# MATLAB-базиран модул за реализация на афинни шифри, прилагани в криптографските системи

Пламен Маноилов, Адриана Бороджиева

***Abstract: MATLAB-Based Module for Implementing Affine Ciphers Applied in Cryptosystems.*** *In this paper the principle of operation of the affine ciphers is given. The publication describes the algorithm and the program module using MATLAB, designed for encryption and decryption of English and Bulgarian texts using affine ciphers. The module may be used by students studying the course "Telecommunications Security" and in other disciplines dealing with cryptographic methods for information protection.*

***Key words:*** *cryptography, cryptosystems, affine ciphers, MATLAB.*

## ВЪВЕДЕНИЕ

Началото на криптографията е поставено още в далечното минало – от времето на древните гърци и спартанци, и император Юлий Цезар. По-нататъшното й развитие се стимулира основно от военното дело, дипломацията и разузнаването. В средните векове с проблемите на тайнописа са се занимавали Франсис Бейкън, Франсоа Виет, Джеронимо Кардано и др. Появили са се множество кодове и алгоритми, някои от които дълго време са били смятани за абсолютно сигурни, като например предложеният от френския дипломат Блез дьо Виженер криптографски алгоритъм. Интересни и надеждни криптографски средства са били използвани от известните разузнавачи на миналия век – Шандор Радо, Рихард Зорге, полковник Абел и др. Окончателното формиране на криптографията като научна дисциплина, занимаваща се с разработката на сигурни криптографски алгоритми, кодове и протоколи, както и на ефективни средства за тяхната реализация, може да се отнесе към средата на XX век [1, 2, 4].

## АФИННИ ШИФРИ, ИЗПОЛЗВАНИ В КРИПТОГРАФСКИТЕ СИСТЕМИ

Афинният шифър е вид моноазбучен субституционен шифър, в който всяка буква в азбуката се съпоставя на нейния цифров еквивалент, криптира се с помощта на проста математическа функция и се превръща обратно в буква. Като субституционен шифър, афинният шифър притежава слабостите на субституционните шифри. Всяка буква в английския език се шифрира с функцията $(ax+b) \mod (26)$, където $b$ е големината на преместването [3, 4].

### 1. Описание

В афинния шифър буквите на азбуката с размер $m$ най-напред се съпоставят на цели числа в диапазона $0 \ldots m-1$. След това се използва модулна аритметика, за да се трансформира цялото число, съответстващо на всяка буква в откритият текст, в друго цяло число, съответстващо на буква от шифрирания текст. Функцията за шифриране за една буква е:

$$E(x) = (ax+b) \mod m, \tag{1}$$

където модулът $m$ е размерът на азбуката, а $a$ и $b$ са ключът на шифъра. Стойността на $a$ трябва да бъде избрана така, че $a$ и $m$ да са взаимно-прости. Функцията за декриптиране е:

$$D(x) = a^{-1}(x-b) \mod m, \tag{2}$$

където $a^{-1}$ е мултипликативната инверсия на $a$ по модул $m$, т.е. тя удовлетворява уравнението:

$$1 = a.a^{-1} \mod m. \tag{3}$$

Мултипликативната инверсия на $a$ съществува само, ако $a$ и $m$ са взаимно-прости. Следователно, без ограничението върху $a$, декриптирането може да не е

възможно. Може да се покаже, че функцията за декриптиране е обратна на функцията за криптиране [3, 4]:

$$D(E(x)) = a^{-1}(E(x) - b) \mod m = a^{-1}(((ax + b) \mod m) - b) \mod m = \\ = a^{-1}(ax + b - b) \mod m = a^{-1}ax \mod m = x \mod m. \quad (4)$$

