+p
endif
z00=mod(z00*d,p)
end subroutine
subroutine f3(t,p,z0)
integer(8)::t,p,i,z0
do i=1,p-1
if(mod(i*t,p)==1)then
z0=i
endif
enddo
end subroutine
```

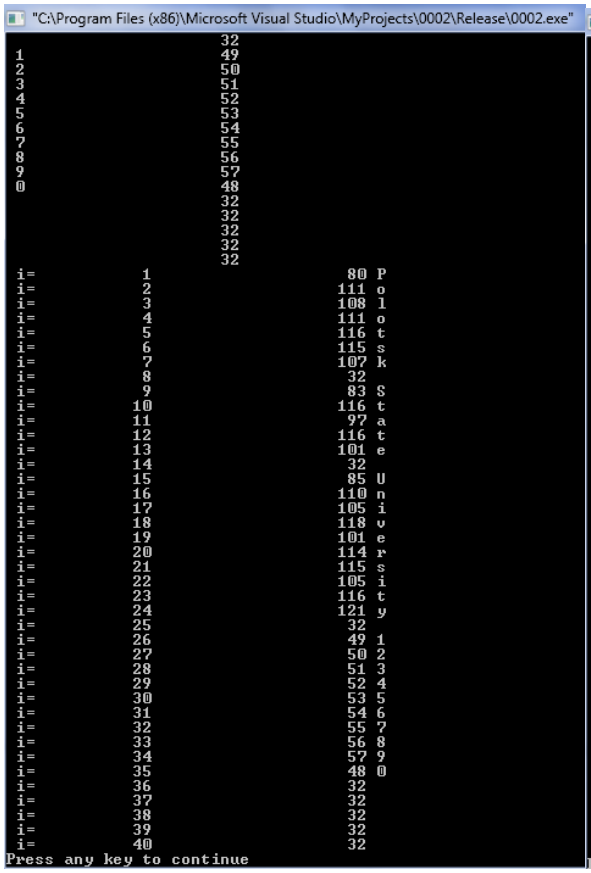

Рис.4. (n=10001) a=1119;b=131; x=103

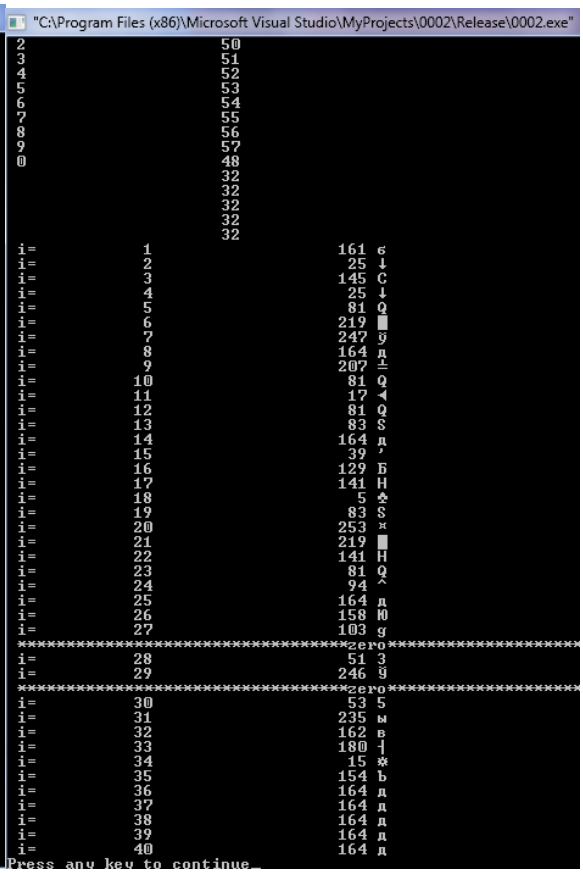

Рис.5.(n=10000) a=1119;b=131; x=103

На рисунках 3,4 показан пример шифрования фразы s="Polotsk State University 1234567890" с ключами a=1119;b=131; x=103; n=10001(Рис.4),n=10000(Рис.5). Рисунок 4 показывает, что ни при каких символах фразы при заданных ключах знаменатель дроби не содержит нулей $\forall y \notin Y_0 : Q(x, y, n) \neq 0$. Рисунки 4, 5 подтверждают, что шифрование формулой (11) возможно только при нечетных значениях степени, программа затрачивает на шифрование время 88 с. Дешифрование происходит значительно быстрее. Но чем дольше и сложнее алгоритм шифрования, тем сложнее работа крипто-аналитика!

