Knutt / Durstenfeld Shuffle Algoritmasının Resim Şifreleme Amacıyla Kullanılması

Erdal GÜVENOĞLU, E.Murat ESİN

ÖZET

Günümüzde veri iletimi için genel kullanıma açık ortamlar yaygın bir şekilde kullanılmaktadır. Bu ortamda bilgi güvenliği büyük önem taşıdığından, resim şifreleme üzerinde yapılan çalışmalar da yoğun bir talep ve ilgi görmektedir.

Bu makalede sunulan çalışmanın ana hedefi, resim şifrelemek amacıyla kullanımı kolay, güçlü ve etkin bir yöntemin geliştirilmesidir. Bu amaçla, bilinen Knutt / Durstenfeld Shuffle Algoritması ile elde edilen bir anahtar dizisi kullanılarak resim piksellerin yerlerini değiştirmeye dayalı yeni bir resim şifreleme yöntemi tanıtılmaktadır. Gerçekleştirilen uygulamada, resim türü ve biçimi önemli olmazsızın resim şifreleme işleminin başarılı bir şekilde gerçekleştirilebildiği görülmüştür.

Anahtar Kelimeler: Şifreleme, Bilgi güvenliği, Resim Güvenliği, Resim Şifreleme

Image Encryption Based On Knutt / Durstenfeld Shuffle Algorithm

ABSTRACT

Nowadays public mediums for data transmission are widely used. Studies on image encryption methods are also interested due to big importance of data security in these mediums.

The main purpose of the study presented in this paper is to develop an easy to use, powerful and effective method for image encryption. In this aim, a new image encryption method based on changing pixels locations in the image by using a key string handled with well known Knutt / Durstenfeld Shuffle Algorithm is presented. It is seen that, image encryption is realized successfully with developed application as independent from image format.

Keywords: Encryption, Data Security, Image Security, Image Encryption

1. GİRİŞ

Günümüz teknolojisinin gelişmesi dijital ortamda verilerin gizliliğinin önemini arttırmıştır. Elektrooptik teknolojilerinin gelişmesiyle çözünürlük yükseldiğinden büyük boyutlu ve çok sayıda resmin depolanması ve yüksek hızlarda iletilmesi gerekmekte ve internet trafiğinde büyük bir artış gözlenmektedir.

İletim ortamının doğası gereği bu iletişimin büyük kısmı izlenmeye ve müdahaleye açıktır. Oysa, haberleşmenin niteliği ve kişi hakları açısından iletilen verinin ilişkisiz kimselerden gizli kalması gerekmektedir. Bilgi gizliliğinin sağlanmasında en güvenli yol şifrelemedir.

Verilerin şifrelenmesi için DES (Data Encryption Standard) ve RSA (Rivest-Shamir-Adleman) gibi geleneksel metin şifreleme yöntemlerinin kullanımı yaygındır. Bu yöntemler aslında bir dizi sayısal veriden ibaret olan resim dosyalarının şifrelenmesinde de kullanılabilmektedir.

Makale 23.02.2009 tarihinde gelmiş,12.10.2009 tarihinde yayınlanmak üzere kabul edilmiştir.
E. GÜVENOĞLU, Maltepe Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü
e-posta : erdalg@maltepe.edu.tr
E.M. ESİN,, Maltepe Üniversitesi Mühendislik Fakültesi Elektronik Mühendiğisliği Bölümü
e-posta : emesin@maltepe.edu.tr
Digital Object Identifier 10.2339/2009.12.3, 151-155

Ancak bu yöntemlerle birlikte iki önemli sakınca ortaya çıkmaktadır. Bunlardan ilki; resim verileri metin verilerine göre çok büyüktür ve geleneksel yöntemlerle şifrelenmesi çok zaman almaktadır. İkincisi, şifrelenmiş metin tam olarak orijinal haline çevrilmedikçe içeriği anlaşılamazken, resim verilerinin kısmen çözülmesi bile içeriğinin anlaşılması için yeterli olmaktadır. İnsan algısı çözümden kaynaklanan hatayı göz ardı edebilmektedir. Görüntülerin kısmen veya tamamen çözülmesini engellemek için çeşitli resim şifreleme teknikleri geliştirilmiştir.

