BEYKENT ÜNİVERSİTESİ FEN VE MÜHENDİSLİK BİLİMLERİ DERGİSİ
CİLT SAYI:13/1
www.dergipark.gov.tr

ASAL KATSAYILARLA DÜZENLENMİŞ DOĞRUSAL ÜRETEÇ

Hakan GENÇOĞLU*, Tarık YERLİKAYA**

ÖZ

Karıştırma algoritmaları, bir veri bütünün parça parça düşünüp bu parçaların yerlerini rastgele seçilen parçalarla değiştirmesi şeklinde işlem yapar. Yani rastgele üretilen bir değere ait konumda bulunan bir veriyi, sıradaki konumdaki veri ile yer değiştirerek karıştırma işlemi sağlanmış olur. Burada karıştırma işleminin gerçekleşmesini sağlayan rastgele üretilen değerlerdir. Karıştırma algoritmalarının temel amacı veriyi karıştırmak olduğundan karıştırılmış veri bloklarının eski haline getirilmesi ile ilgilenmezler. Dolayısıyla ihtiyaç halinde eski veriye ulaşmak gerekirse karıştırma algoritmaları tek başlarına yeterli olmaz. Bu çalışmada geri dönüşsüz karıştırma algoritmaları ile yapılan veri bloklarının geri getirilmesi incelenmiş ve farklı bir algoritma önerilmiştir.

Anahtar Kelimeler: *Karıştırma, Karıştırma Algoritması, Doğrusal Sayı Üreticisi, Fisher Yates, Knutt Durstenfeld*

*Makale Gönderim Tarihi: 16.01.2020 ; Makale Kabul Tarihi : 30.05.2020 Makale Türü: Araştırma
DOI: 10.20854/bujse.666813
*Sorumlu yazar: İstanbul Zaim Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü (hakan.gencoglu@izu.edu.tr) (ORCID ID: 0000-0003-2968-1615)
**Trakya Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü (tarikyer@trakya.edu.tr) (ORCID ID: 0000-0002-9888-0151)

BEYKENT ÜNİVERSİTESİ FEN VE MÜHENDİSLİK BİLİMLERİ DERGİSİ
CİLT SAYI:13/1
www.dergipark.gov.tr

PRIME COEFFICIENT LINEAR NUMBER GENERATOR FOR SHUFFLING

Hakan GENÇOĞLU*, Tarık YERLİKAYA**

ABSTRACT

Principle of shuffle algorithms is to change the place of the data blocks by producing random numbers. A produced number means the new place of the data value. That is, shuffling process is based on to produce random numbers. Because shuffle algorithms aim is to shuffle data blocks, they do not have a function to reverse the data. Therefore if it is needed original data, shuffle algorithms are insufficient. In this study, we examined reverse methods of the shuffled data blocks that shuffled by the shuffle algorithms and suggested a new algorithm to shuffle data blocks.

Keywords: *Shuffle, Shuffle Algorithm, Linear Number Generator, Fisher Yates, Knutt Durstenfeld*

*Makale Gönderim Tarihi: 16.01.2020 ; Makale Kabul Tarihi : 30.05.2020 Makale Türü: Araştırma
DOI: 10.20854/bujse.666813
*Sorumlu yazar: İstanbul Zaim Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü (hakan.gencoglu@izu.edu.tr) (ORCID ID: 0000-0003-2968-1615)
**Trakya Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü (tarikyer@trakya.edu.tr) (ORCID ID: 0000-0002-9888-0151)

1.GİRİŞ

Veriyi veri blokları halinde düşünüp bu blokların yerlerinin değiştirilmesi, karıştırma işlemidir. Bu işlem veri güvenliğinin temelini oluşturmaktadır. Aslında şifreleme algoritmaları da temel olarak veriyi başka bir veri ile değiştirerek güvenliği sağlamaktadır. Günümüzde kullanılan simetrik veya asimetrik algoritmalar çeşitli hesaplama veya yer değiştirme işlemleri ile bir karmaşıklık ortaya çıkararak veriyi anlaşılmaz hale getirmektedir.
Karıştırma işlemini iki şekilde düşünebiliriz.

- Tek yönlü karıştırma
- Geri dönüşlü karıştırma

Tek yönlü karıştırma işlemi için geliştirilen algoritmalar, karıştırılmış olan verilerden asıl veriyi elde etmeyi düşünmez. Sadece veriyi karıştırır. Fisher/Yates algoritması, Knutt/Durnstenfeld algoritmaları bu tür algoritmalardır.

