РАДІОТЕХНІКА, ЕЛЕКТРОНІКА ТА ТЕЛЕКОМУНІКАЦІЇ

УДК 510.6

Л.П. БЕДРАТЮК, Г.І. БЕДРАТЮК
Хмельницький національний університет

ВИКОРИСТАННЯ СИСТЕМИ КОМП'ЮТЕРНОЇ АЛГЕБРИ MAPLE В КЛАСИЧНИХ КРИПТОСИСТЕМАХ

Представлено опис основних команд системи комп'ютерної алгебри Maple. Розглянуто реалізацію деяких класичних криптосистем засобами системи комп'ютерної алгебри Maple. Зокрема розглянуті процедури для криптосистеми Цезаря, афінної криптосистеми, криптосистеми Віженера та криптосистеми Хілла. Надано описи криптосистем і детальні пояснення до кожної з наведених процедур. Розглянуті пакети пропонується використовувати в ході вивчення дисциплін: дискретна математика, засоби захисту інформації, теорія кодування та криптографія.

Ключові слова: криптосистема, відкритий текст, шифротекст, ключ, функція шифрування, функція дешифрування, Maple, шифр Цезаря, афінний шифр, шифр Віженера, шифр Хілла.

L.P. BEDRATIUK, A.I. BEDRATIUK
Khmelnytskyi National University, Ukraine

USING OF THE COMPUTER ALGEBRA SYSTEM MAPLE FOR CLASSICAL CRYPTOSYSTEMS

We describe the basic commands computer algebra system Maple. Realization of some classical cryptosystems via the computer algebra system Maple. Specifically are considered. The procedures for he following classical cryptosystems are considered: the Caesar Cipher the affine cryptosystem, the Vigenere cryptosystem and the Hill cryptosystem. The detailed explanation of every cryptosystems and used procedures are given. The article can be useful in the study of some sections of discrete mathematics, data protection, coding theory and cryptography.

Keywords: cryptosystem, plaintext, ciphertext, key, encryption rule, decryption rule, Maple, Caesar cipher, afinne cipher Vigenere cipher, Hill's cipher.

Постановка проблеми. Останнім часом спостерігається активне проникнення систем комп'ютерної алгебри в освітній процес, що дозволяє формувати інноваційні технології навчання, зокрема математичного навчання в університетах [1–4]. Однією з найбільш популярних систем комп'ютерної алгебри є система Maple, фірми Waterloo Maple, Inc., яка успішно поєднує символьні маніпуляції, обчислювальну математику, потужну графіку та зручну мову програмування. В силу своєї зручності та універсальності система Maple стала незамінним інструментом наукових досліджень для студентів, інженерів та дослідників. Майже для кожного розділу сучасної математики в Maple розроблені окремі спеціалізовані пакети. Проте на даний час ці технології, незважаючи на свою ефективність та наочність, в силу різних причин, ще недостатньо поширені в навчальному процесі, що не сприяє інтеграції системи вищої освіти України у світовий простір вищої освіти. У статті описуються можливі варіанти реалізації класичних криптосистем в середовищі Maple. Розглянуто класичні криптосистеми: Цезаря, афінна криптосистема, криптосистема Віженера та криптосистема Хілла. Для цих систем написані процедури шифрування та дешифрування. Початкові навики роботи в системі комп'ютерної алгебри Maple, детально розглянуто в [5, 6], а основні поняття криптографії розглянуто в [7].

Матеріали статті можуть бути використані студентами та викладачами ВНЗ для розв'язання типових задач, які зустрічаються в процесі вивчення дисциплін "Захист інформації", "Дискретні структури", "Безпека програм та даних", "Теорія кодування та криптографія".

2. Аналіз джерел літературних даних або публікацій та постановка проблеми

Метою даної статті є реалізація в середовищі Maple основних типів класичних криптосистем. Початкові навики роботи в системі комп'ютерної алгебри Maple, детально розглянуто в [5, 6].

