Fırat Üniversitesi Müh. Bil. Dergisi
37(1), 211-222, 2025

Araştırma Makalesi
https://doi.org/10.35234/fumbd.1534269

Matematiksel Yöntemlerle Güçlendirilmiş Yeni Bir Yüksek Güvenlikli Metin Şifreleme Algoritmasının Tasarımı ve Uygulanması

Hüseyin DEMİR[1*], Hünkar ACAR[2]
[1,2]Yazılım Mühendisliği Bölümü, Mühendislik ve Doğa Bilimleri Fakültesi, Samsun Üniversitesi, Samsun, Türkiye
[*1] huseyin.demir@samsun.edu.tr, [2] 211118053@samsun.edu.tr

(Geliş/Received: 16/08/2024; Kabul/Accepted: 24/10/2024)

Öz: Bu çalışmada iyi bilinen geleneksel DES (Data Encryption Standard), AES (Advanced Encryption Standard) ve Hill vb. şifreleme yöntemlerinde meydana gelen eksiklikleri gidermek amacı ile geleneksel yöntemlerden farklı olarak yenilikçi ve mevcut şifreleme yöntemlerindeki boşluklara cevap verebilecek karma bir şifreleme yönteminin tasarımı verilmiştir. Bu yöntem sayesinde daha güvenilir, esnek ve performanslı bir şifreleme yöntemi elde edilecektir. Böylece daha karmaşık ve öngörülemez bir şifreleme süreci sağlayarak saldırılara karşı dayanıklılık artacaktır. Bu sayede günlük konuşmalar ve askeri haberleşmeler gibi çeşitli alanlarda kullanılabilecek güvenliği yüksek bir şifreleme algoritması oluşturulacaktır. Bu yöntem klasik ve modern şifreleme tekniklerinin bir kombinasyonu olarak ortaya çıkmaktadır. Klasik rotor makinelerinin mekanik yapısı ile modern simetrik algoritmaların matematiksel temellerini bir araya getiren bu yöntem hem tarihsel şifreleme metodolojilerini canlandırmakta hem de günümüz güvenlik gereksinimlerini karşılamaktadır. Bu güvenlik gereksinimleri ve yükseklik düzeyi olasılık hesabı ile gösterilecektir. Ayrıca, modüler aritmetik ve rotor mekanizmalarının entegrasyonu, karmaşıklığı ve öngörülemezliği artırarak, kriptanaliz tekniklerine karşı hem etkili bir şifreleme hem de daha güçlü bir savunma mekanizmasına sahip bir karma yöntem sunmaktadır.

Anahtar Kelimeler: Lineer cebir, kaydırma algoritmaları, karıştırma algoritmaları, XOR İşlemi, blok şifreleme, şifrelemenin olasılık hesabı.

Design and Implementation of a New High Security Text Encryption Algorithm Strengthened by Mathematical Methods

Abstract: In this study, a hybrid encryption method is designed to address the shortcomings in well-known traditional encryption methods such as DES (Data Encryption Standard), AES (Advanced Encryption Standard), and Hill cipher. Unlike conventional methods, this new approach offers an innovative solution that fills the gaps in existing encryption techniques. The result is a more secure, flexible, and efficient encryption method. By providing a more complex and unpredictable encryption process, the resilience against attacks will be significantly increased. Consequently, a highly secure encryption algorithm will be created, suitable for various fields such as daily communications and military communications. This method emerges as a combination of classical and modern encryption techniques. By integrating the mechanical structure of classical rotor machines with the mathematical foundations of modern symmetric algorithms, this method not only revives historical encryption methodologies but also meets the security requirements of today. In this study, the security requirements of encryption and the height level will be discussed with probability calculation. Furthermore, the integration of modular arithmetic and rotor mechanisms increases complexity and unpredictability, offering an effective encryption and a stronger defence mechanism against cryptanalysis techniques through this hybrid method.

Keywords: Linear algebra, shift algorithms, mixing algorithms, XOR operation, block cipher, probability calculation of encryption.

1. Giriş

Kriptoloji, verilerin güvenli bir şekilde iletilmesi ve saklanması amacıyla matematiksel tekniklerin kullanıldığı bir bilim dalıdır. Kriptografinin matematiksel temelleri ve şifreleme algoritmalarının kullanılması tarihte ilk çağlarından beri kullanılmış, ilkel hallerden, bilgisayarların kullanıldığı sistematik çağa gelmiştir. Yöntemlerin gelişim sürecini ve bu yöntemlerin günümüz verilerini güvenli bir şekilde koruma işlevi ise önemini korumaktadır [1]. Günümüzde kriptoloji, özellikle internet üzerindeki veri trafiği, finansal işlemler, kişisel bilgilerin korunması ve askeri haberleşmeler gibi kritik alanlarda vazgeçilmez hale gelmiştir. Modern kriptografi, bu süreçleri gerçekleştirmek için çeşitli algoritmalar ve protokoller geliştirmiştir. Bu algoritmalar arasında en bilinenlerinden biri RSA'dır; bu yöntem, büyük asal sayıların çarpanlarına ayrılmasının zorluğuna dayalı olarak güvenlik sağlar [2]. AES (Advanced Encryption Standard) ise hız ve güvenlik dengesi açısından yaygın olarak tercih edilen simetrik bir şifreleme algoritmasıdır [3]. DES ve geliştirilmiş versiyonu 3DES gibi blok şifreleme

[*] Sorumlu yazar: huseyin.demir@samsun.edu.tr. Yazarların ORCID Numarası: [1] 000-0003-3606-878X, [2] 0009-0009-7930-5694

algoritmaları da veri güvenliğini sağlamak için yaygın olarak kullanılmaktadır [4]. Bu yöntemler, farklı alanlarda veri güvenliğini sağlamak için matematiksel temellere dayalı algoritmik teknikler kullanmaktadır [5, 6].