Geri dönüşlü karıştırma işlemleri ise veri karıştırıldıktan sonra karışık halinden tekrar asıl veriyi elde edecek şekilde tasarlanırlar. Veri işlemlerden geçirilerek karıştırılır, karışık halde alıcıya iletilir ve alıcı tarafında işlemlerden geçirilerek asıl veri elde edilir. Günümüzün güçlü simetrik şifreleme algoritması AES örnek olarak verilebilir. AES şifreleme algoritması karıştırma işlemini yaparken bir anahtar kullanır ve yine aynı anahtarı kullanarak ası veriyi elde eder.

Yapı itibari ile tek yönlü algoritmalar, geri dönüşlü algoritmalara göre, çok daha basittirler.
Eğer tek yönlü algoritmalar geri dönüşlü algoritmalar gibi kullanılmak istenirse, uygulama sırasında algoritma içeriğinde bulunmayan ek modüller kullanılmalıdır. Örneğin veri blokları karıştırılıp karışık veri elde edilirse bu veri bloklarının karıştırılmadan önceki halleri de tutularak geri dönüş sağlanabilir. Yani asıl veriye ait blokları bir veri dizisi olarak düşünecek olursak, bu dizinin karıştırıldıktan sonra yeni dizinin elemanlarının asıl yerleri başka bir dizide tutulmalıdır ki bu ikinci dizi kullanılarak karıştırılmış olan veri asıl haline geri getirilebilsin. Bu da ek işlem yükü, veri boyutunun ikiye katlanması problemlerini doğurur.

Bu çalışmamızda tek yönlü karıştırma işlemleri gibi basit, geri dönüşlü bir yapı önerilmiştir. Önce tek yönlü karıştırma algoritmaları açıklanmış, ardından bu algoritmalar kullanılarak gerçekleştirilmiş çalışmalardan bahsedilmiştir. Daha sonra önerdiğimiz yapı açıklanarak ispatı gerçekleştirilmiş ve değerlendirilmeler yapılmıştır.

2. Karıştırma Algoritmaları ve Sayı üreteçleri

Karıştırma algoritmaları isminden de anlaşılacağı üzere var olan bir sayı dizisine ait değerlerin, sayı dizisinde bulunduğu yeri değiştirmek amacıyla kullanılan algoritmalardır. Dolayısıyla, karıştırma algoritmaları, diziye ait olan elemanların değerleri ile değil indisleri ile çalışır. İskambil oyunlarında oyuna başlamadan önce kağıtların karıştırılması böyle bir karıştırma türüdür. Bu algoritmalar bu oyunların dijital karşılıklarında başarılı bir şekilde kullanılmaktadır.
Rastgele sayı üreteçleri belirli bir denklem çerçevesinde, başlangıç değeri kullanılarak yeni sayı üretme ilkesine dayanır. Haritalama işlemleri için kullanılırlar.

3. Önceki Çalışmalar

Karıştırma algoritmaları sadece karıştırma işlemi için değil veri şifreleme amaçlı olarak da kullanılmıştır[2,3]

"Knutt / Durstenfeld Shuffle Algoritmasının Resim Şifreleme Amacıyla Kullanılması" isimli çalışmada resim pikselleri diziye dönüştürülmüş, dizi üzerinde karıştırma işlemi uygulanmıştır. Bununla birlikte başka bir dizide resmin orijinal sıradaki piksel değerleri ile yeni yeri bir eşleşme dizisinde tutulmuştur. Şifrelenmiş resmin geri dönüşümü işlemi de eşleşme dizisi kullanılarak piksellerin eski yerlerine gelmesi sağlanarak başarılmıştır. [2]
Benzer bir çalışma olan "File Encryption using Fisher-Yates Shuffle" isimli çalışmada dosya şifreleme işlemi gerçekleştirilmiştir. [3]
Başka bir çalışmada karıştırma algoritmaları mantığı resim şifreleme işlemi için kullanılmış fakat şifreli resmin geri getirilmesi açıklanmamıştır.[4]

Bu çalışmalarda karıştırma algoritmaları geri dönüşüm gerektiren uygulamalarda kullanılmıştır. Teorik olarak bunun önünde bir engel olmamakla birlikte uygulamada bir takım problemlerle karşılaşılır. Uygulama sırasında bu problemleri ortadan kaldırmak için çözümler üretilmelidir.