**2. Слабости**

Тъй като афинният шифър е едноазбучен субституционен шифър, той наследява слабостите на този клас шифри. Като се има предвид специфичният случай на криптиране на съобщения на английски език (т.е. $m = 26$ ), има общо 286 нетривиални афинни шифри, без да се броят 26-те тривиални шифъра на Цезар. Шифърът на Цезар е афинен шифър, за който $a = 1$, като функцията за криптиране се свежда до линейно преместване. Този брой следва от факта, че има 12 числа, по-малки от 26, които са взаимно-прости с 26 (това са възможните стойности на $a$ ). Всяка стойност на $a$ може да има 26 различни адитивни премествания (стойността на $b$ ); следователно, има 12 * 26 или 312 възможни ключа. Тази липса на разнообразие прави системата много несигурна, когато се разглежда в светлината на принципа на Kerckhoffs [3, 4]. За българската азбука, $m = 30$ , възможните варианти на афинни шифри са още по-малко, 8 * 30 или 240 възможни ключа.

Основната слабост на шифъра се дължи на факта, че ако криптоаналитикът може да открие открития текст на два символа от шифрирания текст, тогава ключът може да бъде получен чрез решаване на едно уравнение. Тъй като се знае, че $a$ и $m$ са взаимно-прости, това може да се използва за бързо отхвърляне на много „фалшиви" ключове в автоматизираната система.

**3. Описание на разработените приложения – алгоритъм и симулационни резултати**

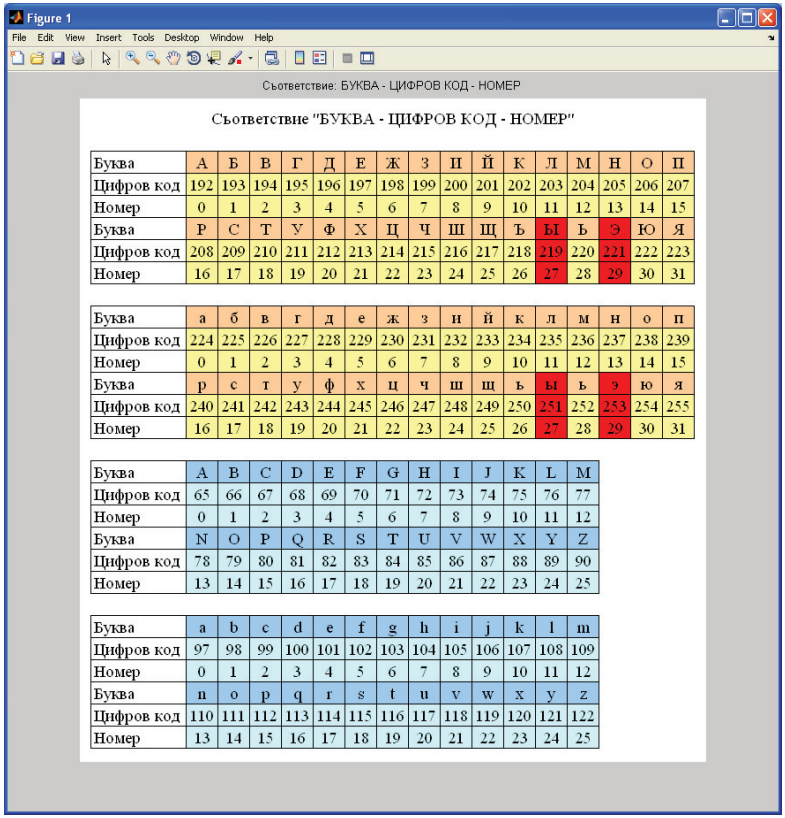
В [4] е описано разработено MS EXCEL-базирано приложение за шифриране и дешифриране на текстове на английски или български език, с малки и/или главни букви, чрез афинен шифър, с цел илюстриране на процесите на шифриране и дешифриране.