Resim şifreleme algoritmalarının üç temel fikri vardır. Bunlar; değer dönüşümü (1-2-3), yerel permütasyon (4-5) ve değer dönüşümü ile yerel permütasyon yöntemlerinin kombinasyonlarıdır (4).

Değer dönüşümü, orijinal pikselin veri değerinin algoritmadaki işleme tabi tutulduktan sonra aldığı yeni değer olarak tanımlanmaktadır.

Yerel permütasyon algoritmaları, orijinal piksel verisinin bulunduğu pozisyonunun yer değiştirmesi olarak ifade edilmektedir.

Bunların kombinasyonları ise; her iki yöntemin birlikte kullanılması ile gerçekleştirilmektedir.

Sonuçta resim şifreleme yaklaşımlarında esas olarak, resmi oluşturan pikselleri temsil eden sayısal değerlerin değiştirilmesini veya bu piksellerin resimdeki yerlerinin değiştirilmesini sağlamaktır. Doğal olarak her

iki yaklaşımda da geriye dönüşüm yapılarak orijinal resmin elde edilebilmesi beklenir.

Jiun-Guo ve Jui-Cheng tarafından ortaya konulan karmaşık resim şifreleme algoritması piksel yeri değiştirmeye dayalı bir resim şifreleme algoritmasıdır (5-6). Algoritmanın uygulamasında herhangi bir veri kaybı olmamaktadır. Bunun nedeni, dönüştürme sırasında resmin piksel değerinin üzerinde yapılacak aritmetik veya mantık işlemlerinin geri dönüşümü tekil olmayan değerler üretmesi nedeniyle bozulması olasılığını içermeyecek şekilde, sadece bulunduğu yerin değiştirilmesidir.

Jiun-Guo ve Jui-Cheng Yen tarafından ortaya konulan bir diğer yöntemde; resmi oluşturan dizi ardışık olarak alt dizilere bölünmekte ve bu alt dizilerin aynı indisli elemanları arasında karşılıklı yer değiştirmelerle şifreleme yapılmaktadır (4). Algoritma 7 adımdan oluşmakta, fakat kullanılan yöntem bilindiğinde kolaylıkla çözülebilmektedir.

YEN J.C. ve Guo J.I. tarafından ortaya atılan bit ötelemeli resim şifreleme algoritması, karmaşık resim şifreleme yaklaşımını kullanan yeni bir yöntemdir (7). Karmaşık bir sistemden bit kaydırmalı bir fonksiyon ve ikili bir dizi tanımlanarak resmin her pikseli gri resim pikseline dönüştürülmektedir. Bu algoritma, özel bir mantığa dayalı olarak her piksel üzerinde bit ötelemesi ile değer dönüşümü yapmaktadır.

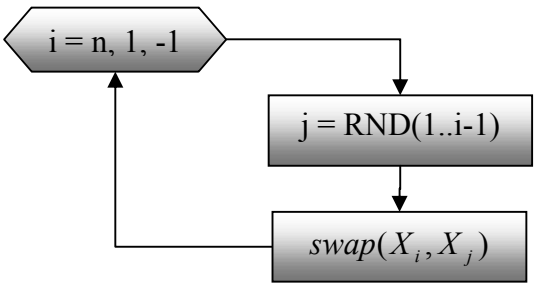
2. KNUTT / DURSTENFELD SHUFFLE ALGORİTMASI

Kart karıştırma, sayılar bilimi, şifreleme ve simülasyon gibi alanlarda günlük yaşantımızda olduğu gibi bilgisayar hesaplamalarında da rastgele sayıların permütasyonu, ya da literatürde bilinen adıyla "shuffle" sıklıkla kullanılmaktadır.