Antik çağlardan günümüze kadar farklı şifreleme yöntemleri geliştirilmiştir. Tarihte en bilinenlerden biri, Gaius Julius Caesar tarafından Galya savaşlarında kullanılan Ceaser Şifrelemesidir; bu şifreleme yöntemi, basit bir kaydırma algoritmasına dayanır [7]. Günümüz savaşlarının etkisi ise özellikle siber ortamın savaş alanı kapsamına girmesi ve bu durumun kriptolojinin evrimine olan katkısı son derece etkilidir. Siber savaşların artan tehdidi, şifreleme tekniklerinin sürekli olarak yenilenmesini ve güçlendirilmesini gerektiren tıpkı eski savaşlarda meydana gelen bir ortam yaratmaktadır [8]. Orta Çağ'da ise polialfabetik bir şifreleme yöntemi olan Vigenère Şifrelemesi, uzun yıllar kırılamamış ve güçlü bir çözüm olarak kabul edilmiştir [9]. Ancak modern dönemde, kriptografik yöntemlerin güçlendirilmesi amacıyla matematiksel temellerin geliştirilmesine yönelik çalışmalar dikkat çekmektedir. Örneğin, integral ve doğal dönüşümlerin şifreleme alanında kullanılması üzerine yapılan araştırmalar, kriptografide yeni yaklaşımlar sunmaktadır [10]. Fırat Üniversitesi Mühendislik Bilimleri Dergisi ile Multimedia Tools ve Applications dergisinde yer alan her iki çalışmada da integral dönüşümüne dayalı şifreleme tekniklerinin veri güvenliğine katkıları incelenmiştir [11, 12]. Diğer taraftan Laplace Dönüşümü, kriptosistemlerin tasarımında iyi bir uygulama alanı olmasına rağmen, bazı çalışmalarda kriptanaliz çalışmaları yeterli düzeyde yapılmadığından dolayı birçok şifreleme algoritması güvenli iletişim için yetersiz kalmaktadır. Yapılan bu çalışmada, Laplace Dönüşümü tabanlı şifreleme sistemlerinin güvenlik analizleri için genel bir saldırı senaryosu önerilmiş ve bu senaryo yeni bir şifreleme yöntemi üzerinde uygulanmıştır [12]. Ayrıca Lucas Dizisi ve Fibonacci Dizisi gibi matematiksel diziler kullanılarak yeni şifreleme yöntemlerinin geliştirildiği çalışmalar da literatürde yer almaktadır. Bu diziler, veri şifreleme süreçlerinde alternatif çözümler sunarak güvenliği artırmayı hedeflemektedir [13]. Ek olarak genetik kod yöntemlerini kullanarak yeni bir veri şifreleme algoritması tasarlayan farklı yaklaşım çalışmalarda mevcuttur. Çalışmada, genetik algoritmaların temel prensipleri ve uygulamaları üzerine odaklanılmıştır. Araştırma, veri güvenliğini artırmak için genetik algoritmaların kullanıldığı yeni bir şifreleme yöntemi geliştirilmesi sürecini detaylandırmaktadır [14].

Bu çalışmanın temel amacı, veri güvenliğini sağlamak için yenilikçi bir şifreleme algoritması geliştirmektir. Çalışmada, şifreleme sürecinin karmaşıklığını artırarak güvenliği güçlendiren matematiksel yöntemler bir araya getirilmiştir. Bu yöntemler arasında lineer cebir, matris işlemleri, kaydırma algoritmaları, karıştırma algoritmaları ve XOR işlemi gibi teknikler bulunmaktadır. Böylece, şifreleme algoritmasının kırılmasını zorlaştıran ve güvenlik seviyesini artıran bir yapı oluşturulmuştur. Bu çalışma, sadece metin şifreleme süreçlerine odaklanmakta ve bu bağlamda geliştirilen yeni bir metin şifreleme algoritmasını sunmaktadır. Geliştirilen algoritmanın hem günlük konuşmaların güvenliğini sağlamak hem de askeri haberleşmeler gibi kritik güvenlik gerektiren alanlarda kullanılabilirliği hedeflenmiştir.

Mevcut şifreleme algoritmalarının zayıf noktaları analiz edilerek bu eksiklikler giderilmiş ve geliştirilmiş bir yapı önerilmiştir. Özellikle, matris işlemleri ve kaydırma algoritmaları ile şifreleme süreci karmaşıklaştırılmış, XOR işlemi ise şifre çözümlemeyi zorlaştıran ek bir güvenlik katmanı olarak eklenmiştir. Bu sayede metin şifreleme sürecinin kırılma olasılığı minimize edilmiştir.

Amacımız, bu çalışma ile veri güvenliğini sağlamaya yönelik yenilikçi ve etkili bir çözüm sunmaktadır. Bu sebeple, diğer şifreleme algoritmalarının karmaşıklığından kaynaklanan zorluklar giderilmiş ve anlaşılması kolay, ancak güvenliği yüksek bir şifreleme algoritması önerilmiştir. Bu algoritmanın, günlük haberleşmelerden savunma sanayine kadar geniş bir kullanım alanı bulması hedeflenmektedir.

2. Geliştirilen Şifreleme Algoritması

Geliştirilen şifreleme algoritması, veri güvenliğini artırmak için bir dizi matematiksel işlem ve teknik içermektedir. Algoritma, temel olarak blok şifreleme mantığı ile çalışır ve şifreleme işlemi 12 karakterlik bloklar halinde gerçekleştirilir. Aşağıda algoritmanın temel prensipleri ve süreçleri detaylı bir şekilde açıklanmıştır:

2.1. Anahtar yönetimi

Algoritma, simetrik anahtar şifreleme yöntemi kullanmaktadır. Bu, şifreleme ve deşifreleme işlemleri için aynı anahtarın kullanıldığı anlamına gelir. Anahtarın güvenliği, algoritmanın genel güvenliğini doğrudan etkiler. Anahtar, rastgele üretilir ve şifreleme işlemi başlamadan önce taraflar arasında güvenli bir şekilde paylaşılır.

2.2. Blok şifreleme

Algoritma, veriyi 12 karakterlik bloklara böler. Her blok, şifreleme işlemi için bağımsız olarak işlenir. Bu yaklaşım, büyük veri setlerinin işlenmesini daha yönetilebilir hale getirir ve paralel işleme imkanını artırır.

2.3. Matematiksel işlemler

Şifreleme işlemi birkaç matematiksel işlemden oluşur: ilk olarak her blok, bir matris temsilcisi olarak işlenir, böylece veriler bloklar halinde matrislere dönüştürülerek matematiksel manipülasyonların uygulanmasını kolaylaştırılır. Bu aşamada, matris çarpması ve dönüşüm işlemleri kullanılarak veri karmaşıklığı artırılır. Daha sonra blok verisi üzerinde kaydırma işlemleri uygulanarak, veri desenleri karıştırılır. Bu işlemler, verinin analizi ve şifre çözme sürecini zorlaştırır. Güvenlik açısından verinin daha da karmaşık hale gelmesi için karıştırma algoritmaları kullanılır. Bu, verinin anlamlı yapılarının ve kalıplarının bozulmasına yardımcı olur. Son olarak şifreleme sürecinde XOR (Exclusive OR) işlemi, veriyi anahtar ile kombinlemek için kullanılır. Bu işlem, verinin bit düzeyinde değiştirilmesini sağlar ve şifreleme işleminin güvenliğini daha da artırır.