Разработени са приложения в MATLAB за шифриране и за дешифриране на текстове, с малки и/или главни букви, на английски, руски или български език, при използване на афинни шифри, с цел илюстриране на процесите шифриране и дешифриране. Разработените скриптове са предназначени за прилагане в учебния процес. Подходящи са за всяка дисциплина, по която се изучават криптографските методи за защита на информацията. По тази причина, скриптовете първоначално извеждат в отделни графични прозорци изображения за разясняване на разглеждания материал и по-лесното му усвояване от страна на обучаемите. Приложени са снимки от екрана при изпълнението на скриптовете с пример за шифриране или дешифриране на текст на английски език.

При стартиране на скрипта *ACencr_EN.m*, първоначално в отделни графични прозорци се извеждат изображения (във формат *.png*). Първото изображение в графичен прозорец Figure 1 представя съответствието „буква – цифров код – номер" за буквите от кирилицата и латиницата (фиг. 1), а второто – в графичен прозорец Figure 2 илюстрира процеса на шифриране на английски текст чрез афинен шифър (фиг. 2). Първият пример, който се разглежда, чрез скрипта за шифриране на английски текст, съвпада с примера от фиг. 2, с цел да се съпоставят двата резултата. В тестовия пример (фиг. 3) на MATLAB е предвидено и въвеждането на празен интервал между двете думи, подлежащи на шифриране.

Алгоритъмът за ***шифриране на текстове на английски език*** с малки и главни букви чрез афинен шифър, заложен в разработения скрипт, съдържа следните стъпки:

1. Въвеждане от клавиатурата на текста за криптиране (открития текст) на английски език с малки и/или с главни букви, и с възможност за празни интервали между думите, който ще се съхранява в стринговата променлива *str* (блок 1, фиг. 3).




Съответствие "БУКВА - ЦИФРОВ КОД - НОМЕР"

| Буква | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Цифров код | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| Номер | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Буква | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |
| Цифров код | 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| Номер | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |

| Буква | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Цифров код | 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| Номер | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Буква | р | с | т | у | ф | х | ц | ч | ш | щ | ъ | ы | ь | э | ю | я |
| Цифров код | 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 |
| Номер | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |

| Буква | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Цифров код | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 |
| Номер | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Буква | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| Цифров код | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| Номер | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

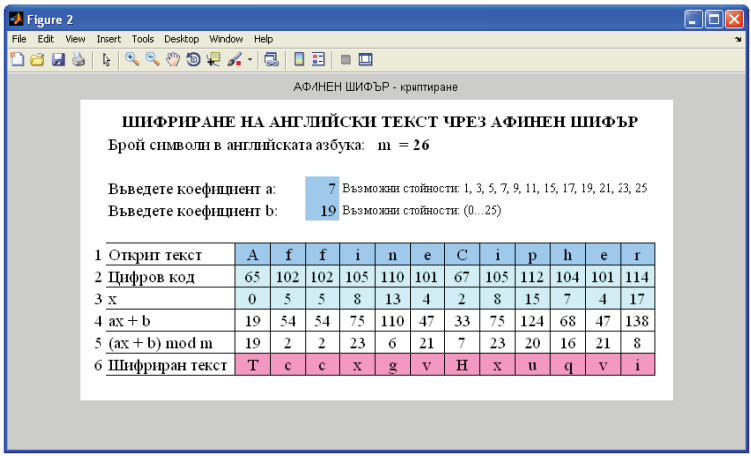
| Буква | a | b | c | d | e | f | g | h | i | j | k | l | m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Цифров код | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 |
| Номер | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Буква | n | o | p | q | r | s | t | u | v | w | x | y | z |
| Цифров код | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 |
| Номер | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

***Фиг. 1. Съответствие „буква – цифров код – номер” за буквите от кирилицата и латиницата***

2. Определяне и извеждане на цифровия код (7-битов ASCII-код) на всяка буква от открития текст чрез функцията *double*(*str*) (блок 2, фиг. 3).