Криптографія це наука яка займається розробкою та вивченням методів конфіденційної та автентичної передачі інформації. Під відкритим текстом розуміються будь які данні призначені для передачі. Відкритий текст завжди можна подати у вигляді послідовності цілих чисел, причому цього можна досягнути багатьма способами, наприклад замінивши текстові символи їхніми ASCII-кодами. Під шифротекстом розуміється результат застосування до відкритого тексту деякого алгоритму шифрування, який залежить від додаткової інформації яка називається ключем. Для задавання криптосистеми потрібно задати алгоритми шифрування та дешифрування, а також задати ключі шифрування та дешифрування. Якщо ключі шифрування та дешифрування співпадають, то такі системи шифрування називаються симетричними.

Більш загально, криптосистемою називається п'ятірка (P,C,K,E,D), де P – множина відкритих текстів, C – множина шифротектів, K – множина ключів, E_k: $P \to C$ – функція шифрування і D_k: $C \to P$ – функція дешифрування, причому $D_k(E_k(x))=x$ для всіх відкритих текстів x і для всіх ключів k. Атака на криптоситему полягає в діях спрямованих на отримання ключа k, що дасть можливість дешифрувати будь-

який шифротекст. Яка саме система шифрування використовується вважається відомою, в силу принципу Керкгоффса. Дана стаття є продовженням статей авторів [8–10], спрямованих на популяризацію систем комп'ютерної алгебри.

Виклад основного матеріалу. Дамо короткий опис тієї частини мови програмування системи Maple та стандартних процедур, які необхідні для вирішення типових задач оптимізації.

Криптосистема Цезаря. Найпростішою і найдавнішою криптосистемою є шифр зсуву, або шифр Цезаря. Для цієї криптосистеми $P=C$, тобто множини шифротекстів і відкритих текстів співпадають і рівні таблиці символів ASCII. Простір ключів K є множиною натуральних чисел від 0 до 255. Функції шифрування і дешифрування визначаються такими формулами

$$E_k(x)=(x+k) \bmod 255,$$
$$D_k(x)=(x-k) \backslash mod\ 255.$$

Тут x є ASCII кодом відповідного символу.

Припустимо, нам потрібно зашифрувати шифром Цезаря текст "Приступайте до виконання місії завтра о 16:20". Спочатку визначаємо відповідну змінну в Maple:

```
> T:="Приступайте до виконання місії завтра о 16:20":
```

Команда Maple **convert**(S, bytes) перетворює рядок символів *S* в рядок із їхніх ASCII-кодів.

Формуємо список кодів *P,* який відповідає повідомленню *T*.

```
>P:=convert(T,bytes);
 T[1]=[207, 240, 232, 241, 242, 243, 239, 224, 233, 242, 229, 32, 228, 238,
32, 226, 232, 234, 238, 237, 224, 237, 237, 255, 32, 236,179, 241, 179, 191,
32, 231, 224, 226, 242, 240, 224, 32, 111,32, 49, 54, 58, 50, 48]
```

Повторне застосування цієї команди відновлює початковий текст:

```
>convert(P,bytes);
"Приступайте до виконання місії завтра о 16:20"
```

В оригінальному шифрі Цезаря зсув k був рівним 3.

Розглянемо процедуру шифрування **caesar**(x,k), яка зсовує число x на k позицій праворуч за модулем 255:

```
>caesar := proc(x, key) ((x + key - 1) mod 255) + 1 end;
```

Застосувавши цю команду до кожного символу відкритого тексту *P,* отримаємо шифротекст

```
> C:=convert(map(caesar,P,3),bytes);
C := " Тулфхцтгмхи#зс#елнсргрр#п¶ф¶В#кгехуг#г#49=53"
```

Тепер ми можемо написати процедуру **encode_caesar(message, key)** для перетворення відкритого тексту у шифротекст при наперед заданому ключі:

```
> encode_caesar := proc(plaintext, key)
 local t:
#перетворюємо повідомлення у набір цілих чисел
 t[1] := convert(plaintext, bytes):
#зсовуємо на розмір ключа
 t[2] := map(caesar, t [1], key):
#повертаємося назад у символьний вигляд
 convert(t [2], bytes);
 end proc:
```

Для декодування повідомлення використовуємо процедуру **decode_caesar(message, key)**

```
> decode_caesar := proc(ciphertext, key)
 local t:
#перетворюємо повідомлення у набір цілих чисел
 t [1] := convert(ciphertext, bytes):
#зсовуємо на розмір ключа
 t [2] := map(caesar, t[1], -key):
#повертаємося назад у символьний вигляд
```

```
 convert(t[2], bytes);
 end proc:
```

Шифр Цезаря має лише 256 можливих ключів, тому його можна легко зламати простим перебором.