2.4. Şifreleme süreci

Şifreleme sürecinde ilk adım, giriş verisinin 12 karakterlik bloklara ayrılmasıdır. Bu bloklar, matris şeklinde yapılandırılarak işlem yapılabilir hale getirilir. Matrislerin üzerinde kaydırma ve karıştırma algoritmaları uygulanır. Kaydırma işlemi, her matrisin satırlarının belirlenen sayıda sola veya sağa kaydırılması şeklinde gerçekleştirilir. Bu işlem, verinin düzenini bozarak güvenliği artırır. Karıştırma aşaması ise matrisin sütunlarındaki elemanların rastgele bir şekilde yer değiştirmesiyle yapılır. Sonraki adımda, her blok matris, belirlenen bir anahtar ile XOR işlemine tabi tutulur. Bu işlem, şifreleme sonucunu daha karmaşık hale getirir ve verinin güvenliğini güçlendirir. Nihayet, tüm bloklar birleştirilerek şifrelenmiş veri oluşturulur. Bu kombinasyon hem kaydırma ve karıştırma işlemlerinin hem de modüler aritmetiğin getirdiği matematiksel karmaşıklıkla güçlü bir güvenlik sağlar.

2.5. Deşifreleme süreci

Deşifreleme, bilgi güvenliği ve gizlilik açısından kritik bir rol oynar ve veri şifreleme sistemlerinin etkili bir şekilde çalışmasını sağlar. Deşifreleme işlemi, şifreleme sürecinin tersidir: Şifrelenmiş metin veya veri, deşifreleme işlemine başlanmadan önce elde edilir ve sonra veri 12 karakterlik bloklara bölünür. Şifreleme sırasında kullanılan anahtar veya anahtar bilgisi, deşifreleme işleminde de kullanılır. Anahtar, veriyi doğru şekilde deşifrelemek için gereklidir. Böylece, şifrelenmiş bloklar anahtar ile XOR ile ters işlemine tabi tutulur. Daha sonra şifreleme algoritmasının tersine çalışacak bir algoritma uygulanır. Bu, şifrelenmiş veriyi çözmek için gerekli matematiksel işlemleri içerir. Bu işlem sırasında kaydırma ve karıştırma algoritmaları ters yönde uygulanır ve matrisler geri dönüştürülür ve verinin orijinal hali elde edilir. Sonunda deşifrelenmiş bloklar birleştirilir ve işlem tamamlandığında, orijinal metin veya veri elde edilir ve okunabilir hale gelir.

2.6. Güvenlik analizi

Geliştirilen algoritma, mevcut şifreleme yöntemlerinin zayıf noktalarını ele alacak şekilde tasarlanmıştır. Matris işlemleri, kaydırma ve karıştırma teknikleri, XOR işlemi ile birleştirilerek veri güvenliğini artırır. Matematiksel temellere dayanan bu yöntemler, algoritmanın karmaşıklığını artırır ve güvenliğini sağlar.

Bu algoritmanın, günlük konuşmalardan askeri haberleşmelere kadar geniş bir uygulama yelpazesi hedeflenmiştir. Şifreleme ve deşifreleme süreçleri, anlaşılması kolay ve etkili bir güvenlik çözümü sunmak amacıyla tasarlanmıştır.

2.7. Şifreleme koşulları ve anahtar matrisinin özellikleri

Algoritmanın belirli koşulları ve gereksinimleri bulunmaktadır. Bu gereksinimler, belirli bir anahtar ile şifreleme işlemi yapıldıktan sonra deşifreleme işleminin doğru bir şekilde gerçekleştirilmesini sağlamaktadır. İlerleyen bölümlerde bu koşullar matematiksel olarak detaylandırılacaktır; burada ise yüzeysel olarak açıklanacaktır.

İlk koşul, anahtarın belirli bir örnek teşkil eden tablodaki karakterlerin sayısal değerlerine bakılarak 4x4 boyutunda bir matris haline getirilmesidir. Aynı şekilde, şifrelenmek istenen metin de bu tablodan alınan sayısal değerlerle 3x4'lük bir matris haline getirilip hazır tutulur. Şifreleme işlemi, bu iki matrisin kullanılmasıyla gerçekleştirilir. Şifreleme işlemi, anahtar matris (4 × 4) ile 12 karakterlik düz metni 3 × 4'lük bir matris haline getirip çarpmakla başlar. Daha sonra, elde edilen 3 × 4'lük matrisin değerlerinin 63'e göre mod değeri alınır. Bunun nedeni, Tablo 1' de gösterildiği 63 karakterin bulunması ve değerlerin bu aralıkta kalmasını sağlamaktır.

Bu noktada önemli bir koşul, deşifreleme işleminin anahtar matris kullanılarak yapılacak olması nedeniyle, anahtar matrisin determinantının sıfırdan farklı olması gerekliliğidir. Bu, matrisin tersinin alınabilmesi için

zorunludur. Anahtar matrisin tersi varsa deşifreleme işlemi yapılabilir. Bir diğer koşul ise anahtar matrisin determinantı ile 63 değeri arasında asal olmasıdır. Bu, modüler tersin bulunarak anahtar matrisin tersinin ek matrisi ile çarpılması ve mod 63 alınarak düz metnin elde edilmesi için gereklidir.

Bu koşullar sağlandıktan sonra, anahtar değerinin sayısal değerleri toplanır ve mod 10 değeri alınır. Elde edilen bu sonuç, matrislerin çarpımından elde edilen sonucun mod 63 işleminden sonra, değerlerin S1 ve S2 rotorlarından geçmeden önce bu rotorların kaydırma işleminde kullanılır. Örneğin, sonuç 5 ise bu, rotorlardaki sayıların 5 kaydırılması anlamına gelir. Bu adım, sonuçların daha karmaşık hale getirilmesini sağlar. Daha sonra, mod değeri kullanılarak sıfırdan dokuza kadar olan karıştırma algoritması uygulanır ve bu rotorlardan çıkan değerler indekslerine göre karıştırılır.

Son aşamada, anahtar herhangi bir işleme tabi tutulmadan S3 rotorlarından geçirilir ve bu değerler, yukarıda bahsedilen işlemlerden elde edilen değerlerle birlikte XOR işlemine tabi tutulur. Bu işlemlerin sonucunda, şifrelenmiş metin elde edilir. Burada gizlilik tamamen anahtarın gizliliğine bağlıdır; algoritma bilinse dahi anahtar olmadan deşifreleme işlemi yapılamaz.

Tablo 1. Sayı ve harf değerleri modeli.

