УДК 004.056.55
Гринюк С.В., Поліщук М.М., Міскевич О.І., Харковець Р.В.
Луцький національний технічний університет

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ШИФРУВАННЯ ТА ДЕШИФРУВАННЯ ІНФОРМАЦІЇ КРИПТОГРАФІЧНИМИ МЕТОДАМИ ЗАСОБАМИ VISUAL STUDIO

Гринюк С.В., Поліщук М.М., Міскевич О.І., Харковець Р.В. Програмне забезпечення для шифрування та дешифрування інформації криптографічними методами засобами Visual Studio. У статті розглядаються проблеми захисту інформації в комп'ютерних системах та мережах. Дається класифікація алгоритмів шифрування та дешифрування інформації. На основі розглянутих алгоритмів шифрування було створено програмне забезпечення для шифрування та дешифрування файлів, використовуючи засоби Visual Studio.
Ключові слова: програмне забезпечення, криптографія, алгоритм, інформація.

Гринюк С.В., Полищук М.М., Мискевич О.І., Харковець Р.В. Программное обеспечение для шифрования и дешифрования информации криптографическими методами средствами Visual Studio. В статье рассматриваются проблемы защиты информации в компьютерных системах и сетях. Дается классификация алгоритмов шифрования и дешифрования информации. На основе рассмотренных алгоритмов шифрования было создано программное обеспечение для шифрования и дешифрования файлов, используя средства Visual Studio.
Ключевые слова: программное обеспечение, криптография, алгоритм, информация.

Hrynyuk S.V. Polishchuk M.M, Miskevich O.I., Kharkovets R.V. Software for encrypting and decrypting information by cryptographic methods with Visual Studio tools. This article deals with problems of information security in computer systems and networks. The classification of algorithms of encryption and decryption of information is given. Based on the encryption algorithms under consideration, software for encrypting and decrypting files was created using Visual Studio tools.
Key words: software, cryptography, algorithm, information.

Вступ. Однозначно та безперечно є той факт, що рівень розвитку будь якої держави та суспільства визначається рівнем їх інформатизації. У зв'язку з цим в Україні та й інших країнах світу новітні технології досить часто використовуються у всіх сферах діяльності людини. На даний момент не можливо представити своє життя без технологій. Сфера використання технологій досить широка. Всі підприємства, школи, банки, аеропорти, університети, атомні електростанції, різні міністерства використовують комп'ютери для працездатності, передаючи різного роду інформацію.

Постановка наукової проблеми. Основною проблемою є те, що необхідно захистити інформацію, яка передається між комп'ютерами. Досить недавно на прикладі вірусу Petya, всі люди переконались, що потрібно захищати свої системи, щоб ніхто в них не проникав. Також недавно виник такий термін як кібервійна. Для того щоб перемогти в ній, достатньо показати свою перевагу в інформаційній грамотності, в тому що ти вмієш захищати свої системи й знаєш як обійти системи захисту твого супротивника.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Вирішенням цієї проблеми є криптографія та алгоритми шифрування/дешифрування інформації. На основі алгоритмів шифрування я розробив програму за допомогою засобів Visual Studio на мові C# для шифрування та дешифрування файлів. Вона нам допоможе безпечно обмінюватись файлами між комп'ютерами. Використовувати ми будемо алгоритм шифрування AES.

Алгоритм шифрування AES використовується в досить популярному месенджері Telegram, в якому шифруються повідомлення проходячи кілька етапів. Враховуючи те, що Telegram був створений для обміну захищеними повідомленнями які важко перехватити, я вирішив створити прикладну програму на основі алгоритму AES, для шифрування файлів.

Advanced Encryption Standard (AES), також його часто називають Rijndael — симетричний алгоритм блочного шифрування (розмір блока 128 біт, ключ 128/192/256 біт). Прийнятий стандартом шифрування в США. AES має фіксовану довжину у 128 біт, а розмір ключа може приймати значення 128, 192 або 256 біт. Через фіксований розмір блоку AES оперує із масивом 4×4 байт. Для ключа 128 біт алгоритм має 10 раундів у яких послідовно виконуються операції:

SubBytes() – обробляє кожен байт стану, проводячи нелінійну заміну байтів використовуючи таблицю замін S-box. Це забезпечує нелінійність шифрування;

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Рис.1 - S-Box таблиця замін

Наприклад на вході маємо 54 то на виході будемо мати 20. Ця заміна буде повторюватись 10 разів. Після проходження всіх етапів які зазначені нижче.