Shuffle algoritmalarının bu versiyonu, 1963 yılında L.E.Moses ve R.V.Oakford (8) ve 1964 te Richard Durstenfeld (9) tarafından yayınlanmıştır. Fakat yaygın olarak Knuth Shuffle olarak bilinmiş ve 1969 daki The art of Computer Programming kitabının ikinci bölümünde yayınlanmıştır (10).

Knutt / Durstenfeld Shuffle Algoritması (K/DSA), bir dizi elemanın kendi içerisinde yer değiştirmesi suretiyle karıştırılması için tasarlanmıştır. Eleman sayısı n olan bir X dizisi olsun. Bu dizinin elemanları karıştırarak elde edilen diziye ise X' adını verelim. X' dizisi X in elemanlarının yerlerinin rastgele değiştirilmesiyle oluşturulmuş olacaktır.

Buna göre K/DSA nın çalışma biçimi şöyledir: X dizinin elemanları 1 den n e kadar sıra numaraları alabileceğine göre $1 \le k \le (n-1)$ olacak şekilde rastgele bir k sayısı seçilir. Ardından X_k ile X_n elemanlarının yerleri değiştirilir. Sonraki adımda bu defa $1 \le k \le (n-2)$ olacak şekilde yeni bir rastgele k belirlenir ve X_k ile $X_{(n-1)}$ elemanları yer değiştirir. İşlem X_1 ile X_2 nin yer değiştirilmesine kadar devam eder. Doğal olarak sonraki adımlarda sağ tarafa aktarılmış elemanlar bir daha yer değiştiremezken sol tarafa aktarılmış elemanların yerleri tekrar tekrar değiştirilmiş olacaktır. K/DSA 'nın akış şeması Şekil 1 de gösterilmektedir.



Şekil 1. K/DSA 'nın akış şeması

Örneğin; 10 elemanlı bir X dizisi düşünelim. $n = 10$ olduğuna göre X dizisi

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$$

olsun. Bunun K/DSA ile karıştırılmasından elde edilecek olan X' dizilerinden mümkün bir tanesi aşağıdaki şekilde olacaktır.

$$X' = \{x_2, x_{10}, x_5, x_1, x_4, x_7, x_6, x_9, x_3, x_8\}$$

Bu yöntemde karıştırılmış dizinin karıştırılma sırası hakkında bir kayıt tutulmadığından geriye dönülmesi mümkün değildir. Bu algoritma ortaya atıldığında, karıştırılan dizilerin geriye dönüşmesini gerektirecek uygulamalar amaçlanmadığından böyle bir gereksinim tanımlanmamıştır.

Yapılan işin doğası gereği algoritma her çalıştırıldığında farklı bir X' dizisi elde edilecektir (11). Dizideki eleman sayısı n in büyüklüğüne bağlı olarak elde edilecek farklı X' sayısının $n!$ kadar olacağı bilinmektedir. Bu sayı aynı zamanda orijinal dizinin bulunması için maksimum deneme sayısını vermektedir.

3. KNUTT / DURSTENFELD SHUFFLE ALGORİTMASI İLE RESİM ŞİFRELEME

K/DSA ile karıştırılan dizinin geri dönüşünün mümkün olmadığı ifade edilmişti. Ancak özel bir X dizisinin karıştırılması halinde, ortaya çıkacak X' dizinin anahtar olarak kullanıldığı bu çalışma ile metin ve resim şifrelemesinin yapılabileceği aşağıda gösterilmektedir.

Sayısal resim dosyaları, temel olarak ekrandaki piksellerin renklerini tanımlayan bir diziden ibarettir. Önceleri, ekran çözünürlükleri düşük olduğundan her bir pikselin rengini tanımlamak için bellekte tek bir adres yeterliydi. Çözünürlüğün artmasına paralel olarak renk koduna karşı gelen sayılar büyüdüğünden bir piksele karşılık birden fazla adrese yayılmış verilerin yazılması gerekti. Her bir piksel için bir ya da daha fazla bellek adresi kullanıldığına bakılmaksızın, salt bellekte yerleşim düzeni açısından resmin bir dizi olarak ele alınması mümkündür.