Sayı	Sayı Değeri	Sayı	Harf Değeri	Sayı	Harf Değeri	Sayı	Harf Değeri	Sayı	Harf Değeri	Sayı	Harf Değeri
0	0	11	A	22	L	33	W	44	h	55	s
1	1	12	B	23	M	34	X	45	i	56	t
2	2	13	C	24	N	35	Y	46	j	57	u
3	3	14	D	25	O	36	Z	47	k	58	v
4	4	15	E	26	P	37	a	48	l	59	w
5	5	16	F	27	Q	38	b	49	m	60	x
6	6	17	G	28	R	39	c	50	n	61	y
7	7	18	H	29	S	40	d	51	o	62	z
8	8	19	I	30	T	41	e	52	p		
9	9	20	J	31	U	42	f	53	q		
10		21	K	32	V	43	g	54	r		

Tablo 2. Düz metin ve anahtar değerleri.

Düz Metin		Anahtar	
Harf	Değeri	Harf	Değeri
s	55	m	49
a	37	a	37
b	38	t	56
a	37	e	41
h	44	m	49
l	1	A	11
		t	56
		i	45
		K	21

Şifreleme işleminin doğru ve güvenli bir şekilde yapılabilmesi için aşağıdaki iki temel koşulun sağlanması gerekmektedir. İlk olarak anahtar matrisinin determinantı sıfırdan farklı olmalıdır. Bu koşul, şifrelemenin ve deşifrelemenin matematiksel olarak geçerli olabilmesi için gereklidir. Şifreleme ve deşifreleme işlemlerinde matrisin tersi kullanılır, dolayısıyla matrisin tersi alınabilmelidir. Aksi takdirde bu durum şifreleme sürecinin geçersiz olmasına neden olur. Bir diğer şart ise matrisin determinantı, kullanılan modül değeri ile aralarında asal olmalıdır. Bu, matrisin tersinin bulunabilmesi ve şifreleme sürecinin matematiksel geçerliliği için önemlidir. Şifreleme ve deşifreleme işlemlerinde, determinantın modüler tersinin alınması gerektiğinden, bu ters değerin modül ile aralarında asal olması gerekir. Bu işlem ile şifrelenmiş verilerin doğru şekilde geri dönüştürülmesini sağlanır. Bu iki şartın kontrolü yapıldıktan sonra öncelikle anahtar matrisi oluşturma ve şifreleme süreci tamamlanmalıdır. Fakat anahtar matrisini oluştururken, anahtarın boyutunun şifreleme algoritmasının

gereksinimlerine uygun olması gerektiğine dikkat edilmelidir. Örneğin, 4 × 4 boyutunda bir matris oluşturmak için anahtar karakterlerini sayısal değerlere dönüştürüp matrise yerleştirmek gerekir. Eğer anahtar, 4 × 4 matris için yeterli uzunlukta değilse, eksik kalan kısımlar en baştan itibaren tekrar kullanılarak tamamlanır.

3. Deneysel Çalışma

Şimdi düz metin "sabah1" ifadesini anahtar kelime "matemAtiK" olmak üzere yukarıda anlatılan sürece göre işleme tabi tutalım. Burada düz metin ve anahtar ifadelerinin karakterlerinin sayısal değerleri Tablo 2 ile verilmiştir.

Anahtar matrisinin sayısal değerlerinin toplamı, mod 10 işlemine tabi tutulur. Bu sonuç, genellikle şifreleme sürecindeki çeşitli parametreleri belirlemede kullanılır. Örneğin, bu değer S1, S2 ve S3 rotorlarında veya karıştırma algoritmalarında kayma ve indeks karıştırma işlemlerinin nasıl yapılacağını belirlemek için bir metrik olarak kullanılabilir. Örnek olarak, anahtar matrisindeki sayısal değerlerin toplamı 45 ise ve bu toplam mod 10 işlemine tabi tutulduğunda sonuç 5 olacaktır. Bu sonuç, ilgili şifreleme rotorlarının ve algoritmalarının ayarlarını yapılandırmada kullanılır.

Bu koşulların ve hesaplamaların doğru bir şekilde uygulanması, güvenli ve etkili bir şifreleme algoritması geliştirmek için esastır. Şifreleme ve deşifreleme süreçlerinin matematiksel geçerliliği, bu temel prensiplere uygunlukla sağlanır.

Böylece anahtar "matemAtiK" olmak üzere anahtarın Tablo 2 ile verilen değerlerinden 4x4 lük bir matris oluşturulur. Böylece bu matrisin determinant değeri -2393684 olarak bulunur.

Böylece, determinant değerinin sıfırdan farklı olma şartı sağlanmaktadır. Diğer taraftan mod değeri ile determinant değerleri aralarından asal olup olmadıkları modüler aritmetik uygulanarak kontrol edilirse gerçekten mod değeri ile $\det(A)$ değerleri aralarından asal olduklarından dolayı diğer şartın da sağlandığı gösterilmiş olur. Böylece her iki şart sağlanmış olduğundan şifreleme işlemine devam edilirse ilk olarak anahtar matris ile Tablo 2'de verilen sayısal değerlerden oluşan düz metin matrisinin matris çarpımları yapılır ve değer matrisi değerleri mod 63 ile hesaplanır. Böylece mod 63 işlemine göre çıkan değerler 19, 6, 35, 56, 26, 36, 48, 8, 41, 49, 43, 10 olarak bulunur.

Daha sonra da indeks karıştırma ve kaydırma işlemleri uygulanır. Karıştırma işlemine başlamadan önce mod 10'a göre anahtar matrisinin tek veya çift olduğunu kontrol etmek önemlidir. Eğer mod 10 sonucunda elde edilen değer tek ise, indeks ve karıştırma işlemlerinden elde edilen sonuçları XOR işleminde tersine çevirmek gerekir. Eğer mod 10 sonucunda elde edilen değer çift ise, indeks ve karıştırma işlemlerinden elde edilen sonuçları XOR işleminde olduğu gibi kullanmak gerekir. Bu süreç, şifreleme algoritmasının güvenliğini ve doğruluğunu sağlamak için gereklidir. O halde Tablo 2 ile belirtilen anahtar değerleri toplanıp çıkan sonucun mod 10 değeri alınarak devam edilir. Buna göre, 49 + 37 + 56 + 41 + 49 + 11 + 56 + 45 + 21 = 365 olmak üzere 365 mod 10 = 5 olarak hesaplanır. Dolayısıyla Tablo 3 ile verilen karıştırma algoritması, yukarıda belirtilen kurallara göre dizilerin içindeki değerlerin yer değiştirilmesini sağlar ve aşağıdaki kurallara göre indeksler değiştirilir:

Tablo 3. İndeks karıştırma algoritma modeli.