Для атаки на криптосистему методом грубої сили і підбору ключа при відомому шифротексті використаємо процедуру:

```
> breakcaesar:= proc(ciphertext) local t, key;
 for key from 1 to 255 do
 t := encode_caesar(ciphertext, - key):
 print(`Ключ `||key,` повертає - `,t);
 end do;
 end proc:
```

Афінна криптосистема. Афінна криптосистема визначається такими параметрами $P=C=K=Z_{26}^{m}$. Для ключа $K=[k_1,k_2]$, де числа a і 255 взаємно прості, (a,255)=1, функції шифрування та дешифрування мають вигляд

$$E_k(x)=(k_1 x+k_2) \bmod 255,$$
$$D_k(x) =(x-k_2)\, k_1^{-1} \bmod 255$$

Для попередньої перевірки взаємної простоти k_1 і 255, тобто того чи їх найбільший спільний дільник чисел a і 255 використовуємо процедуру *gcd(a,b),* яка знаходить найбільший спільний дільник чисел a і b.

Процедура яка виконує шифрування має вигляд

```
>encode_Aff := proc(plaintext, key)
 local t,Aff:
#перевірка допустимості ключа
> if gcd(key[1],255)<>1 then print("змініть ключ"):break end if:
 #процедура для шифрування одного коду символу
 Aff:=proc(x, key) ((key[1]*x + key[2] - 1) mod 255) + 1 end proc;
#перетворюємо повідомлення у набір цілих чисел:
 t[1] := convert(plaintext, bytes):
#перетворюємо список кодів символів
 t[2] := map(Aff, t[1], key):
#повертаємося назад у символьний вигляд
 convert(t[2], bytes);
 end proc:
```

Приклад шифрування цієї процедурою відкритого тексту T з ключем K=[7,100]:

```
> encode_Aff(T,[7,100]);
                "□ъВ□□□уЉЙ□-Е¦мЕ□ВРмеЉeedЕЮN□NўЕ»Љ□□ъЉЕрЕjЯыГµ"
```

Для дешифрування повідомлення потрібно знаходити мультиплікативно обернений елемент кільця Z_{255}. Для цього використаємо команду `msolve(система рівнянь, модуль)`, яка розв'язує рівняння і системи рівнянь за модулем цілого числа. Відповідна процедура `inv_Aff` має вигляд

```
>   inv_Aff:=proc(x,key)local    a,y:a:=subs(msolve(key[1]*y=1,255),y):    (x-
key[2])*a mod 255 end proc:
```

Процедура для дешифрування має вигляд

```
> decode_Aff := proc(ciphertext, key)
 local t, Aff:
#перевірка допустимості ключа
> if gcd(key[1],255)<>1 then print("змініть ключ"):break end if:
inv_Aff:=proc(x,key)local  a,y:a:=subs(msolve(key[1]*y=1,255),y):  (x-key[2])*a
mod 255 end proc:
#перетворюємо повідомлення у набір цілих чисел
 t[1] := convert(ciphertext, bytes):
#зсовуємо на розмір ключа
```

```
 t[2] := map(inv_Aff, t[1], key):
#повертаємося назад у символьний вигляд
 convert(t[2], bytes);
 end proc:
```

Ця реалізація афінної криптосистеми має всього φ(255) ?255 =32640 ключів, де φ – функція Ейлера, тому вона також може бути легко зламана перебором ключів.

```
breakaffin:= proc(ciphertext) local t, key, key1,key2;
 for key1 from 1 to 255 do
 for key2 from 1 to 255 do
 key:=[key1,key2]:
 if gcd(key1,255)=1 then t := encode_Aff(ciphertext, key):
 print(`Ключ `||key,` повертає - `,t);
 end do;
 end proc:
```

Криптосистема Віженера. Криптосистема Віженера узагальнює криптосистему Цезаря тим що розбиває відкритий текст на блоки довжини m і до кожного символу блоку застосовується шифр Цезаря із своїм ключем. Формально криптосистема Віженера визначається параметрами $P=C=K=Z_{26}^m$. Для ключа $K=(k_1,k_2,\ldots,k_m)$, функції шифрування та дешифрування мають вигляд

$$E_k(x_1,x_2,\ldots,x_m)=(x_1+k_1,\ldots,x_m+k_m) \bmod 26$$
$$D_k(x_1,x_2,\ldots,x_m)=(x_1-k_1,\ldots,x_m-k_m) \bmod 26$$

Отже, криптосистема Віженера довжини m є прямим добутком m криптосистем Цезаря.