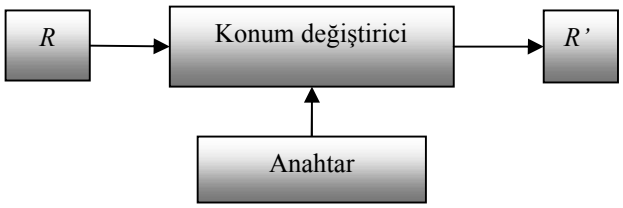
Bu durumda; üzerinde algoritma geliştirilen resim dizisinin iki farklı düşünce ile ele alınması mümkündür. Resim belleğinin kendisi bir dizi olarak ele alınabileceği gibi, piksel verileri birden fazla bellek adresine yerleşmiş olsa bile tek eleman olarak düşünülebilir.

İlk yaklaşımda resmin piksel renklerini oluşturan bileşenleri de karıştırılırken, ikinci yaklaşımda sadece pikselin resim çerçevesindeki yeri değiştirilmektedir.

Bu çalışmada kullanılan K/DSA temelde eleman yerlerini değiştirmeye dayalı olarak tasarlandığından her piksel bir eleman olarak tanımlanmaktadır. Bu nedenle algoritmanın çalışma hızı bellek erişimi dışındaki nedenlerle çözünürlüğe bağımlı değildir.

Resim, her bir pikseli oluşturan renk kodunu tek bir eleman olarak kabul eden diziye *R* ve *R* nin şifrelenmesiyle ortaya çıkacak diziye *R'* adını verelim. *R* den *R'* ye ve gerektiğinde *R'* den *R* ye dönüşümü sağlayacak yer değişimlerini belirleyecek bir anahtar diziye gerek olacaktır. Yani tek bir anahtar her iki amaçla da kullanılabilecektir.

Bu yaklaşımda anahtar dizinin resim dizisiyle aynı boyutta olması gerekmektedir. Şekil 2 şifreleme işini şematik olarak göstermektedir.



Şekil 2. Şifreleme mekanizması

R dizisinin şifrelenmesi için kullanılacak anahtar dizisi, yer değiştirmeleri yönetecek, belli bir algoritmayla rastgele üretilmiş *n* boyutlu bir dizi olmalıdır.

3.1. Anahtar Dizisinin K/DSA İle Üretilmesi

Bu özelliklere sahip bir anahtar dizisi K/DSA ile üretilebilir. Bu amaçla *n* boyutlu ve elemanları 1 den *n* e kadar sayılar olan bir *X* dizisi oluşturulur. Ardından *X* dizisine K/DSA uygulanarak tamamen rastgele bir *X'* dizisi elde edilir. *X'* dizisinde 1 den *n* e kadar sayılar rastgele sıralanmıştır.

Örneğin; *X* dizisi 10 elemanlı olsun.

$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$

Notasyonuyla verilecek dizinin elemanları sıralı tam sayılar olacağından;

$X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

olacaktır. Buna karşılık *X'* dizisinin mümkün kombinasyonlardan birisi;

$X' = \{3, 7, 1, 9, 8, 5, 6, 10, 4, 2\}$

olabilir. Bu durumda *X'* dizisinin genel hali;

$X' = \{x'_1, x'_2, x'_3, x'_4, x'_5, x'_6, x'_7, x'_8, x'_9, x'_{10}\}$

demektir.

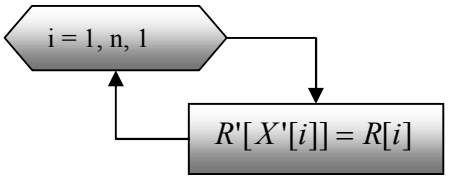
3.2. Resim Şifreleme ve Geriye Dönüşüm

Aşağıda görüleceği üzere şifreleme piksellerin resim çerçevesindeki yerlerinin geriye dönüştürülebilir şekilde değiştirilmesinden ibarettir. Yer değiştirmede *X'* dizisinin indisleri *R* dizisinde yeri değiştirilecek elemanın indisini, *X'* dizisinde indisi bu olan elemanın kendisi ise *R* den okunacak elemanın *R'* de yerleştirileceği yerin indisini gösterecektir.