Indeks	Yer değiştiren indeks değerleri
0:	Aynen kalır
1:	2-7, 8-11 indeksleri yer değiştirir.
2:	0-5, 3-7 indeksleri yer değiştirir.
3:	3-4, 1-6 indeksleri yer değiştirir.
4:	5-9, 10-11 indeksleri yer değiştirir.
5:	1-2, 4-9 indeksleri yer değiştirir.
6:	4-10, 5-11 indeksleri yer değiştirir.
7:	0-1, 2-4 indeksleri yer değiştirir.
8:	3-11, 6-7 indeksleri yer değiştirir.
9:	7-8, 2-9 indeksleri yer değiştirir.

Karıştırma işlemi sonucunda elde edilen değerler 19, 35, 6, 56, 49, 36, 48, 8, 41, 26, 43, 10 olarak bulunur. Daha sonra bu değerler, S1 ve S2 rotorlarından geçer. Başlangıçta S1 ve S2 rotorlarının değerleri rastgele belirlenmiştir. Anahtar değerinin sayısal değerlerinin toplamının mod 10 ile elde edilen sonucu doğrultusunda, buradaki sayılar kayma işlemine tabi tutulur. Örneğin, mod 10 sonucunda 5 elde edildiğinde, S1 rotorundaki

değerler 5 birim kaydırılırken, S2 rotorundaki değerler 5+1=6 birim kaydırılır. Bu ek kayma işlemi, mod 10 sonucunun sıfır (0) olduğu durumlarda bile uygulanır, bu yüzden her durumda artı bir (+1) kayma eklenir.

Bu şekilde, algoritmanın çalışma mantığı belirli bir düzen içinde verilerin işlenmesini ve şifreleme sürecini sağlar.

Tablo 4. S1 rotor kaydırma ve yeni değerleri modeli.

S1 Eski Rotor	S1 Yeni Rotor
0: 4	0: 9
1: 8	1: 2
2: 0	2: 6
3: 5	3: 3
4: 1	4: 7
5: 9	5: 4
6: 2	6: 8
7: 6	7: 0
8: 3	8: 5
9: 7	9: 1

Tablo 5. S2 rotor kaydırma ve yeni değerleri modeli.

S2 Eski Rotor	S2 Yeni Rotor
0:9	0:8
1:5	1:0
2:4	2:3
3:6	3:2
4:8	4:7
5:0	5:1
6:3	6:9
7:2	7:5
8:7	8:4
9:1	9:6

Bu işlemde, verilen eski değerler şifreleme algoritmasının rotorlar aracılığıyla nasıl dönüştürüldüğünü anlamak için detaylandırılmıştır. İşlem adımları şu şekildedir:

3.1 Eski değerlerin dönüştürülmesi

İlk olarak, her bir eski değeri S1 yeni ve S2 yeni rotorları kullanarak dönüştüreceğiz. Bu işlemde önce S1 yeni rotorundan, ardından S2 yeni rotorundan geçilerek sonuçlar elde edilecektir. Bu şekilde rotor dönüşüm adımları aşağıdaki gibi uygulanır.

- Örnek: Eski değeri 19 olan bir değeri ele alalım.
- S1 Yeni Rotoru: Değer 1 olduğunda, S1 yeni rotorunda 1 değeri, 2 değerine karşılık gelir.
- S2 Yeni Rotoru: Elde edilen 2 değeri, S2 yeni rotorunda da 3 değerine karşılık gelir.
- Sonuç olarak, 1 değeri 1'den 3'e dönüştürülür.
- Aynı işlemi diğer değerler için de uygulayacağız. Buna göre eski değeri 9 olan değer için işlemi uygularsak:
- S1 Yeni Rotoru: Değer 9 olduğunda, S1 yeni rotorunda 9 değeri, 1 değerine karşılık gelir.
- S2 Yeni Rotoru: Elde edilen 1 değeri, S2 yeni rotorunda 0 değerine karşılık gelir.

Sonuç olarak, 9 değeri 0'e dönüştürülür. En sonunda 19 giren değer rotorlardan 30 olarak çıkar.

Eski değerlerin tamamı üzerinde bu dönüşüm işlemi yapılır ve yeni değerler elde edilir. Bu işlemde sırasıyla elde edilen yeni değerler şunlardır: 30, 27, 4, 74, 50, 24, 51, 1, 53, 94, 52, 36.*

Bu dönüşüm işlemi, belirli bir algoritmanın parçası olarak verilerin güvenli bir şekilde şifrelenmesi amacıyla gerçekleştirilir. Fakat ilk önce anahtar değerimizin sayısal değeri S3 rotorundan geçmesi gereklidir. Anahtar değerlerinin işlenmesi şu adımlarla gerçekleştirilir:

3.2 Anahtarın sayısal toplamı ve mod 10 hesaplaması:

Anahtarın sayısal değerleri toplanır ve bu toplamın mod 10 değeri hesaplanır. Bu değer, Tablo 6 ile gösterilen S3 rotorundaki kaydırma miktarını belirler. Elde edilen kaydırma miktarı bir sonraki aşama olan XOR işlemi için tek veya çift olma durum kontrolüne göre S3 rotorundan geçen sayıların tersten ya da aynen yazılmasını da belirler.

Mod 10 sonucuna göre:

- Tek ise: S3 rotorundan çıkan değerler XOR işleminde tersten yazılır.
- Çift ise: S3 rotorundan çıkan değerler XOR işleminde aynen yazılır.

Bu adımlar, şifreleme sürecinin anahtar yönetimini ve verilerin güvenliğini sağlar.

Tablo 6. S3 rotor kaydırma ve yeni değerleri modeli.

S3 Eski Rotor	S3 Yeni Rotor
0: 4	0: 3
1: 2	1: 0
2: 8	2: 1
3: 5	3: 9
4: 6	4: 7
5: 3	5: 4
6: 0	6: 2
7: 1	7: 8
8: 9	8: 5
9: 7	9: 6

Anahtar Tablo değerleri: 49, 37, 56, 41, 49, 11, 56, 45, 21
S3 Rotorundan Çıkan Değerler: 76, 98, 42, 70, 76, 0, 42, 74, 10

Bu çalışmada verilen değerlerin şifreleme işlemi şu şekilde gerçekleştirilmiştir: Öncelikle, matris çarpımı ve karıştırma algoritmalarından elde edilen yeni değerler hesaplanmıştır. Bu değerler sırasıyla 30, 27, 4, 74, 50, 24, 51, 1, 53, 94, 52, 36 olarak elde edilmiştir. Ardından, bu değerler S3 rotorundan geçirilmiştir. S3 rotorunun kaydırma miktarı ve rotor değerleri göz önüne alındığında, elde edilen değerler 76, 98, 42, 70, 76, 0, 42, 74, 10 olarak bulunmuş fakat sonucun doğru hesaplanabilmesi için bu değerler tersten yazılarak 10, 74, 42, 0, 76, 70, 42, 98,76 şeklinde düzeltilmiştir. Çünkü mod 10 değerinden çıkan sonucun tek olmasından dolayı S3 rotorundan çıkan değerler tersten yazılmalıdır. Bu işlemlerden elde edilen sonuçlar Tablo 7 ile gösterilmektedir. Bu değerlerin ASCII' deki karşılığına bakarak elde edilen karakterleri görebiliriz.