Для шифрування відкритого тексту чисел використовуємо процедуру

```
> encode_vigener := proc(message, key)
 local T:
#перетворюємо повідомлення у набір цілих чисел
 T[1] := convert(message, bytes):
#перетворюємо ключ у набір цілих чисел
 T[2] := convert(key, bytes):
#зсовуємо на розмір ключа
 T[3] := T[1]+T[2] mod 255:
#повертаємося назад у символьний вигляд
 convert(T[3], bytes);
 end;
```

Для дешифрування тексту використовуємо процедуру

```
>encode_vigener := proc(message, key)
 local T:
#перетворюємо повідомлення у набір цілих чисел
 T[1] := convert(message, bytes):
#перетворюємо ключ у набір цілих чисел
 T[2] := convert(key, bytes):
#зсовуємо на розмір ключа
 T[3] := T[1]-T[2] mod 255:
#повертаємося назад у символьний вигляд
 convert(T[3], bytes);
 end;
```

Криптосистема Хілла. Криптосистема Хілла узагальнює всі наведені раніше криптосистеми. Основна ідея криптосистеми полягає і тому що текст розбивається на блоки, як у криптосистемі Віженера, ключем є невироджена в кільці Z_{255} цілочисельна матриця, а процедури шифрування та дешифрування зводяться до операцій множення векторів на ключ на та обернену матрицю до ключа. Формально криптосистема Хілла задається такими параметрами: $P=C=K=Z_{127}^m$. Множина ключів співпадає із групою $GL(m, Z_{127})$ всіх невироджених квадратних матриць порядку m. Функції шифруванняі дешифрування для ключа(матриці) K і блоку тексту $(x_1,x_2,\ldots,x_m)$мають вигляд

$$E_K(x_1,x_2,\ldots,x_m)=(x_1,x_2,\ldots,x_m)K \bmod 127,$$
$$D_K(x_1,x_2,\ldots,x_m)=(x_1,x_2,\ldots,x_m)\,K^{-1} \bmod 127.$$

В Maple пакет Linear Algebra містить підпакет Modular для проведення обчислень у скінченних кільцях, який викликається командою

```
> with(LinearAlgebra[Modular]):
```

Ключ K формується рандомно процедурою

```
>Key:=m->Mod(127,Matrix(m,m,(i,j)->rand()),integer[]);
```

Наприклад

```
> K:=Key(3):
```

$$K := \begin{bmatrix} 60 & 64 & 126 \\ 6 & 26 & 39 \\ 54 & 110 & 116 \end{bmatrix}$$

Невирожденість ключа перевіряємо знаходячи його визначник

```
> Determinant(127,K);
```
66

Тепер знаходимо обернений ключ $M=K^{-1}$ відповідною командою пакету:

```
> M := Inverse(127,K);
```
$$M := \begin{bmatrix} 100 & 9 & 69 \\ 56 & 37 & 103 \\ 58 & 114 & 64 \end{bmatrix}$$

Для прикладу, перетворюємо повідомлення "abg" у вектор кодів:

```
> T:=map(x->x mod 127,convert("abg",bytes));
```

Для шифрування використаємо команду множення `multiply` з паркету `linalg`:

```
> with(linalg):C:=map(x->x mod 127,multiply(T,K));
```
$C := [32, 20, 52]$

Для дешифрування шифротексту C виконаємо множення на обернену матрицю M

```
 >map(x->x mod 127,multiply(C,M));
```
[97 , 98 , 103]

Переводимо отриманий рядок кодів у текст

```
> convert([97,98,103],bytes);
```
"abg"

Загальна процедура яка генерує пару ключ-обернений ключ має вигляд

```
> key_create:=proc(m) local K,M,Key:
 Key:=m->Mod(127,Matrix(m,m,(i,j)->rand()),integer[]):
K:=Key(3):
 if Determinant(127,K0 then print("спробуйте згенерувати ключ ще раз"):
break: end if:
 M := Inverse(127,K): return([K,M])
 end proc;
```