Оценку пространства ключей и время его прямого перебора возьмем из примера 1. Мощность пространства $N \approx 256^3 \cdot 10000 = 1.677 \cdot 10^{11}$,за время $T \approx 1.677 \cdot 10^{11-3} c = 1.677 \cdot 10^8 c = 1941 суток \approx 5{,}3 лет$. При необходимости в алгоритм(7)-(10) можно добавить еще два ключа. Как и в примере 1 отдельные массивы – шифры двух алгоритмов на конечном этапе после дешифрования объединяются в один массив (складываются после дешифрования в один массив по правилам арифметики). Из рисунка 4 видно, что символы фразы s="Polotsk State University 1234567890" с числом знаков 40(лишние символы не входящие во фразу дешифруются пробелами – с порядковым номером символа 32) декодируются взаимно-однозначно.

Пример 3(Algebra 3).

Обозначим 12 ключей шифрования символами $A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$ (21)

Для каждой тройки последовательных символов фразы (x, y, z) найдем тройку символов шифрованной фразы (x_1, y_1, z_1) используя линейное (аффинное) отображение $Z_p \times Z_p \times Z_p \to Z_p \times Z_p \times Z_p$ по формуле (шифрования)

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ z \end{bmatrix} + b \Leftrightarrow \begin{cases} x_1 = a_{1,1}x + a_{1,2}y + a_{1,3}z + b_1 \\ y_1 = a_{2,1}x + a_{2,2}y + a_{2,3}z + b_2 \\ z_1 = a_{3,1}x + a_{3,2}y + a_{3,3}z + b_3 \end{cases} \quad (22)$$

Тогда, из формулы(22) получим исходную тройку символов

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = A^{-1} \begin{bmatrix} x_1 - b_1 \\ y_1 - b_2 \\ z_1 - b_3 \end{bmatrix} \Leftrightarrow \frac{1}{\Delta} \begin{cases} A_{1,1}(x_1 - b_1) + A_{1,2}(y_1 - b_2) + A_{1,3}(z_1 - b_3) \\ A_{2,1}(x_1 - b_1) + A_{2,2}(y_1 - b_2) + A_{2,3}(z_1 - b_3), \Delta \neq 0 \\ A_{3,1}(x_1 - b_1) + A_{3,2}(y_1 - b_2) + A_{3,3}(z_1 - b_3) \end{cases} \quad (23)$$

Где:

$$\Delta=\begin{vmatrix}a_{1,1} & a_{1,2} & a_{1,3}\\ a_{2,1} & a_{2,2} & a_{2,3}\\ a_{3,1} & a_{3,2} & a_{3,3}\end{vmatrix}=a_{1,1}\begin{vmatrix}a_{2,2} & a_{2,3}\\ a_{3,2} & a_{3,3}\end{vmatrix}-a_{1,2}\begin{vmatrix}a_{2,3} & a_{2,3}\\ a_{3,1} & a_{3,3}\end{vmatrix}+a_{1,3}\begin{vmatrix}a_{2,1} & a_{2,2}\\ a_{3,2} & a_{3,3}\end{vmatrix}=a_{1,1}\left(a_{2,2}a_{3,3}-a_{3,2}a_{2,3}\right)+a_{1,2}\left(a_{3,1}a_{2,3}-a_{2,3}a_{3,3}\right)+a_{1,3}\left(a_{2,1}a_{3,3}-a_{3,2}a_{2,2}\right)$$

$$\begin{cases}A_{1,1}=a_{2,2}a_{3,3}-a_{3,2}a_{2,3};A_{2,1}=a_{2,3}a_{3,1}-a_{2,1}a_{3,3};A_{3,1}=a_{2,1}a_{3,2}-a_{3,1}a_{2,2}\\ A_{1,2}=a_{3,2}a_{1,3}-a_{1,2}a_{3,3};A_{2,2}=a_{1,1}a_{3,3}-a_{1,3}a_{3,1};A_{3,2}=a_{1,2}a_{3,1}-a_{1,1}a_{3,2}\\ A_{1,3}=a_{1,2}a_{2,3}-a_{1,3}a_{2,2};A_{2,3}=a_{2,1}a_{1,3}-a_{1,1}a_{2,3};A_{3,3}=a_{2,2}a_{1,1}-a_{1,2}a_{2,1}\end{cases}\quad(24)$$

Тогда формула шифрования над полем целых остатков Z_p с учетом формулы(22) имеет вид

$$\begin{bmatrix}x_1\\ y_1\\ z_1\end{bmatrix}\pmod p\equiv A\begin{bmatrix}x\\ y\\ z\end{bmatrix}\pmod p+b\pmod p\Leftrightarrow\begin{cases}x_1\equiv a_{1,1}x+a_{1,2}y+a_{1,3}z+b_1\pmod p\\ y_1\equiv a_{2,1}x+a_{2,2}y+a_{2,3}z+b_2\pmod p\\ z_1\equiv a_{3,1}x+a_{3,2}y+a_{3,3}z+b_3\pmod p\end{cases}\quad(25)$$