Tablo 7. XOR işlem modeli.

Yeni değerler	30	27	4	74	50	24	51	1	53	94	52	36	
S3 Rotorundan çıkan değerler	10	74	42	0	76	70	42	98	76	10	74	42	(Tersten yazılır)
XOR Sonuçları	20	81	46	74	126	94	25	99	121	84	126	14	

XOR Sonuçları: Son olarak, elde edilen S3 rotor değerleri ile yeni değerlere XOR işlemi uygulanarak şifrelenmiş metin oluşturulmuştur. Bu işlem sonucunda elde edilen XOR sonuçları 20, 81, 46, 74, 26, 94, 25, 99, 121, 84, 126, 14 şeklindedir. Bu adımlar, verilerin güvenli bir şekilde şifrelenmesini sağlar ve sonuç, standart ASCII karakter tablosuna karşılık gelen metinle eşleşir. Örnek olarak şifrelenmiş değer içerisinde 20 sayısı standart ASCII tablosunda '\x14' hex karakterine, 81 sayısı ise Q karakterine tekabül eder. Diğer şifrelenmiş değerde bulunan sayılarda standart ASCII tablosundan tekabül ettiği karakterlere bakılarak yazılır. Bu şifrelenmiş değerler, mod 63 uygulanmaksızın, ASCII tablosundaki karşılık geldiği karakterle eşlenirse, şifrelenmiş metin

“14Q.J~^19cyT~0e” olmak üzere bu şifre metnin açılmış hali “14 Q. J ~ ^ 19 c y T ~ 0e“olarak elde edilir. Burada, dikkat edilirse, ASCII tablosunda 0-31 ve 127 değerlerine karşılık gelen herhangi bir karakter olmadığından bu değerlerin hex karşılıkları alınmıştır.

Sonuç olarak, XOR işlemi sonucunda şifrelenmiş metin elde edilmiştir. Anahtar uzunluğumuz 9 karakter ve düz metin uzunluğumuz 6 karakterdir. Matris çarpımının doğası gereği, 3 × 4’lük bir matris ile 4 × 4’lük bir matris çarpıldığında, 3 × 4’lük bir matris elde edilir. Bu nedenle, şifreleme sürecinde her zaman 12 karakterli bir çıktı elde ederiz. Anahtar uzunluğunun 9 karakter olması, anahtarın 3 kez tekrar etmesine neden olmaktadır. Ancak, matris çarpımı, mod 63 işlemi, indeks karıştırma algoritmaları ve S1, S2, S3 rotorları sayesinde farklı değerler elde edilebilmektedir. Bu açıklanan süreç, şifreleme işlemini gerçekleştirmektedir.

3.3 Deşifreleme işlemi

Deşifreleme işlemi ise şifreleme adımlarının tersine gerçekleştirildiği ve şifrelenmiş metni veya veriyi anlaşılır bir biçime dönüştürmek için belirli adımları içerir. Bu süreç aşağıda ifade edilen işlemler kümesi ile uygulanır. İşlem verinin gizliliğini ve güvenliğini korurken, doğru anahtar ve algoritma kullanılarak geri dönüşüm imkânı sağlar.

Deşifreleme işlemini gerçekleştirebilmek için öncelikle şifreleme sırasında kullanılan anahtara ve şifrelenmiş metne sahip olmamız gerekmektedir. Tablo 2 ile verilen anahtar ifadesi “matematik” değerleri XOR işlemine tabi tutularak sonuçlar Tablo 7 ile gösterilmiştir. Böylece bu değerlere göre şifrelenmiş metin olarak “14Q.J~^19cyT~0e” elde edilmiştir.

İlk adımda, şifrelenmiş metnin ASCII değerleri elde edilir. Bu işlem, şifreleme aşamasında kullanılan XOR işlemlerinin tersini uygulayarak orijinal değerleri geri elde etmeyi sağlar. Ancak, bu adım öncesinde anahtarın tablodaki değerlerine bakılmalı, bu değerler toplanmalı ve mod 10 değeri hesaplanmalıdır. Elde edilen mod 10 sonucu, S3 rotorundaki kaydırma miktarını belirleyecektir. Tablo 7 ile verildiği gibi bu kaydırma, şifreleme sırasında S3 rotorunun nasıl kaydırıldığını anlamak ve deşifreleme sürecinde doğru şekilde uygulamak için kritik öneme sahiptir.

Buna göre Tablo 2 ile verilen anahtar değerleri ile Tablo 6 ile verilen S3 rotor kaydırma işlemi ve yeni değerler kullanılarak deşifreleme işlemini uygulamaya devam edeceğiz. Böylece kaydırma işlemi için anahtar değerin tablo değerleri toplanır ve mod 10 değeri alınır. Bu değerlerin toplamı 49 + 37 + 56 + 41 + 49 + 37 + 56 + 45 + 21 = 391 olmak üzere mod 10 değeri 391 mod 10 = 1 olarak bulunur. Daha sonra bulunan 1 değeri, S3 rotorunda işleme tabi tutulup 1 kaydırma işlevini yerine getirerek yeni değerler elde edilir. Yeni değerler tekrar aynı şekilde 49, 37, 56, 41, 49, 11, 56, 45, 21 ve S3 rotorundan çıkan anahtar yeni değerleri de 76, 98, 42, 70, 76, 0, 42, 74, 10 olacak şekilde tekrar elde edilir.

Şifrelenmiş metnin ASCII değerlerini elde edilmesinin nedeni ise şifrelemede kullanılan XOR işlemini tersine çevirerek orijinal verilere ulaşmak içindir. Bu işlem, şifreleme sırasında uygulanan matematiksel ve algoritmik adımların tersini içerir. Öncelikle, anahtarın sayısal değerlerinin toplamı hesaplanır ve bu toplamın mod 10 değeri alınarak, deşifreleme sırasında S3 rotorundaki kaydırma miktarı belirlenir. Örneğin, “matematik” anahtarının ASCII değerlerinin toplamı 365 olarak bulunur. Bu değerin mod 10 sonucu 5 değeridir, yani S3 rotoru için 5 kaydırma yapılacaktır. Bu kaydırma sonucunda S3 rotorundan geçen değerler tersten yazılır.