ShiftRows(): це процедура перестановки байтів в рядках. Перший рядок залишається без змін, в другому рядку йде зміщення на байт, в наступному на два байта, а в останньому на три, як показано на наступному рис2:

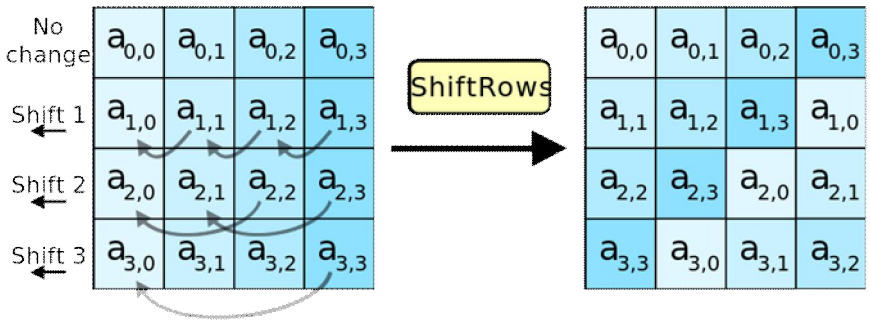

Рис. 2 – Процедура ShiftRows побітова заміна в рядках

У процедурі MixColumns , чотири байти кожної колонки статичних данних змішуються, використовуючи для цього зворотну лінійну трансформацію. MixColumns працює по колонках, трактуючи кожну з них як рівняння четвертого степеня. Над цими рівнянням виконується множення в GF(2^8) по модулю x^4+1 x^4+1

на фіксований многочлен $C(x) = 3x^2 + x^2 + x + 2$ та разом з ShiftRows MixColumns вносить

це що вийде в шифр.

Під час цієї операції, кожен стовпчик множиться на матрицю, яка для 128-бітного ключа має вигляд:

2 3 1 1
1 2 3 1
1 1 2 3
3 1 1 2

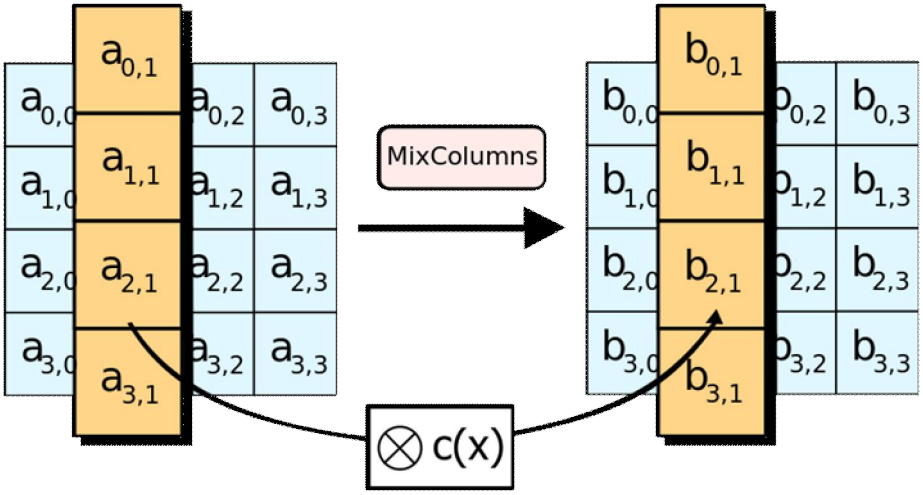

Рис. 3 - Приклад роботи процедури MixColumns

У процедурі AddRoundKey , RoundKey кожного раунду об'єднується зі статичними данними. Для кожного раунду Roundkey виходить з шифру ключа використовуючи процедуру KeyExpansion ; кожен RoundKey такого ж розміру, що і статичні дані. Фактично це звичайний побайтовий XOR байт ключа з байтами таблиці статичних данних як показано на рисунку 3.