Karıştırma algoritmaları ile ilgili başka bir çalışmada, shuffle algoritması, tanımlayıcı sürekli zamanlı ve ayrık zamanlı doğrusal sistemlerin pozitifliğini kontrol etmek için kullanımıştır.[5]

Rastgele sayı üreteçlerinin en önemli örneklerinden biri Lehmer tarafından önerilen "Lehmer's congruential method" tur. Belirli bir mod değerine göre x_i başlangıç değerini a katsayısı ile çarparak genişletme şeklinde tanımlanmıştır.

$$x_{i+1} = ax_i mod(m)$$

Doğrusal bir yapıda tanımlanmış bu denklem ile istenildiği kadar değer üretilebilmesi m değerine bağlıdır. [6]

3.1.FISHER/YATES Algoritması

Ronald Fisher ve Frank Yates tarafından 1938 yılında yayınlanmıştır. Sonlu kümelerde elemanların yerlerini değiştirerek karıştırmak için kullanılır. N elemanlı sonlu bir dizi için algoritma şu şekilde çalışır:
1. Dizinin elemanlarını 1 den N e kadar sırala
2. $1 \leq k \leq N$ olacak şekilde rastgele k sayısı seçilir.
3. k numaralı eleman dizinin en sonundan itibaren öne doğru dizilir.
4. 2 ve 3 numaralı adımlar dizinin yeri değiştirilmeyen elemanı kalmayana dek devam eder.

Fisher/Yates Karıştırma Algoritmasının Uygulamasının bir örneği şekil 1 de gösterilmiştir.[1,2]

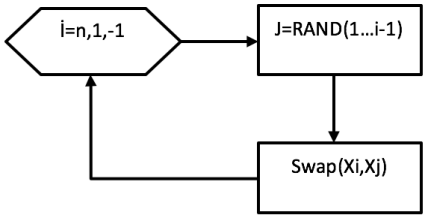
Orijinal Dizi	K. numaralı Eleman	Yeni Dizi
1-2-3-4-5-6-7-8-		
1-2-4-5-6-7-8-9	3	3
1-2-4-5-7-8-9	5	6-3
1-2-4-7-8-9	4	5-6-3
1-4-7-8-9	2	2-5-6-3
1-4-7-8	5	9-2-5-6-3
4-7-8	1	1-9-2-5-6-3
4-7	3	8-1-9-2-5-6-3
7	1	4-8-1-9-2-5-6-
		7-4-8-1-9-2-5-

Şekil 1. Fisher/Yates Karıştırma Algoritmasının Uygulaması

3.2 KNUTT/DURNSTENFELD Algoritması

1964 yılında Richard Durnstenfeld tarafından yayınlanmıştır. KNUTT/DURNSTENFELD Shuffle Algorithm (K/DSA) bir dizinin elemanlarının kendi içinde yer değiştirilmesi ile karıştırılması işlemini gerçekleştirir. K/DSA şöyledir:
Bir A dizisi n elemanlı olsun.
1. $1 \leq k \leq n$ olacak şekilde rastgele bir k numarası seçilir
2. Dizinin k numaralı elemanı ile n numaralı elemanı yer değiştirilir. ($A_k \leftrightarrow A_n$)
3. 1. Adım $1 \leq k \leq n-1$ olacak şekilde tekrarlanır, 2 numaralı adım ($A_k \leftrightarrow A_{n-1}$) olacak şekilde tekrarlanır.
4. 1 ve 2 numaralı adımlar 3. Numaralı adımda olduğu gibi sürekli en son işleme sokulan elemandan bir önceki eleman işleme sokularak tekrarlanır.
5. 1 ve 2 numaralı işlemler 3 ve 4 de olduğu gibi $A_1 \leftrightarrow A_2$ olana kadar devam eder.

Şekil 2 de Algoritmanın akış şeması görülmektedir.



Şekil 2. KNUTT/DURNSTENFELD Algoritmasının akış şeması

K/DSA kullanılarak karıştırılmış bir dizinin karıştırılma işlemine ait bir kayıt tutulmadığı için geri dönüşü olmayan tek yönlü bir algoritmadır. Seçilen k değerleri rastgele olduğundan ve algoritmanın doğası gereği, algoritma her uygulandığında farklı bir sonuç elde edilecektir.