Deşifreleme adımlarında, şifrelenmiş metindeki her bir değer, S3 rotorundan geçirilir ve kaydırma sonucunda elde edilen yeni değerlerle işlem yapılır. Bu yeni değerler, XOR işlemi kullanılarak elde edilen sonuçlarla karşılaştırılır. Sonuç olarak, şifrelenmiş metni geri elde edebilmek için XOR işlemi ve kaydırma adımlarının tersine uygulanır. Değerlerin tekrar etmesi, anahtarın yeterli olamaması nedeniyle bazı verilerin aynı kalmasına yol açabilir. Bu işlem, verinin güvenliğini sağlamak için kullanılan karmaşık algoritmaların etkinliğini test etmek açısından önemlidir.

XOR (eXclusive OR), yani “özel veya” işlemi, dijital mantıkta önemli bir işlemdir ve özellikle kriptografi gibi alanlarda sıkça kullanılır. XOR işleminin temel mantığı, iki girişin karşılaştırılması üzerine kuruludur. Giriş değerleri farklı ise sonuç 1, girişler aynı ise sonuç 0 olur. Bu sayede, iki bitlik veriyi karşılaştırarak onları şifreleme, hata bulma gibi işlemlerde kullanılabilir hale getirir. XOR işleminin tanımı:

Tablo 8. XOR işlem şeması.

Birinci bit	İkinci bit	XOR işlemi sonucu oluşan bit
1	0	1
0	0	0
1	1	0
0	1	1

Tablo 8 ile gösterilen işlem, özellikle şifreleme algoritmalarında (örneğin OTP- One Time Pad) kullanılır. Örneğin onluk sayı sisteminden iki sayı 45 ve 24 olmak üzere, bu sayılarının decimal (onluk) hallerini XOR işlemine sokalım. Ancak önce bu sayıları binary (ikilik) sisteme çevirmemiz gerekiyor, çünkü XOR işlemi binary değerler üzerinden yapılır. Bu sayıların sırasıyla ikilik sistemdeki karşılıkları Denklem 1 ile gösterilmiştir.

$$45 = 101101_2$$
$$24 = 011000_2 \quad (1)$$

Bu sayılar XOR işlemine tabi tutulduğunda ikilik tabanda 110101 değeri elde edilir. Elde edilen ikilik taban değerinin onluk tabanda karşılığı ise 53 sayına tekabül eder. Benzer işlemler her bir yeni değere uygulanırsa elde edilen XOR sonuçları Tablo 9 ile verilmiştir. Buna göre bulunan kaydırma oranı kullanılarak S3 rotoru tersten yazılmak suretiyle elde edilen değerler XOR ile işleme tabi tutularak yeni değerler "30, 27, 4, 74, 50, 24, 51, 1, 53, 94, 52, 36" olarak bulunmaktadır.

Tablo 9. XOR işlem modeli.

XOR Sonuçları	20	81	46	74	126	94	25	99	121	84	126	14	
S3 Rotorundan çıkan değerler	10	74	42	0	76	70	42	98	76	10	74	42	(Tersten yazılır)
Yeni Değerler	30	27	4	74	50	24	51	1	53	94	52	36	

S1 ve S2 rotorlarının kaydırma işlemleri, anahtarın mod 10 ile hesaplanan değerine göre belirlenir. Anahtarın ASCII değerlerinin toplamı alındıktan sonra bu toplamın mod 10 değeri hesaplanır. Bu mod 10 sonucu, S1 ve S2 rotorlarının kaydırma miktarını belirler. Kaydırma işlemi, rotorlardaki indekslerin konumlarını değiştirir. Bu sayede verinin şifrelenmesi veya deşifre edilmesi sürecinde rotorların etkili bir şekilde kullanılmasını sağlar.

Tablo 4 ve Tablo 5 ile gösterilmiş olan bu değerler şifreleme işleminin tersi yönünde işleme tabi tutulup, daha sonra indeks karıştırma işleminden geçerek Tablo 2 ile verilmiş olan değerlerin orijinal halinin elde edilmesi gerekmektedir. Bu yol, S2 rotorundan başlayarak geriye doğru S1 rotoruna giderek ilk değerleri elde etmemize yardımcı olur.

Şifrelenmiş metnin üzerindeki değerler, indeks karıştırma işleminden geçmiştir. Bu değerler Tablo 3 ile gösterilmiştir. Bu nedenle, bu değerlerin orijinal, karıştırma öncesi halleriyle elde edilmesi gerekmektedir. Bunun için, anahtar değerinin tablo değerlerinin toplamı hesaplanır ve mod 10 işlemi uygulanarak bir sonuç elde edilir. Bu sonuç, hangi karıştırma algoritmasının uygulandığını belirlemede kullanılır. Örneğin, anahtar değerinin toplamı 365 olup, bu değerin mod 10 işlemi sonucu 5 elde edilmiştir. Bu durumda, 5 değerine göre belirlenen karıştırma algoritması uygulanmıştır.

Tablo 3 ile indeks karıştırma algoritmasının kuralları verilmiştir. Deşifreleme işlemi yaparken tersten hareket edildiği için yukarıdaki işlemde anahtardan elde edilen mod 10 değeri ile çıkan sonuç değeri ile kural belirlenir, daha önceden karıştırılan değerler bu vasıta ile eski karıştırılmamış indekslerine geri getirilir. Çıkan değer 5, kuralı ise Tablo 3 ile gösterildiği gibi 1-2, 4-9 indeksleri yer değiştirmiştir. Bunu göz önünde bulundurarak eski değerlere doğru gidilir. Böylece en son 19, 6, 35, 56, 26, 36, 48, 8, 41, 49, 43, 10 değerleri elde edilir. (Bu değerler daha sonra kullanılarak düz metin elde edilecektir.)

Deşifreleme işlemi, şifrelenmiş veriyi geri dönüşüm sağlayarak orijinal düz metne ulaşmak için belirli adımları içerir. İlk olarak, anahtar matrisinin determinantı hesaplanır. Bu hesaplama modüler aritmetik işlemleri için kritik bir rol oynar. Şöyle ki matrisin determinantı sıfırdan farklı olması bize matrisin tersinin varlığını sağlayacaktır. Diğer taraftan determinant değeri ile 63 mod değeri aralarında asal olmalıdır. Bu özellik ise modüler tersin varlığını sağlamak için gereklidir. Bu iki şartın sağlandığı şifreleme aşamasında gösterilmişti. Böylece bu asal ilişki sağlandığından, determinantın modüler tersi bulunabilir ve bu değer, anahtar matrisinin tersini