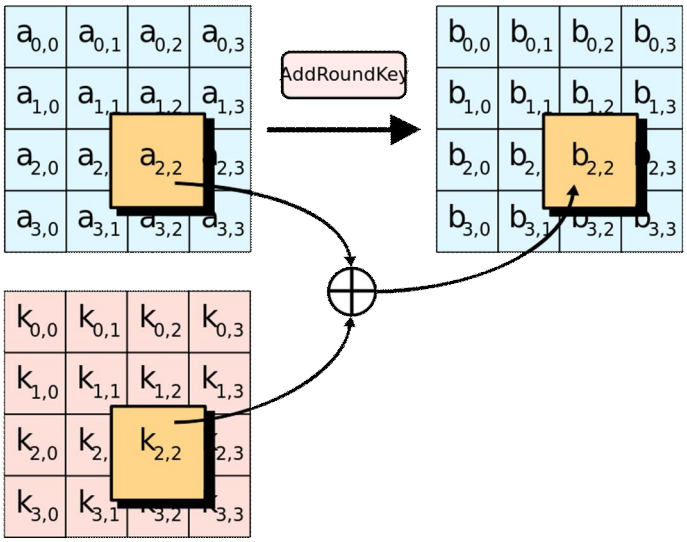

Рис. 4 - Процедура AddRoundKey

Робота алгоритму шифрування закінчиться тоді, коли пройде 10 разів всі ці процедури. В кінці ми отримуємо зашифрований файл, який без ключа не можливо буде розшифрувати. Знаючи ключ ми зможемо запустити алгоритм розшифрування.

Спираючись на цей алгоритм була розроблена програма CryptoCode 1.0 (Рис.5).

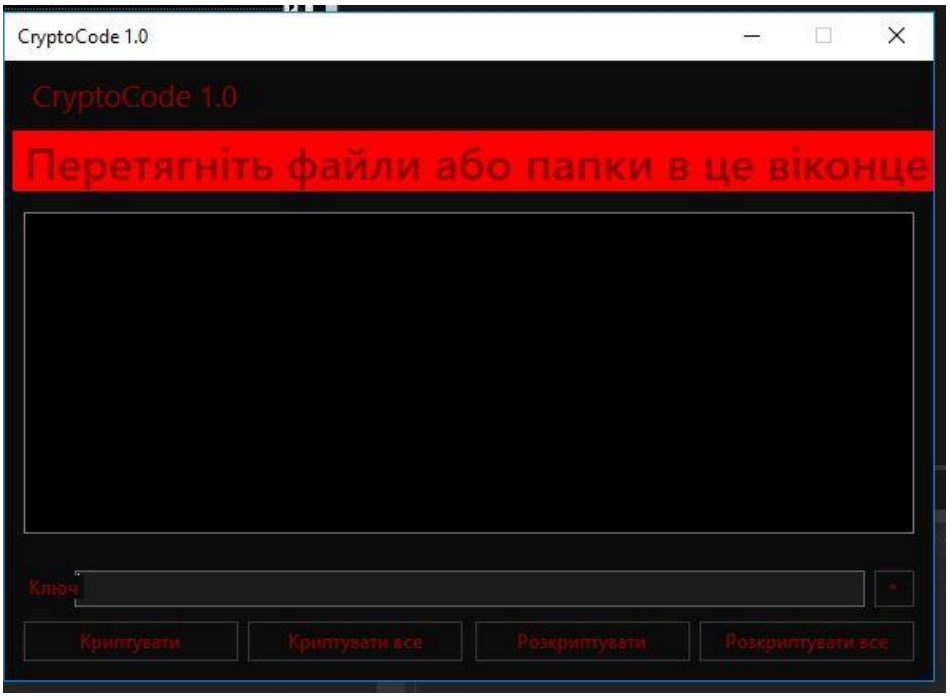

Рис. 5 - Вікно програми CryptoCode 1.0

Щоб реалізувати AES програмно потрібно підключити простір імен для криптографії:

```
using System.Security.Cryptography;
```

Приклад AES шифрування реалізували так :

```
   public static byte[] Encrypt(byte[] rawBytes, string password)
        {
            RijndaelManaged AES = new RijndaelManaged();
            byte[] hash = new byte[32];
            byte[] temp = new
MD5CryptoServiceProvider().ComputeHash(Encoding.ASCII.GetBytes(password));
            System.Buffer.BlockCopy(temp, 0, hash, 0, 16);
            System.Buffer.BlockCopy(temp, 0, hash, 15, 16);
            AES.Key = hash;
            AES.Mode = CipherMode.ECB;
            ICryptoTransform AESEncrypter = AES.CreateEncryptor();
            return AESEncrypter.TransformFinalBlock(rawBytes, 0, rawBytes.Length);
        }
```

