AN APPROACH FOR BUILDING SYMMETRIC CRYPTOGRAPHIC ALGORITHMS FOR VIDEO ENCRYPTION IN MATLAB SOFTWARE

GEORGI G. DIMITROV

ABSTRACT: *In this article, an approach for processing video files in the MATLAB software environment is shown. This approach gives the opportunity to build symmetric cryptographic algorithms to protect this type of files in order to ensure their secure storage in computer systems and safe transfer over computer networks.*

KEYWORDS: *symmetric cryptographic algorithms, video encryption, video processing in MATLAB.*

ПОДХОД ЗА ИЗГРАЖДАНЕТО НА СИМЕТРИЧНИ КРИПТОГРАФСКИ АЛГОРИТМИ ЗА КРИПТИРАНЕ НА ВИДЕО ФАЙЛОВЕ В ПРОГРАМНАТА СРЕДА НА MATLAB

ГЕОРГИ Г. ДИМИТРОВ

АБСТРАКТ: *В настоящата статия е показан подход за обработка на видеофайлове в програмната среда на MATLAB. Този подход дава възможност за изграждане на симетрични криптографски алгоритми за защита на този вид файлове, с цел да осигури тяхното сигурно съхраняване в компютърни системи и безопасен трансфер в компютърни мрежи.*

КЛЮЧОВИ ДУМИ: *симетрични криптографски алгоритми, видео криптиране, видео обработка в MATLAB*

1 Въведение

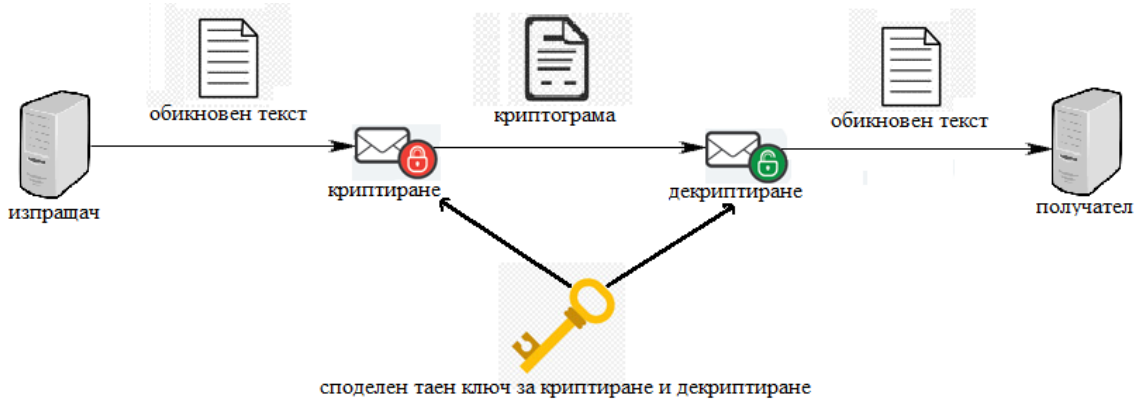
В днешно време развитите технологии позволяват много бърза комуникация. Те позволяват обмен на информация във вид на текст, снимки или видео буквално за секунди. Криптографията дава възможност на важните съобщения да останат тайни за всички, за които не са предназначени, дори и да попадне в техни ръце. Криптографските алгоритми[1,2] могат да бъдат класифицирани в три основни групи: алгоритми със секретен ключ (симетрични), алгоритми с публичен ключ (асиметрични), алгоритми без ключ. В този доклад ще бъде разгледана същността на симетричните криптографски алгоритми и начинът им за прилагане върху видео файлове в програмната среда на MATLAB.

MATLAB е софтуерен пакет, създаден от Math Works Inc.[3] Неговите възможности за аналитични преобразувания и числени пресмятания с помощта на вградените математически функции го правят добър инструмент при разработването на криптографски алгоритми, които се базират именно на математически операции.

2 Симетрични криптографски алгоритми

Симетричните криптографски алгоритми[4,5] използват един и същ ключ за криптиране (шифриране) и декриптиране (дешифриране) на информацията. Ключът трябва да е известен и на криптиращата и на декриптиращата страна, затова той трябва да се съхранява на надеждно място, да се разпространява внимателно и периодично да се обновява с цел подобряване на сигурността. Фигура 1 представя графично пътят на едно

съобщение, осъществен със симетрична криптографска система – структура, имаща за цел да защити информацията от неправомерен достъп и включваща набор от симетрични криптографски алгоритми за криптиране, декриптиране и генериране на ключове.



Фигура 1. Комуникация, осъществена чрез симетрична криптографска система

3 Видео криптиране в програмната среда на MATLAB

MATLAB предоставя много възможности за работа с различни типове файлове[6]. Една от тях е обработката на видео файлове[7], към която може да се приложат различни методи за криптиране.

Изображенията са основната структурна единица на всеки видео файл. Всеки кадър е съставен от пиксели с различен цвят, а всеки пиксел представлява комбинация от цветовете червено, зелено и синьо. Ако се разгледа пиксела като 24-битова стойност, както е при цветните изображения, то той ще има три последователни стойности с големина 8 бита в диапазона от 0 до 255, характеризиращи съответно червеното, зеленото и синьото (RGB).

При шифрирането на видео файла самото създаване на тайния ключ може да се осъществи в самия криптографски алгоритъм с помощта на псевдослучайни генератори [8-11], които генерират последователности от битове, като така се избягва нуждата от съхранението на ключа. При използването на симетричен криптографски алгоритъм се използва един и същ ключ за криптиране и декриптиране, затова е достатъчно да се изпълни същия алгоритъм върху криптограмата, за да получим отново видео файла в първоначалния му вид.