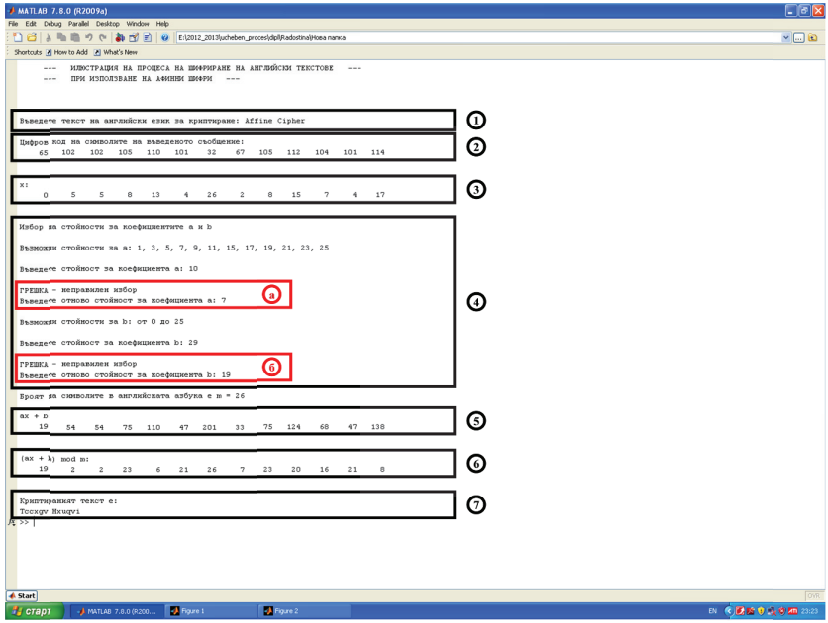
3. Определяне на номера на съответната буква в английската азбука, напр. буквата *A* има ASCII-код 65 (в десетична бройна система), но е прието да се номерира с 0. За целта, за всяко *I* = 1:*length*(*str*), се проверява дали буквата е малка, т.е. проверява се дали *double*(*str*(*i*)) >= 97, за да се извади числото 97 от десетичния й код, или е главна (ако *double*(*str*(*i*)) >= 95), за да се извади числото 65 от десетичния й код. Трябва да се отбележи, че главните букви в латиницата са с цифрови кодове от 65 до 90, а малките букви в латиницата са с кодове от 97 до 122. В случай, че *double*(*str*(*i*)) не е по-голямо или равно на 97, респ. 65, тогава се приема, че въведеният символ е „празен интервал” с ASCII-код „32”, който се номерира с 26. Резултатът се съхранява във вектор-ред *Vec*, обозначен на фигурата като вектор *x* (блок 3, фиг. 3). Следователно, изходът на тази операция е вектор *Vec* от неотрицателни цели числа в диапазона (0, 1,..., 25, 26) при стандартното съответствие:

0 ↔ *a*, *A*; 1 ↔ *b*, *B*; ...; 25 ↔ *z*, *Z*; 26 ↔ празен интервал.




| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Открит текст | A | f | f | i | n | e | C | i | p | h | e | r |
| 2 | Цифров код | 65 | 102 | 102 | 105 | 110 | 101 | 67 | 105 | 112 | 104 | 101 | 114 |
| 3 | x | 0 | 5 | 5 | 8 | 13 | 4 | 2 | 8 | 15 | 7 | 4 | 17 |
| 4 | ax + b | 19 | 54 | 54 | 75 | 110 | 47 | 33 | 75 | 124 | 68 | 47 | 138 |
| 5 | (ax + b) mod m | 19 | 2 | 2 | 23 | 6 | 21 | 7 | 23 | 20 | 16 | 21 | 8 |
| 6 | Шифриран текст | T | c | c | x | g | v | H | x | u | q | v | i |

***Фиг. 2. Илюстрация на процеса на шифриране на английски текст чрез афинен шифър***




***Фиг. 3. Шифриране на английския текст "Affine Cipher", с малки и главни букви, чрез афинен шифър с параметри a = 7; b = 19; шифриран текст: "Tccxgv Hxuqvi"***

**4.** Избор на параметрите на афинния шифър: *a* и *b*. Скриптът извежда подсказващо съобщение относно възможните стойности на двата параметъра (блок 4, фиг. 3) и прави проверка относно коректността на въвежданите данни. В случай, че не е въведена подходяща стойност (блок *а* и блок *б*, фиг. 3), се извежда съобщение за грешка и потребителят отново се подканва да въведе стойност на съответния параметър.