Далее для дешифрования нужно вычислить таблицу вспомогательных чисел согласно(24)

$$\begin{cases}A_{1,1}\equiv a_{2,2}a_{3,3}-a_{3,2}a_{2,3}\pmod p;A_{2,1}\equiv a_{2,3}a_{3,1}-a_{2,1}a_{3,3}\pmod p;A_{3,1}\equiv a_{2,1}a_{3,2}-a_{3,1}a_{2,2}\pmod p\\ A_{1,2}\equiv a_{3,2}a_{1,3}-a_{1,2}a_{3,3}\pmod p;A_{2,2}\equiv a_{1,1}a_{3,3}-a_{1,3}a_{3,1}\pmod p;A_{3,2}\equiv a_{1,2}a_{3,1}-a_{1,1}a_{3,2}\pmod p\\ A_{1,3}\equiv a_{1,2}a_{2,3}-a_{1,3}a_{2,2}\pmod p;A_{2,3}\equiv a_{2,1}a_{1,3}-a_{1,1}a_{2,3}\pmod p;A_{3,3}\equiv a_{2,2}a_{1,1}-a_{1,2}a_{2,1}\pmod p\end{cases}\quad(26)$$

Найти обратное число к матрице системы(25)

$$\Delta^{-1}\pmod p:\Delta^{-1}\cdot\left(a_{1,1}\left(a_{2,2}a_{3,3}-a_{3,2}a_{2,3}\right)+a_{1,2}\left(a_{3,1}a_{2,3}-a_{2,3}a_{3,3}\right)+a_{1,3}\left(a_{2,1}a_{3,3}-a_{3,2}a_{2,2}\right)\right)\equiv1\pmod p\quad(27)$$

Если определитель матрицы системы(25) сравним с нулем $\Delta\equiv0\pmod p$, то изменить какие-нибудь из 9 ключей матрицы системы шифрования(21). Наконец, запишем формулы дешифрования аналогично формулам(23)

$$\begin{bmatrix}x\\ y\\ z\end{bmatrix}\equiv\Delta^{-1}\pmod p\cdot\begin{cases}A_{1,1}\left(x_1-b_1\right)+A_{1,2}\left(y_1-b_2\right)+A_{1,3}\left(z_1-b_3\right)\pmod p\\ A_{2,1}\left(x_1-b_1\right)+A_{2,2}\left(y_1-b_2\right)+A_{2,3}\left(z_1-b_3\right)\pmod p\\ A_{3,1}\left(x_1-b_1\right)+A_{3,2}\left(y_1-b_2\right)+A_{3,3}\left(z_1-b_3\right)\pmod p\end{cases}\Leftrightarrow$$

$$\Leftrightarrow\begin{cases}x\equiv\Delta^{-1}\pmod p\left(A_{1,1}\left(x_1-b_1\right)+A_{1,2}\left(y_1-b_2\right)+A_{1,3}\left(z_1-b_3\right)\right)\pmod p\\ y\equiv\Delta^{-1}\pmod p\left(A_{2,1}\left(x_1-b_1\right)+A_{2,2}\left(y_1-b_2\right)+A_{2,3}\left(z_1-b_3\right)\right)\pmod p,\Delta\neq0\pmod p\\ z\equiv\Delta^{-1}\pmod p\left(A_{3,1}\left(x_1-b_1\right)+A_{3,2}\left(y_1-b_2\right)+A_{3,3}\left(z_1-b_3\right)\right)\pmod p\end{cases}\quad(28)$$

Очевидно, что указанную последовательность алгоритма нужно дополнить способом разбиения исходной фразы на тройки символов. Если число символов сообщения не делится на три, то дополнить массив сообщения до ближайшего целого числа кратного трём. Избыточные символы массива сообщения (один либо два) при дешифровании окажутся пробелами с порядковым номером 32 и полезной информации не несут и не искажают исходное сообщение.

Ниже написана программа с использованием алгоритма(25)-(28) на языке FORTRAN, в которой шифруется символьная фраза s="Moscow State University 265 years" с ключами b1=123;b2=66;b3=38;a11=1;a12=2;a13=5;a21=23;a31=4;a22=5;a23=6;a32=7;a33=8.Простое число p=257 выбрано ближайшим простым к мощности клавиатуры ASCII равной 256. Допустимы и большие простые числа, чем 257, однако клавиатура ASCII f не сможет в консоли отобразить все символы шифра для произвольного сообщения, однако декодирование все равно производится верно.

```
program algebra