4. Karıştırma Amaçlı Doğrusal Üreteç Tasarımı

Geri dönüşüm gerektiren işlemlerde karıştırma algoritmaları yerine doğrusal üreteçler (Lineer kongrüans üreteçler - LKÜ) tercih edilebilir. Bu üreteçler aşağıdaki gibi tanımlanır:
(m > 0) bir doğal sayı olmak
üzere, $X_i \in \{1,2,..., m-1\}$ baslangıç değerini seçip,
$X_{i+1} = aX_i + c \ mod \ m$ algoritmasına göre $X_1, X_2, \ldots$ sayılarını ve bu sayılar yardımıyla,
$u_1 = X_1/m$, $u_2 = X_2/m$, $u_3 = X_3/m \in (0,1)$ sayılarını üretmektedir.
Böyle bir fonksiyon eğer a ve c katsayıları biliniyorsa geri dönüşümü olan bir karıştırma işlemi sağlayabilir. Eğer geri dönüşüm olması isteniyorsa a ve c katsayıları, fonksiyon mod m e göre bijektif olacak şekilde seçilmelidir. Bu şekilde tersi alınabilir bir fonksiyon olur ki geri dönüş sağlanabilir.

Algoritmanın amacı asıl verinin indislerini değiştirmek olduğundan bijektiflik sağlanabilmeli ve geri dönüş sağlanabilmelidir. Önerilen üreteç de katsayılar mod değerinden küçük olacak şekilde göreceli olarak asal sayılar olacak şekilde seçilmiştir.

5. Asal Katsayılarla Düzenlenmiş Doğrusal Üreteç AKDÜ

5.1 Karıştırma işlemi

$A[n]=\{a_1, a_2, a_3, \ldots, a_n\}$ ve $1 \leq x \leq n$ olacak şekilde A dizisinin her x numaralı elemanını başka bir numara ile değiştiren algoritma şöyledir:

1. $2 \leq p \leq n$ ve $2 \leq q \leq n$ olacak şekilde p,q asal sayıları seçilir.
2. Sh:[1,n] → [1,n]
Sh(x)=px+q karıştırma fonksiyonu oluşturulur.
3. $(T[x] = A[Sh(x)])_{x=1}^{n}$ A dizisinin, karıştırma fonksiyonu ile belirlenmiş sıradaki elemanı, T dizisine sırayla yerleştirilir.
4. $(A[x] = T[x])_{x=1}^{n}$ Geçici T dizisinin elemanları A dizisine aktarılarak A dizisinin elemanları yer değiştirilmiş olur.

Buradaki asıl soru $Sh:[1,n] \rightarrow [1,n]$, $Sh(x)=px+q$ fonksiyonunun $1 \rightarrow 1$ bir fonksiyon olup olmadığıdır. Yani farklı indislerdeki elemanları aynı indise atayıp atamayacağıdır.

İspat: $x_1 \neq x_2$ olmak üzere iki farklı indis numaralarına sahip eleman seçelim.
$px_1 + q = px_2 + q \bmod n$ olsun.
$px_1 + q = px_2 + q + kn$
$px_1 = px_2 + kn$
$px_1 - px_2 = kr$
$p(x_1 - x_2) = kn$ bulunur.

i. k=0 ise $p \neq 0$ olduğundan $x_1 - x_2 = 0$ olacağından $x_1 = x_2$ bulunur ki bu baştaki kabulümüze aykırıdır.
ii. $x_1 - x_2 = \frac{nk}{p}$ ise ebob(p,n)=1 olduğundan p|k dir. Yani p, n yi bölmez. p|k olduğunu varsayalım: $\frac{k}{p} = k'$ olsun. Bu durumda $x_1 - x_2 = nk' \Rightarrow x_1 = nk' + x_2$ bulunur. Bu sonuç $x_1 > n$ demektir ki bu durum $x_1 \leq n$ olma durumu ile çelişir.
Dolayısıyla Sh(x) = px+q fonksiyonu mod n e göre $1 \rightarrow 1$ bir fonksiyondur. Yani her $x_1 \neq x_2$ için $Sh(x_1) \neq Sh(x_2)$ olur.