Розшифрування всього алгоритму виконано так :

```
public static byte[] Decrypt(byte[] rawBytes, string password)
        {
            RijndaelManaged AES = new RijndaelManaged();
            byte[] hash = new byte[32];
            byte[] temp = new
MD5CryptoServiceProvider().ComputeHash(Encoding.ASCII.GetBytes(password));
```

```
        System.Buffer.BlockCopy(temp, 0, hash, 0, 16);
        System.Buffer.BlockCopy(temp, 0, hash, 15, 16);
        AES.Key = hash;
        AES.Mode = CipherMode.ECB;
        ICryptoTransform AESDecrypter = AES.CreateDecryptor();
        return AESDecrypter.TransformFinalBlock(rawBytes, 0, rawBytes.Length);
    }
```

Перетягуємо файл який хочемо зашифрувати у вікно програми, вводимо ключ (хоча це не обов'язково) й натискаємо кнопку «Криптувати» й отримуємо зашифрований файл:

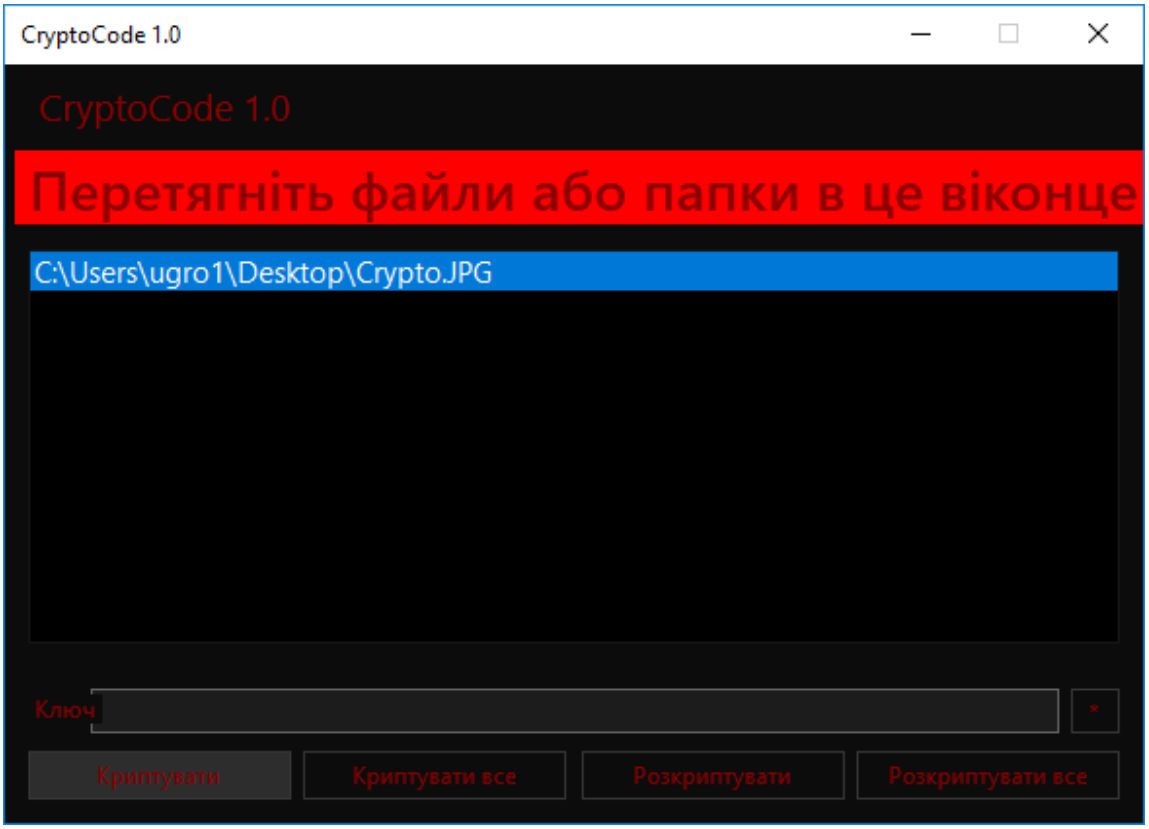

Рис. 6 - Вікно програми та зашифрований файл

Перетягуємо зашифрований файл назад у вікно програми вводимо в поле «Ключ» правильний ключ, який ми вказували перед шифруванням, і нажимаємо «Розкриптувати». Як можемо бачити на Рис.7, що розшифрувавши, ми отримали файл з розширеням .JPG .