integer(8),parameter::p=257,kk=12,len=3*kk
integer(8)::x,y,z,z1,z2,z3,z4,z00,mas1(len+1),mas2(len+1),mas3(len+1)
integer(8)::a11,a12,a13,a21,a22,a23,a31,a32,a33,b1,b2,b3
integer(8)::da11,da12,da13,da21,da22,da23,da31,da32,da33,delta
integer(8)::xx,yy,zz,x1,y1,shifr(4,len+1),i,j,k,mas0(3,len+1)
character(len+2)::s
integer(8)::mas(len+2)
s="Moscow State University 265 years"
b1=123;b2=66;b3=38;
a11=3;a12=0;a13=5;a21=13;a31=4;a22=15;a23=7;a32=6;a33=18
da11=mod((a22*a33-a23*a32),p)
da21=mod((a31*a23-a21*a33),p)
da31=mod((a21*a32-a22*a31),p)
da12=mod((a32*a13-a12*a33),p)
```

```
da22=mod((a11*a33-a13*a31),p)
da32=mod((a12*a31-a11*a32),p)
da13=mod((a12*a23-a13*a22),p)
da23=mod((-a11*a23+a21*a13),p)
da33=mod((a22*a11-a12*a21),p)
delta=mod(a11*(a22*a33-a32*a23)-a12*(a21*a33-a31*a23)+a13*(a21*a32-a31*a22),p)
call f(delta,p,z3)
print*,mod(delta*z3,p)
if(.not.delta==0)then
print*,"delta=",delta
else
print*,"viberete drugie kluchi"
print*,"delta=",delta
end if
print*,"delta^-1=",z3
do i=1,len
mas(i)=ichar(s(i:i))
print*,i,s(i:i),mas(i)
enddo
k=0
print*,"*****************************shifr******************"
do i=1,len
if(mod(i,3)==1)then
x=mas(i)
k=k+1
elseif(mod(i,3)==2)then
y=mas(i)
k=k+1
elseif(mod(i,3)==0)then
z=mas(i)
k=k+1
endif
if(mod(k,3)==0)then
x1=mod((a11*x+a12*y+a13*z+b1),p)
y1=mod((a21*x+a22*y+a23*z+b2),p)
z1=mod((a31*x+a32*y+a33*z+b3),p)
shifr(1,k/3)=x1
shifr(2,k/3)=y1
shifr(3,k/3)=z1
print*,x1,y1,z1
endif
enddo
print*,"*****************************deshifr******************"
do i=1,len/3
x=z3*mod((da11*(shifr(1,i)-b1)+da12*(shifr(2,i)-b2)+da13*(shifr(3,i)-b3)),p)
y=z3*mod((da21*(shifr(1,i)-b1)+da22*(shifr(2,i)-b2)+da23*(shifr(3,i)-b3)),p)
z=z3*mod((da31*(shifr(1,i)-b1)+da32*(shifr(2,i)-b2)+da33*(shifr(3,i)-b3)),p)
x=mod(x,p)
y=mod(y,p)
z=mod(z,p)
if(x<0)then
x=x+p
endif
if(y<0)then
y=y+p
endif
if(z<0)then
z=z+p
end if
mas0(1,i)=x
mas0(2,i)=y
mas0(3,i)=z
print*,x,y,z
```

```
enddo
print*,"*****************************pause*******************"
!pause
do i=1,len/3
print*,char(mas0(1,i))
print*,char(mas0(2,i))
print*,char(mas0(3,i))
enddo
end program algebra
subroutine f(t,p,z0)
integer(8)::t,p,i,z0
do i=1,p-1
if(mod(i*t,p)==1)then
z0=i
endif
enddo
end subroutine
```

Оценим мощность пространства ключей. Учитывая, что каждый из 12 ключей можно выбирать независимо из 256 вариантов, то мощность искомого множества равна $N = 256^{12} \approx 7.92 \cdot 10^{28}$. Так как процессор крипто-аналитика не сможет перебирать ключи быстрее чем один набор за 1 такт процессора (одна миллиардная одной секунды), то получим полное время перебора $T \approx 7.92 \cdot 10^{28} \cdot 10^{-9} = 7.92 \cdot 10^{19} s = 2.51 \cdot 10^{12} years.$

На рисунках 6,7 показан пример шифрования фразы “Moscow State University 265 years” . На рисунке 6 видно, что в массиве программы шифр, а также дешифрованный текст хранится тройками в каждой строке, затем указанные векторы переписываются последовательно в один столбец и переводятся в символы таблицей ASCII. На рисунке 7 видно, что исходная фраза полностью дешифруется верно.