MATLAB разполага с обектите VideoReader и VideoWriter, с чиято помощ може да се осъществи криптирането на видео файл.

Първата стъпка е „прочитането“ на входящ видеофайл и създаването на нов празен видеофайл по следния начин:

reader = VideoReader (‘input.avi’);
writer = VideoWriter (‘output.avi’, ‘Uncompressed AVI’);

По този начин са избрани имената на входния и изходния файл - съответно *input.avi* и *output.avi*, а за вид компресия на новосъздадения видео файл е избран – *Uncompressed AVI*.

Важно е да се отбележи, че ако не се посочи вид компресия за запис на видео файла, по подразбиране MATLAB ще избере *Motion JPEG AVI*.

Следващата стъпка е да изравним кадрите в секунда на видео файловете – входния файл и резултатния криптиран/декриптиран файл. Тази стъпка е необходима за да се запази същата стуктурата при крайния видеофайл. Приравняването на кадрите в секунда се извършва, чрез:

```
writer.FrameRate = reader.FrameRate;
```

За обработка на всеки кадър е необходимо да се извлече общия брой на кадрите от началното видео, което се извършва чрез:

```
frames = get (reader, 'NumberOfFrames');
```

Обработката на новосъздадения видео файл за запис (в който се извършва криптиране/декриптиране) се осъществява чрез:

```
open (writer);
```

Прихващането на кадър от видеофайл се извършва чрез:

```
frame = read (reader, 1);
```

Вторият параметър във функцията *read*, показва номерът на кадърът, който се обработва.

Прихванатият кадър се третира като статично растерно изображение, което се обработва при процеса на криптиране или декриптиране. Растерните изображения са структурирани по специфичен начин, като изграждащите структурни единици се наричат пиксели, които се идентифицират с местоположение в изображението (номер на ред и номер на колона) и цветова стойност (състояща се от стойност за червен, зелен и син цвят). С два оператора за цикъла *for* се обхождат всички пиксели на съответния кадър, като за всеки пиксел извличаме стойностите на червено, зелено и синьо в битове (RGB) и се променя тяхната цветова стойност с помощта на криптографски алгоритъм по избор, като по този начин криптираме съдържанието на кадъра:

```
for j = 1:rows
  for i = 1:columns
        R = frame (j, i, 1); %червено
        G = frame (j, i, 2); %зелено
        B = frame (j, i, 3); %синьо
```

```
            frame (j, i, 1) = getNewValue(R);
            frame (j, i, 2) = getNewValue(G);
            frame (j, i, 3) = getNewValue(B);
        end
    end
```

След промяна на цветовите стойности на всички пиксели, новополученият кадър се записва в изходния видеофайл чрез:

writeVideo (writer, frame);

Тези операции се прилагат за всички кадри на видеофайлът. Прихващането на всички кадри може да се извърши отново чрез оператор за цикъл *for*, като за начало се използва стойност 1, а за крайна стойност - *frames*. След като обработката на всички кадри, затварянето и записването на крайният видеофайл се извършва чрез:

close(writer)

По този начин се извършва криптиране / декриптиране на видео файл с помощта на симетричен криптографски алгоритъм в програмната среда на MATLAB. Използвайки симетричен криптографски алгоритъм, трябва да имаме в предвид следното:

1) При криптиране: входният файл е оригиналният видео файл, а изходният файл – криптираният видео файл (криптограмата).
2) При декриптиране: входният файл е криптираният видео файл (криптограмата), а изходният файл – оригиналният видео файл

4 Заключение

Реализацията на криптографски алгоритми е свързана с използването на програмни среди за обработка на информацията. В настоящата статия е демонстриран подход за изграждане на симетрични криптографски алгоритми за защита на видеофайлове, чрез използване на програмната среда MATLAB. Демонстрирани са методите за видео обработка със средствата на математическия софтуер, като са описани стъпките за прихващане на кадрите на видеофайловете и обработката на пикселите на всеки кадър. При модификацията на пикселите могат да се приложат криптографски методи за промяна на местоположението на пикселите и промяна на тяхната цветова стойност.

ЛИТЕРАТУРА:

[1] Нонинска, И., Криптография (Алгоритми и протоколи за защита на информацията в компютърните системи). Технически университет - София (2005).

[2] Тужаров Хр., Архитектура на сигурността. Асеневци (2010).

[3] URL: https://www.mathworks.com/products/matlab.html - официален сайт на MATLAB (посетен на 20.09.2020)

[4] Delfs H., Knebl H., Symmetric-Key Cryptography. In: Introduction to Cryptography. Information Security and Cryptography. Springer, Berlin, Heidelberg (2015), 11-48.

[5] Bokhari, M.U., Shallal, Q.M., A Review on Symmetric Key Encryption Techniques in Cryptography. In: International Journal of Computer Applications (0975-8887), Vol. 147 – No.10, August (2016).

[6] Gilat, A., MATLAB: An Introduction with Applications. Wiley, 5th Edition (2014).

[7] Marques, O., Practical Image and Video Processing Using MATLAB. Wiley-IEEE Press (2011).

[8] Kordov, K. Modified pseudo-random bit generation scheme based on two circle maps and XOR function. Applied Mathematical Sciences, 9(3), 129-135 (2015).

[9] Kordov, K. M. Modified Chebyshev Map Based Pseudo-random Bit Generator. In AIP Conference Proceedings, Vol. 1629, 432-436 (2014).

[10] Kordov, K. Signature attractor based pseudorandom generation algorithm. Advanced Studies in Theoretical Physics, 9(6), 287-293 (2015).

[11] Kordov, K. A novel audio encryption algorithm with permutation-substitution architecture. Electronics, 8(5), 530 (2019).