هذا الكود هو نظام بسيط لتشفير الصور باستخدام تسلسل فيبوناتشي.

ملف Main.m:

يتم قراءة صورة من ملف الصور باستخدام imread.

يتم تشفير الصورة باستخدام دالة encryptImage الموجودة في ملف encryptImage.m.

يمكن إضافة عمليات إضافية أو تحليلات بعد عملية التشفير.

دالة generateFibonacci:

هي دالة تقوم بإنشاء تسلسل فيبوناتشي بطول n.

يتم بدء التسلسل بالقيم 1 و 1، ومن ثم يتم حساب القيم اللاحقة باستخدام العلاقة التراكمية.

ملف encryptImage.m:

دالة encryptImage تحدد نوع الصورة (ملونة أم رمادية) باستخدام ndims وتقوم بتوجيه الصورة إلى الدالة المناسبة.

إذا كانت الصورة ملونة، يتم استدعاء encryptColorImage.

إذا كانت الصورة رمادية، يتم استدعاء encryptGrayscaleImage.

دالة encryptColorImage:

تقوم بتشفير الصور الملونة.

تقوم بتحويل الصورة إلى مصفوفة 1D باستخدام reshape.

تقوم بإنشاء تسلسل فيبوناتشي بناءً على حجم الصورة الملونة.

يتم استخدام bitxor لتنفيذ العملية الرئيسية للتشفير.

الصورة المشفرة تُعيد بعد تغيير شكلها إلى الحجم الأصلي باستخدام reshape.

دالة encryptGrayscaleImage:

تقوم بتشفير الصور الرمادية بطريقة مشابهة لـ encryptColorImage.

الفرق الرئيسي هو أنها تعمل على الصور الرمادية مباشرة دون تحويلها.

الكود يقوم بعملية بسيطة لتشفير الصورة باستخدام تسلسل فيبوناتشي وعملية XOR. يُظهر الكود الصورة الأصلية والصورة المشفرة جنبًا إلى جنب للمقارنة.

**ملف Main.m**

inputImage = imread('img1.jpg');

encryptedImage = encryptImage(inputImage);

disp('Encryption process completed.');

* يتم قراءة الصورة من ملف 'img1.jpg' باستخدام الدالة imread.
* يتم تشفير الصورة باستخدام الدالة encryptImage الموجودة في ملف encryptImage.m.
* يمكن إضافة عمليات إضافية أو تحليلات بعد عملية التشفير.
* يتم عرض رسالة "Encryption process completed." لإشعار المستخدم بانتهاء عملية التشفير.

**ملف encryptImage.m**

function encryptedImage = encryptImage(inputImage)

if nargin < 1

error('Input image is required.');

end

if ndims(inputImage) == 3

disp('Input image is RGB.');

encryptedImage = encryptColorImage(inputImage);

else

disp('Input image is grayscale.');

encryptedImage = encryptGrayscaleImage(inputImage);

end

end

* تأكيد على توفر الصورة المدخلة.
* فحص نوع الصورة (ملونة أم رمادية) باستخدام ndims.
* استدعاء الدالة المناسبة (encryptColorImage للصور الملونة و encryptGrayscaleImage للصور الرمادية).

**ملف encryptColorImage.m**

function encryptedImage = encryptColorImage(inputImage)

disp('Processing color image...');

[rows, cols, channels] = size(inputImage);

fibonacciSeq = generateFibonacci(rows \* cols \* channels);

imageArray = reshape(inputImage, 1, []);

fibonacciSeq = cast(fibonacciSeq, class(imageArray));

encryptedArray = bitxor(imageArray, fibonacciSeq);

encryptedImage = reshape(encryptedArray, size(inputImage));

figure;

subplot(1, 2, 1); imshow(inputImage); title('Original Color Image');

subplot(1, 2, 2); imshow(encryptedImage); title('Encrypted Color Image');

end

* عرض رسالة تشير إلى أنه جارٍ معالجة صورة ملونة.
* الحصول على حجم الصورة (صفوف، أعمدة، قنوات).
* إنشاء تسلسل فيبوناتشي بناءً على حجم الصورة الملونة.
* تحويل الصورة الملونة إلى مصفوفة 1D باستخدام reshape.
* التأكد من أن كلتا المصفوفتين لهما نفس النوع.
* استخدام bitxor لتشفير كل بكسل في الصورة.
* إعادة تشكيل المصفوفة المشفرة إلى الحجم الأصلي للصورة.
* عرض الصورة الأصلية والمشفرة جنبًا إلى جنب.