```
    18 e                  101
    19 r                  114
    20 s                  115
    21 i                  105
    22 t                  116
    23 y                  121
    24                     32
    25 2                   50
    26 6                   54
    27 5                   53
    28                     32
    29 y                  121
    30 e                  101
    31 a                   97
    32 r                  114
    33 s                  115
    34                     32
    35                     32
    36                     32
****************shifr*******************
   158                   196                   255
   244                   253                   158
    28                   226                   182
   148                   176                   113
   255                   214                    86
   172                    53                   157
   219                   153                   247
   117                    15                     5
    24                    98                   231
   210                   177                   140
   218                   244                    96
   122                   158                   163
****************deshifr*******************
    77                   111                   115
    99                   111                   119
    32                    83                   116
    97                   116                   101
    32                    85                   110
   105                   118                   101
   114                   115                   105
   116                   121                    32
    50                    54                    53
    32                   121                   101
    97                   114                   115
    32                    32                    32
****************pause*******************
```

Рис.6

```
                  97                   116                   101
                  32                    85                   110
                 105                   118                   101
                 114                   115                   105
                 116                   121                    32
                  50                    54                    53
                  32                   121                   101
                  97                   114                   115
                  32                    32                    32
*****************************pause*******************
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
M
o
s
c
o
w

S
t
a
t
e

U
n
i
v
e
r
s
i
t
y

2
6
5

y
e
a
r
s
```

Рис.7

Число символов равно 35, однако исходный массив расширяется до ближайшего целого кратного 3 - 36, то есть 36 символ по счету (не несущий полезной информации) идентифицируется как пробел с порядковым номером 32, что и видно из рисунка 7.