Наприклад

```
> Key:=key_create(3);
```
$$Key := \left[\begin{bmatrix} 74 & 33 & 58 \\ 6 & 103 & 30 \\ 81 & 123 & 65 \end{bmatrix}, \begin{bmatrix} 90 & 93 & 121 \\ 63 & 120 & 117 \\ 7 & 124 & 44 \end{bmatrix} \right]$$

Процедура шифрування відкритого тексту:

```
> endcode_Hill_1block:=proc(ciphretext) local T,C:
T:=map(x->x mod 127,convert(ciphretext,bytes)):
C:=convert(map(x->x mod 127,multiply(T,Key[1])),list):
 end proc;
```

Процедура дешифрування шифротексту

```
> decode_Hill_1block:=proc(ciphretext) local T,TT:
T:=map(x->x mod 127,multiply(ciphretext,Key[2])):
TT:=convert(T,bytes):TT
 end proc;
```

Для шифрування та дешифрування повідомлення перед застосуванням цих процедур повідомлення потрібно розбивати на блоки фіксованої довжини.

Висновки. В статті запропонована методика вивчення основ криптографії з використанням відомої системи комп'ютерної алгебри Maple. Такі підходи, на думку авторів, сприятимуть активізації навчально-пізнавальної діяльності студентів та підвищать ефективність організації їхньої самостійної роботи. Крім того вони внесуть новизну в традиційні методи навчання, які зараз характеризуються пасивністю та епізодичним безсистемним використанням інформаційних технологій. В статті дано варіант застосування основних команд пакетів `LinearAlgebra[Modular]` та `linalg` системи комп'ютерної алгебри Maple для ілюстрації роботи класичних криптосистем – криптосистеми Цезаря, афінної криптосистеми, криптосистеми Віженера та криптосистеми Хілла. Розглянуто типові приклади застосування розроблених процедур для шифрування да дешифрування повідомлень. Також дано короткий опис названих криптосистем.

Література

1. Blythab B. Using Maple to implement eLearning integrated with computer aided assessment / B. Blythab, A. Labovic // International Journal of Mathematical Education in Science and Technology. – 2009. – 40(7). – P. 975–988.

2. Chvatalova Z. Education of Economics with Maple / Z. Chvatalova, J. Hrebicek // Proceedings of the 30th international conference mathematical methods in economics, Karvin, Czech Republic, 2012. – P. 435–440.

3. Wiest L. The Role of Computers in Mathematics Teaching and Learning / Lynda R. Wiest // Computers in the Schools: Interdisciplinary Journal of Practice, Theory, and Applied Research. – 2001. – 17(1-2). – P. 41–55.

4. Adym E. The use of computers in mathematics education: A paradigm shift from "computer assisted instruction" towards "student rogramming" / E. Adym // The Turkish Online Journal of Educational Technology. – 2005. – 4(2). – P. 27–34.

5. Adams P. Introduction To Mathematics With Maple / Adams P., Smith K., Vyborny R. – World Scientific Pub Co Inc, 2004. – 544 p.

6. Васильев А. Н. Maple 8. Самоучитель / А.Н. Васильєв. – М. : Диалектика, 2003. – 352 с.

7. Мао В. Современная криптографія / Мао В. ; пер. с англ. – М. : Вильямс, 2005. – 768 с.

8. Бедратюк Л.П. Використання системи комп'ютерної алгебри Maple для розв'язання задач оптимізації / Л. Бедратюк, Г. Бедратюк // Вісник ХНУ. Технічні науки. – Хмельницький : ХНУ, 2014. – № 5(217). – С. 247–252.

9. Бедратюк Л.П. Computer algebra systems in graph theory / Л. Бедратюк, Г. Бедратюк // Eastern-European Journal of Enterprise Technologies. – 2012. – Vol. 6, N 4(60). – P. 43–46.

10. Бедратюк Л.П. Computer algebra systems in the elementary number theory / Л. Бедратюк, Г. Бедратюк // Eastern-European Journal of Enterprise Technologies. – 2013. – Vol. 6, N 4(66). – P. 10–13.

Рецензія/Peer review : 9.11.2015 р. Надрукована/Printed :5.12.2015 р.
Стаття рецензована редакційною колегією