5.2 Geri Dönüşüm İşlemi

Karıştırma işlemi sonucunda elde edilen verinin eski haline gelebilmesi için Sh(x)=px+q fonksiyonunun tersi ile işlem yapılmalıdır.
Sh(x)=px+q mod n fonksiyonun tersi $Sh' = \frac{x-q}{p}$ şeklindedir. Bu doğrusal fonksiyon bir LKÜ olduğundan p ve q değerlerinin mod n e göre tersleri alınarak işlem yapılabilir. p' ve q', p ve q değerlerinin mod n e göre tersi olmak üzere $p' = \frac{1}{p} \bmod n$ ve $q' = -q \bmod n$ değerleri hesaplanırsa $Sh' = (x + q')p'$ mod n halini alır.
Karıştırma işlemi için, dizinin her bir elemanı Sh(x)=px+q mod n fonksiyonunda işleme sokulup yeni yeri belirlenir. Karışık verinin geri dönüşümü için ise $Sh'(x) = (x + q')p'$ mod n fonksiyonu kullanılmalıdır. Karışık haldeki verinin tüm elemanları sırasıyla $Sh'(x)$ fonksiyonunda işleme sokulursa orijinal dizi elde edilmiş olur.
Örnek olarak "this is a message" verisini düşünelim. Bu verideki her bir karakterin ascii tablosundaki karşılığıyla işlem yapılacak olursa 17 elemanlı bir dizi elde edilir. Bu durumda n=17 olur. $2 \leq p \leq n$ ve $2 \leq q \leq n$ olacak şekilde p=11, q=13 seçilsin. Karıştırma işlemi sonucunda "s hssteiem g saai" elde edilir.
this is a message

116	t	115	s
104	h	32	
105	i	104	h
115	s	115	s
32		115	s
105	i	116	t
115	s	101	e
32		105	i
97	a	101	e
32		109	m
109	m	32	
101	e	103	g
115	s	32	
115	s	115	s
97	a	97	a
103	g	97	a
101	e	105	i

s hssteiem g saai

Geri dönüşüm işlemi için $p' = \frac{1}{p} \bmod n = 14$ ve $q' = -q \bmod n = 4$ değerleri hesaplanıp işlem yapılırsa "this is a message" verisine ulaşılır.
s hssteiem g saai

115	s	116	t
32		104	h
104	h	105	i
115	s	115	s
115	s	32	
116	t	105	i
101	e	115	s
105	i	32	
101	e	97	a
109	m	32	
32		109	m
103	g	101	e
32		115	s
115	s	115	s
97	a	97	a
97	a	103	g
105	i	101	e

this is a message

6.Sonuç

Şifreleme işlemlerinin çeşitli aşamalarında veya geri dönüşlü karıştırma işlemlerinde kullanılabilecek olan AKDÜ'nün çalışabilmesi için sadece p,q ve n yeterlidir. Yer değiştirme ve geri dönüşüm işlemleri için fazladan dizinin oluşturulmasına, saklanmasına, iletilmesine gerek yoktur. Bu durum özellikle cihazlar arasında iletişim sırasında, veri boyutunu son derece azaltacağından daha hızlı bir iletişim söz konusu olacaktır.

KAYNAKLAR

[1] https://en.wikipedia.org/wiki/Fisher–Yates_shuffle [Erişim tarihi 05.09.2019]

[2] Hazra T. K., Ghosh R., Kumar S., Dutta S., Chakraborty A. K., File Encryption using Fisher-Yates Shuffle, In 2015 International Conference and Workshop on Computing and Communication (IEMCON) (pp. 1-7). IEEE.

[3] Güvenoğlu E., Esin E. M. (2009), "Knutt / Durstenfeld Shuffle Algoritmasının Resim Şifreleme Amacıyla Kullanılması", Politeknik Dergisi, 12(3), 151-155.

[4] Abdelfatah A. Y., Ayman M. A. (2008), "A Shuffle Image-Encryption Algorithm", Journal of Computer Science, 4(12), 999-1002.

[5] Kaczorek T. (2011), "Checking of the positivity of descriptor linear systems by the use of the shuffle algorithm", Archives of Control Sciences, Vol: 21(LVII), No: 3, 287–298.

[6] Fuller A. T. (1976), "The period of pseudo-random numbers generated by Lehmer's congruential method", The Computer Journal, 19(2), 173-177.