Отметим, что перемножение многочленов в примерах Algebra 1, Algebra 2 можно свести матричному кодированию или групповому кодированию (используется в примере Algebra 3). Матричное кодирование обладает рядом замечательных свойств[1]. Если пространство исходных слов обладает групповыми свойствами, то пространство шифрованных слов также является группой. Множество целых остатков по модулю p Z_p является полем, а операция сложения двух чисел по модулю всегда обратима. Следовательно, пространство слов шифра одинаковой длины образуют группу с операцией сложения по модулю p(в примере Algebra 3). Кроме того, если коэффициенты и аргументы многочленов из группы Z_2, то минимальное расстояние между шифрованными словами равно минимальному весу слова, отличного от нуля. Весом слова двоичного кода равен числу единиц в его записи или сумме всех его цифр[1].

Литература

1. Лидовский В.В. Теория информации: Учебное пособие. – М.: Компания Спутник +,2004. – 111 с. – ISSN 5-93406-661-7.
2. Пастухов Ю.Ф., Пастухов Д.Ф., Мередова М.Б. Динамическое кодирование: Учебное пособие. – Новополоцк: ПГУ,2019. – 19 с. (chttp:/ elib. psu.by:8080/ handle/123456789/23776).
3. Пастухов Д.Ф., Пастухов Ю.Ф., Сонич А.Д., Ченторицкий А.Ю. Обобщение метода шифрования Владимира Сизова на случай произвольных периодических и апериодических функций: Учебное пособие. – Новополоцк,ПГУ,2019. – 15 с. (http:/elib.psu.by:8080/handle/123456789/23747)
4. Пастухов Д.Ф., Пастухов Ю.Ф., Глебко И.С. Полиномиальное кодирование: Учебное пособие – Новополоцк, ПГУ,2019. – 23 с. (http:/elib.psu.by:8080/handle/123456789/23952)
5. Пастухов Д.Ф., Пастухов Ю.Ф., Смоляк А.И.. Шифрование гиперболическими функциями: Учебное пособие – Новополоцк, ПГУ,2018. – 33 с. (http:/elib.psu.by:8080/handle/123456789/22089)
6. Пастухов Д.Ф., Пастухов Ю.Ф., Синица П.Р. Шифрование данных на базе эллиптических кривых: Учебное пособие – Новополоцк, ПГУ,2016. – 72 с. (http:/elib.psu.by:8080/handle/123456789/16814)
7. Пастухов Д.Ф., Пастухов Ю.Ф., Зеленуха А.Ю. Шифрование с помощью нелинейных функций: Учебное пособие – Новополоцк, ПГУ,2017. – 43 с. (http:/elib.psu.by:8080/handle/123456789/16814)
8. Бартеньев О.В. Фортран для профессионалов. Математическая библиотека IMSL: Ч.1. – М.:ДИАЛОГ – МИФИ, 2001. – 448 с.
9. Бартеньев О.В. Фортран для профессионалов. Математическая библиотека IMSL: Ч.2. – М.:ДИАЛОГ – МИФИ, 2001. – 319 с.
10. Бартеньев О.В. Фортран для профессионалов. Математическая библиотека IMSL: Ч.3. – М.:ДИАЛОГ – МИФИ, 2001. – 368 с.

УДК 517. 66

Дмитрий Феликсович Пастухов
(Полоцкий университет)

Наталья Константиновна Волосова
(МГТУ им. Н.Э. Баумана)

Юрий Феликсович Пастухов
Серый Т.А.;Баталко И.И.; Василевич В.В; Смоляк А.И.
(Полоцкий университет)

АЛГЕБРАИЧЕСКИЕ МЕТОДЫ ШИФРОВАНИЯ

2020