5. Определяне на (*ax* + *b*), където *x* е номерът на буквата в английската азбука (блок 5, фиг. 3).

6. Определяне на (*ax* + *b*) (*mod* 26), т.е. на остатъка при деление на (*ax* + *b*) с 26, ако *Vec*(*i*) е различно от 26, за всяко *I* = 1:*length*(*Vec*). Получените в резултат числа са в диапазона от 0 до 25 и се съхраняват във вектор-ред *VecOut*. Ако *Vec*(*i*) = 26, тогава *VecOut*(*i*) = *Vec*(*i*).

7. Преобразуване на цифровия код в символ, посредством функция *CHAR*. Тук отново, за всяко *I* = 1:*length*(*VecOut*), ако *VecOut*(*i*) < 26, се проверява дали буквата е малка (при *double*(*str*(*i*)) >= 97) или главна и при трансформирането ѝ в символ се добавя кодът на '*a*' (за малка буква) или кода на '*A*' (за главна буква), като позициите на празните интервали се запазват (чрез функцията *char*(32)). Резултатът се съхранява в променливата *StrOut* (блок 6, фиг. 3).

8. Извеждане на криптирания текст, съхранен в стринговата променлива *StrOut* (блок 7, фиг. 3).

По аналогичен начин може да се опише алгоритъмът за дешифриране чрез афинни шифри.

## ЗАКЛЮЧЕНИЕ

В публикацията е описан алгоритъм, заложен в програмен модул, реализиран в MATLAB, за шифриране и дешифриране чрез афинни шифри на текстове на английски, руски или български език, съставени от малки и/или главни букви. Приложени са резултатите от шифрирането и дешифрирането на произволен текст. Разработеният програмен модул може да намери приложение в учебния процес по дисциплината „Телекомуникационна сигурност", включена като задължителна в учебния план на специалността „Телекомуникационни системи" за образователно-квалификационна степен „Бакалавър", както и в други дисциплини, разглеждащи криптографските методи за защита на информацията.

Предвижда се разработване на аналогичен модул с графичен потребителски интерфейс с използване на възможностите на средата за разработване на графични потребителски интерфейси GUIDE за криптиране и декриптиране чрез афинни шифри на текстове на английски или български език, който да улеснява процесите на шифриране и дешифриране.

## ЛИТЕРАТУРА

[1]. Антонов, П., С. Малчев. Криптография в компютърните комуникации. Варна, 2000.

[2]. Скляр, Б. Цифровая связь. Теоретические основы и практическое применение. Москва, Вильямс, 2003.

[3]. http://en.wikipedia.org/wiki/Affine_cipher

[4]. Иванова, Р. MS EXCEL-базиран модул за реализация на афинни шифри, прилагани в криптографските системи. Студентска научна сесия – СНС '2012; Русе; 10.05.2012 г.; Сборник доклади на Студентска научна сесия – СНС'12; Русенски университет „Ангел Кънчев"; 79 – 83 стр.; ISSN 1311-3321.

**За контакти:**

Доц. д-р Пламен Маноилов, Катедра „Информатика и информационни технологии", Русенски университет „Ангел Кънчев", тел.: 082-888 646, e-mail: pmanoilov@ecs.uni-ruse.bg.

Гл.ас. д-р Адриана Бороджиева, Катедра „Телекомуникации", Русенски университет „Ангел Кънчев", тел.: 082-888 734, e-mail: aborodjieva@ecs.uni-ruse.bg.