Şifrelenmiş *R'* dizinin herhangi bir elemanının genel ifadesi;

$$R'_{X'_i} = R_i$$

olur. *X'* anahtar dizisi yardımıyla *R* resminin şifrelemesi için akış şeması Şekil 3 de gösterildiği gibidir.



Şekil 3. Yer değiştirme ile resim şifreleme akış şeması

İşleyişi bir örnek üzerinde gösterecek olursak; yine 10 elemanlı bir *R* dizisi ele alalım. Anlaşılırlığı kolaylaştırmak açısından *R* dizisi A dan J ye kadar sıralı alfabetik karakterlerden oluşmuş olsun. Bu elemanların yerleri *X'* nün indisleri ve karşı gelen elemanları yardımıyla yer değiştireceklerdir. Buna göre *R* dizisinde r_1 elemanının değeri olan A nın *R'* deki yeri *X'* dizisinde x'_1 elemanının değeri olan 3. sıradır. Benzer şekilde; r_2 nin değeri olan B nin yeri *R'* de 7, r_3 ün değeri olan C nin yeni yeri ise 1 dir.

Tablo 1. Anahtara göre yer değiştirmelerin gerçekleştirilmesi.

R nin indisleri	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
R nin elemanları	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**

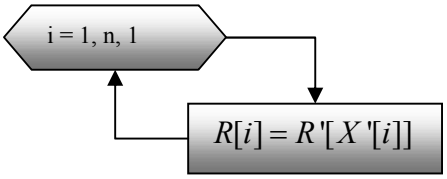
X' nün indisleri	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
X' nün elemanları	3	7	1	9	8	5	6	10	4	2

R' nün indisleri	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
R' nün elemanları	**C**	**J**	**A**	**I**	**F**	**G**	**B**	**E**	**D**	**H**

Açıkça görüleceği üzere şifrelenmiş resmin geriye dönüştürülmesi için benzer bir işlemin bu defa *R'* üzerine uygulanması yeterli olacaktır. Geriye dönüşüm için genel ifade;

$$R_i = R'_{X'_i}$$

olur. *X'* yardımıyla resmin geriye dönüştürülmesi için akış şeması Şekil 4. de verilmiştir.



Şekil 4. Yer değiştirme ile şifre çözme akış şeması

Yukarıda verilen örnek dizinin geriye dönüşümü Tablo 2 den izlenebilir.

Tablo 2. Anahtara göre yer değiştirmeyle geri dönüşüm

R' nün indisleri	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
R' nün elemanları	**C**	**J**	**A**	**I**	**F**	**G**	**B**	**E**	**D**	**H**

X' nün indisleri	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
X' nün elemanları	3	7	1	9	8	5	6	10	4	2

R nin indisleri	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
R nin elemanları	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**

4. UYGULAMA

Bu bölümde makalenin önceki kısımlarında detayları verilen yöntemin uygulanması ile elde edilen sonuçlar ele alınmaktadır.

Günümüzde bilinen birçok yöntemin uygulanması, kolay olması nedeni ile gri resimler üzerinde gerçekleştirilmektedir. Bu yöntemde resmin türünün ve biçiminin önemli olmadığı görülmüştür.