**ملف encryptGrayscaleImage.m**

function encryptedImage = encryptGrayscaleImage(inputImage)

disp('Processing grayscale image...');

[rows, cols] = size(inputImage);

fibonacciSeq = generateFibonacci(rows \* cols);

imageArray = reshape(inputImage, 1, []);

fibonacciSeq = cast(fibonacciSeq, class(imageArray));

encryptedArray = bitxor(imageArray, fibonacciSeq);

encryptedImage = reshape(encryptedArray, size(inputImage));

figure;

subplot(1, 2, 1); imshow(inputImage); title('Original Grayscale Image');

subplot(1, 2, 2); imshow(encryptedImage); title('Encrypted Grayscale Image');

end

* عرض رسالة تشير إلى أنه جارٍ معالجة صورة رمادية.
* الحصول على حجم الصورة (صفوف وأعمدة).
* إنشاء تسلسل فيبوناتشي بناءً على حجم الصورة الرمادية.
* تحويل الصورة الرمادية إلى مصفوفة 1D باستخدام reshape.
* التأكد من أن كلتا المصفوفتين لهما نفس النوع.
* استخدام bitxor لتشفير كل بكسل في الصورة.
* إعادة تشكيل المصفوفة المشفرة إلى الحجم الأصلي للصورة.
* عرض الصورة الأصلية والم

**ملف generateFibonacci.m**

function fibonacciSeq = generateFibonacci(n)

fibonacciSeq = zeros(1, n);

fibonacciSeq(1) = 1;

fibonacciSeq(2) = 1;

for i = 3:n

fibonacciSeq(i) = fibonacciSeq(i-1) + fibonacciSeq(i-2);

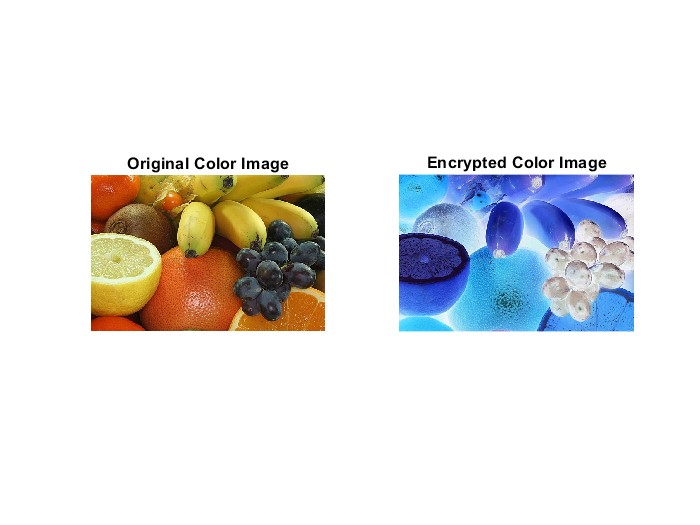
end

end

* تُنشئ fibonacciSeq، وهي مصفوفة بأطول n عناصر وتعبأ بقيم ابتدائية صفر.
* تُضيف قيمتين أوليتين لتسلسل فيبوناتشي (1 و 1).
* يتم استخدام حلقة for لحساب باقي القيم في التسلسل بناءً على القيمتين السابقتين.

باختصار، هذه الدالة تنشئ تسلسل فيبوناتشي حتى العدد n.

**نتائج التشفير**



*نتيجة تشفير الصور الملونة*



*نتيجة تشفير الصور الرمادية*

# المخطط الانسيابي للشفرة البرمجية