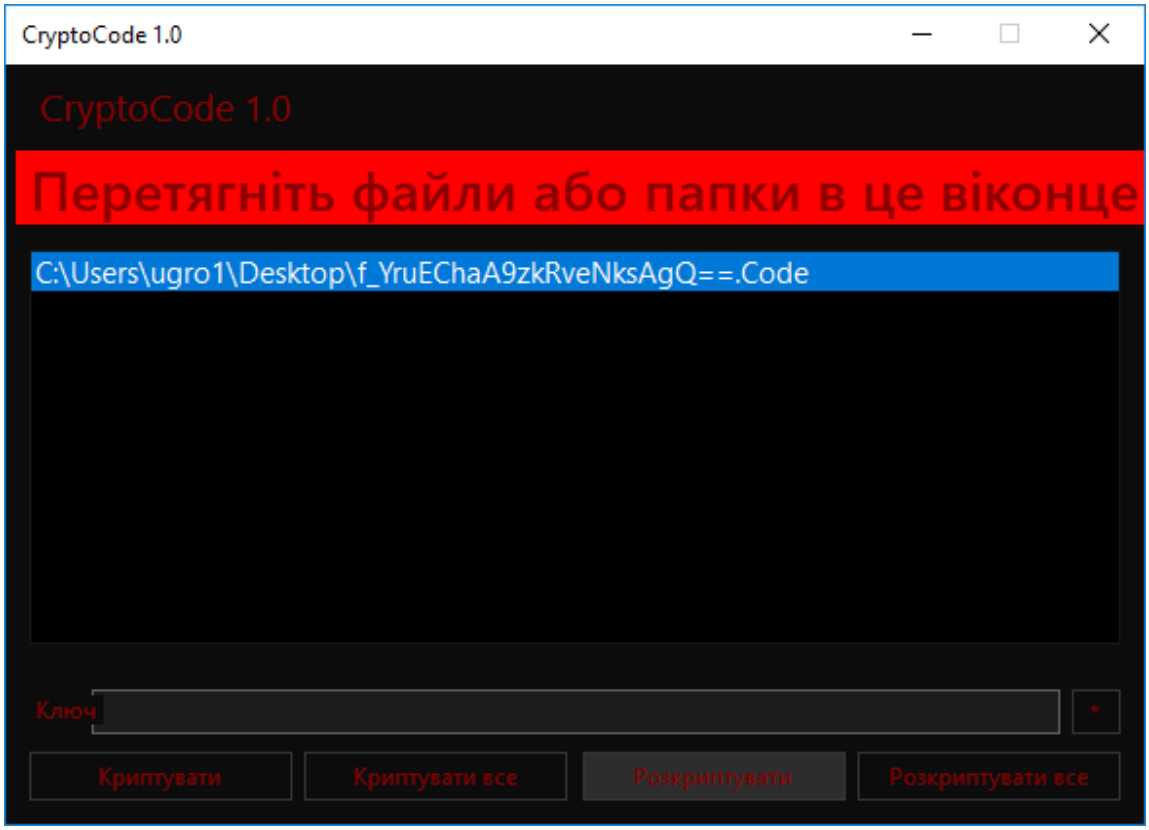

Рис. 7 - Розшифрування зашифрованого файла

Також основною особливістю програми є те, що при введені невірного ключа зашифрований файл видаляється з комп'ютера. Можна також підмітити те, що зашифрований файл можна ще пару раз зашифровувати змінюючи ключ.

Висновок. У підсумку можна сказати те, що необхідно захищати файли, системи, мережі від несанкціонованого доступу. Алгоритм шифрування AES, який є стандартом в США дозволяє захистити необхідну інформацію. Захист інформації є перспективним напрямком, оскільки це забезпечує тебе впевненістю в тому, що твоє приватне не стане публічним.

1. Баричев С. Г., Гончаров В. В., Серов Р. Е. 2.4.2. Стандарт AES. Алгоритм Rijdael // Основы современной криптографии — 3-е изд. — М.: Диалог-МИФИ, 2011. — С. 30–35. — 176 с.
2. Mollin, R. A. Introduction to Cryptography. — CRC Press, 2007. — P. 80. — 413 p.
3. https://msdn.microsoft.com/ru-ru/library/system.security.cryptography.aes(v=vs.110).aspx
4. https://uk.wikipedia.org/wiki/Advanced_Encryption_Standard