Zira, *X'* anahtar dizisinin uzunluğu, *R* resmi için *m* resmin satır sayısı, *n* resmin sütun sayısı olmak üzere *m x n* olacağından, bu dizi için üretilebilecek anahtar sayısının;

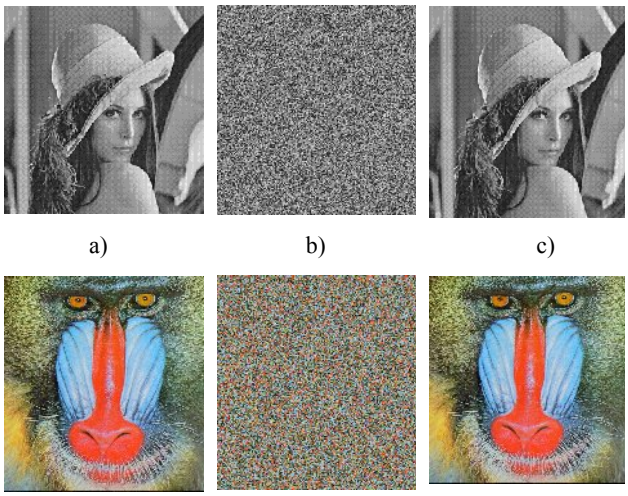
$$P = m.n!$$

olacağı açıktır.

Örneğin 256x256 piksel boyutunda bir *R* resminin şifrelendiğini varsayalım. *R'* resmi ve *X'* anahtar dizisi de aynı boyutlarda olacağından, *X'* anahtar dizisinin maksimum uzunluğu 256x256 = 65536 olacaktır. *X'* anahtar dizisinin uzunluğunun 65536 olduğunu düşünürsek, 65536! farklı anahtar olacaktır.

Geliştirilen algoritmayı test etmek amacıyla oluşturulan yazılım, Delphi 7.0 programlama dilinde kodlanıp ve Windows XP işletim sisteminde Intel 2.00 GHz işlemci ve 2GB ana belleğe sahip bilgisayar üzerinde çalıştırılmıştır. Uygulamanın hızına dair bir fikir olması bakımından elde edilen değerler Tablo 3 de incelenebilir.

Resim şifreleme alanında sık kullanılan gri tonlamalı lena.bmp resmi ile 24 bit renkli baboon.bmp resimlerinin şifrelenmiş ve şifresi çözülmüş görüntüleri ise Şekil 5 de gösterilmektedir.



a) b) c)

d) e) f)

Şekil 5. (a)Orijinal lena.bmp, (b) Şifrelenmiş lena.bmp, (c) Şifresi çözülmüş lena.bmp, (d) Orijinal 24 bit renkli baboon.bmp, (e) Şifrelenmiş baboon.bmp ve (f) Şifresi çözülmüş baboon.bmp görüntüleri

Şifreleme ve şifre çözme işlemlerinin başarılı olabilmesi için herhangi bir veri kaybının olmaması gerekmektedir. Bu nedenle şifreleme ve deşifreleme işlemleri yapıldıktan sonra şifreli ve orijinal resim arasındaki ortalama karesel hatanın (MSE - mean sequared error) bulunması geliştirilen yöntemin başarısı hakkında fikir elde etmemizi sağlayacaktır.

MSE, iki resim arasındaki farkı belirlemek için kullanılmaktadır.

$$\mathrm{MSE} = \frac{1}{\mathrm{MN}} \sum \left[\mathrm{I}_1(m,n) - \mathrm{I}_2(m,n) \right]^2$$

Yukarıda I_1 ve I_2 sırasıyla orijinal görüntü ve şifrelenmiş görüntüleri, M ve N orijinal görüntünün gerçek boyutlarını göstermektedir. MSE denklemi kullanılarak yapılan testte herhangi bir piksel kaybının olmadığı görülmüştür.

Tablo 3. K/DSA' nın uygulanmasında elde edilen sonuçlar

Resim	Resmin Boyutu	Şifrelenen Veri Miktarı[Byte]	Şifresi Çözülen Veri Miktarı [Byte]	Şifreleme Süresi [msn]	Şifre Çözme Süresi [msn]
Lena.bmp(gri)	256x256	65536	65536	828	476
Baboon.bmp(renkli)	256x256	196608	196608	871	484

5. SONUÇ

Bu çalışmada, K/DSA nın resim şifreleme amacıyla kullanılabileceği ve değişik uygulama örneklerinde başarıma etkisi gösterilmiştir. İ.Öztürk ve İ.Soğukpınar tarafından mevcut yöntemlerin karşılaştırmaları yapılmıştır (12). K/DSA ile önerilen yöntem, mevcut şifreleme algoritmaları ile karşılaştırıldığında resmin özelliğinin hiçbir öneminin olmadığı çok daha etkin bir yöntem elde edilmiştir. Yapılan uygulama örneklerinde elde edilen diğer önemli sonuçlarda şunlardır:

- Resim şifreleme amacıyla kullanılan mevcut yöntemlerden kullanımı daha kolay ve etkin bir yöntem elde edilmiştir.
- Bu yöntemde, resmin piksel pozisyonlarında değişiklik yapılırken piksellerin çakışması söz konusu değildir. Resmin piksellerinin yerleri değiştirilirken bir anahtar dizisi kullanılmaktadır. Aynı şekilde resmi çözmek içinde aynı anahtar dizisi kullanıldığından herhangi bir veri kaybı olmamaktadır.
- Resimler şifrelendiğinde, şifrelenmiş resimlerin insan gözüyle algılanabilmesi mümkün değildir. Orijinal resmin elde edilmesi ancak doğru anahtarın bilinmesi ile mümkündür.
- Şifrelenmiş resimde pikseller arasında herhangi benzer bir ilişki bulunmaması bilinen saldırı yöntemleri ile orijinal resmin elde edilmesini zorlaştırmaktadır.
- Mevcut resim şifreleme yöntemleri genellikle gri resimler üzerinde uygulanabiliyorken, bu yöntemde resmin özelliği ne olursa olsun resim şifreleme ve çözme işlemleri kolaylıkla uygulanabilmektedir.

Bu sonuçlar nedeniyle gösterilen yöntemin kolay uygulanabilir bir yöntem olarak kullanılması mümkün görülmektedir.

6. KAYNAKLAR

1. Sinha,A., Singh, K., "A technique for image encryption using digital signature", Optics Communications, 1-6, 2003.
2. Maniccam, S.S., Bourbakis, N.G., "Lossless image compression and encryption using SCAN", Pattern Recognition , 34, 1229-1245, 2001.
3. Chang, C. C., Hwang, M. S., Chen T. S., "A new encription algorithm for image cryptosystems", The Journal of Systems and Software, 58, 83-91, 2001.
4. Guo, J. I., Yen, J. C., "A new mirror-like image encryption algorithm and its VLSI architecture", In Proc. 10th (Taiwan) VLSI Design/CAD Symposium, 319-322, 1999.
5. Yen J.C., Guo, J.-I., "A new chaotic image encryption algorithm",In: Proceedings of (Taiwan) National Symposium on Telecommunications, pp. 358-362, 1998.
6. YEN J.C, Guo J.I., "A new encryption algorithm for image cryptosystems", Journal of Systems and Software, Volume 58, Number 2, pp. 83- 91(9), 2001.
7. YEN J.C, Guo J.I., "A new ımage encryption algorithm and its VLSI architecture", IEEE Workshop on Signal Processing Systems, 430–437, 1999.
8. Moses, L.E., Oakford R.V., "Tables of Random Permutations", Stanford University Pres, ISBN-13: 978-0804701488, 1963.
9. Durstenfeld R., "Communications of the ACM", Association for Computing Machinery , ISSN:0001-0782, vol.7, issue 7, Page 420,1964.
10. Knuth D. "The Art of Computer Programming", 2th edition, Addison-Wesley, pp.139-140, 1969.
11. ESİN, E.M., GÜVENOĞLU, E., "Resim İçine Yazı Gizlenmesi Amacıyla Kullanılan EDB Ekleme Yönteminin Shuffle Algoritmasıyla İyileştirilmesi", Türkiye Bilişim Vakfı Bilgisayar Bilimleri Dergisi, sayı 2, sayfa 73-79, 2006.
12. Öztürk, İ., Soğukpınar, İ., "Analysis and Comparison of Image Encryption Algorithms", International Journal of Information Technology, Vol. 1. No. 2, pages 108-114, 2004