hesaplamak için kullanılır. Daha sonra, şifrelenmiş değerler, anahtar matrisinin tersi ile çarpılır. Çarpım işlemi sonucunda elde edilen her bir değere mod 63 işlemi uygulanır. Bu adım, genellikle matrislerin belirli bir aralıkta kalmasını garantiler ve verilerin uyumlu bir şekilde analiz edilmesine olanak tanır. Sonuç olarak, bu işlemler neticesinde elde edilen değerler düz metne dönüştürülür. Bu adımlar, şifrelenmiş metni geri çözmek ve orijinal metni elde etmek için gerekli olan standart kriptografik işlemleri içerir. Bu işlemleri, işlem sırasına uygun olarak uygularsak determinant değeri ile anahtar matrisin ek matrisini kullanarak anahtar matrisin tersini genelleştirilmiş Öklid Algoritmasına göre mod 63 uygulayarak buluruz [1]. Böylece elde edilen yeni matris:

$$\begin{pmatrix} -41314 & -1638 & 79158 & -47594 \\ -76194 & 81338 & -7028 & -2548 \\ -31780 & -1260 & -31174 & 55454 \\ 103160 & -69724 & -45682 & -16562 \end{pmatrix} \quad (1)$$

olarak bulunur. Böylece başvuru matrisi olarak artık yeni elde edilen (1) matrisi kullanılacaktır. Şimdi şifrelenmiş verinin elde edilmesi için şifrelenmiş değer matrisi ile elde edilen başvuru matrisi çarpılır. Çarpım sonucunda elde edilen matris, başlangıçtaki verinin şifrelenmiş hali olarak kabul edilir. Burada, şifrelenmiş matrisin elemanlarının belirli bir aralıkta kalmasını sağlamak amacıyla, her bir eleman mod 63 işlemine tabi tutulur. Bu yöntem, özellikle sınırlı sayıda olasılığın bulunduğu tablolarda kullanılır ve veri bütünlüğünün korunmasına yardımcı olur. Böylece bu işlemle çıkan düz metin matrisi yeniden elde edilir. Tablo 2 ile verilmiş düz metin "sabah1" deşifrelenmiş metin olarak elde edilir.

Görüldüğü üzere birçok karmaşık işlemden geçmiş, çeşitli matematiksel yöntemlerle işleme tabi tutulup hesaplanmış yollarla şifreleme ve deşifreleme işlemi yapılmıştır. Böylece "matemAtiK" anahtarı ile şifrelediğimiz "sabah1" düz metnini tekrar anahtar kullanarak deşifreleme işlemine tabi tutup aynı düz metin elde edilmiştir.

4. Güvenliği Artırılmış Şifrelemenin Olasılık Hesabı

Bu bölümde ilk olarak şifreleme aşamasında oluşması muhtemel olan olasılık hesaplarını yaparak çalışmanın güvenilirlik seviyesinin üst seviyede olduğunu ve şifreleme yapısının frekans analizine karşı güçlü bir direnç sağladığının matematiksel hesaplarını vereceğiz. Bu hesaplamayı çalışmamızın güvenilirliğini arttırmak için kullandığımız kaydırma ve yer değiştirme işlemlerine bağlı olarak yapacağız. Bu işlemleri gösterebilmek için sırasıyla aşağıdaki olasılık hesapları yapılmalıdır.

4.1. Anahtar seçimi olasılığı: İlk olarak, anahtar seçiminde 63 karakterden oluşan bir tablo olduğunu ve anahtar uzunluğunun 9 karakter olduğunu belirtelim. Bu duruma göre olasılık hesaplarını yapıp karıştırma algoritması, S1 ve S2 rotorlarının da kriptografi algoritmasına katmış olduğu olasılığı analiz edelim. Bu durumda her karakterin olasılığı bulunur.

Burada her karakter bağımsız olarak seçildiği için olasılık 63 tabandaki her bir seçim için 1/63 ihtimali ile çarpılır. Böylece karakterlerin olasılığı toplam $(1/63)^9$ olmak üzere düşük ihtimalli bir olasılığı ifade eder.

4.2. Matris çarpımı ve mod 63: Şifreleme algoritmasında anahtar 4x4'lük, düz metin ise 3x4'lük bir matris olarak tanımlanmıştır. Şifreleme sürecinin bu aşamasında, iki matrisin çarpımı gerçekleştirilir. Matris çarpımı sonucunda elde edilen her bir eleman, mod 63 işlemi ile işlenir. Bu adım, şifreleme sürecinin temel taşlarından biridir ve şifrelenmiş verinin rastgelelik düzeyini artırarak frekans analizine karşı ek bir koruma sağlar.

Çarpım sonucu elde edilen her bir matris elemanına mod 63 işlemi uygulandığında, sonuçlar 0 ile 62 arasında bir değer aralığına indirgenir. Bu işlem, büyük sayıların daha küçük ve kontrol edilebilir bir alanda tutulmasını sağlarken, şifrelenmiş metindeki sembollerin frekanslarının homojen bir şekilde dağılmasına katkıda bulunur. Böylece, frekans analizine karşı direnç oluşturulmuş olur.

4.3 Karıştırma algoritması olasılık hesabı: Her karıştırma adımı, bir mod 10 sonucuna göre uygulanır ve 0'dan 9'a kadar olan değerlerden biri çıkabilir. Bu, 1/10 olasılığa sahiptir. Bu durumda karıştırma işlemi uygulanacak ve karışıklık indekslerine bağlı olarak permütasyon yapılacaktır.

Bir adımda, belirli indeksler arasında 4 farklı yer değiştirme uygulanmaktadır (örneğin, 0-5, 3-7 gibi).

Bu işlemde her iki indeks arasında farklı yer değişimleri vardır. Bir adımda 4 indeks yer değiştirdiği için, bu adımda olasılıkları hesaplarken 4! kullanılırız, çünkü bu 4 indeksin tüm permütasyonları dikkate alınmalıdır. 4! = 24 değerine eşittir. Dolayısıyla, her karıştırma adımında mevcut 4 indeksin farklı dizilimlerinden biri seçileceği için, bu adımın toplam karıştırma olasılığı ise 1/240 olarak bulunur.

4.4. S1 ve S2 rotorlarının olasılığı: S1 ve S2 rotorları, anahtar değerlerine bağlı olarak mod 10 ile belirlenen bir kaydırma işlemi gerçekleştirir. Her iki rotor için de kaydırma olasılığı, mod 10 işleminin sonucuna göre belirlenir ve S1 ve S2 rotoru için her bir olasılık değeri 1/10 olarak bulunur.

S1 ve S2 rotorlarının kaydırma işlemleri bağımsız olduğundan, her rotor kendi kaydırma olasılıkları çerçevesinde işlem yapar. Bu işlem, her rotorun farklı bir kayma miktarı ile çalışmasına olanak sağlar. Örneğin, S1'in bir kaydırma yapması ve S2'nin iki kaydırma yapması gibi farklı durumlar, olasılık hesaplamasına ek bir katman getirir. Bu adım, frekans analizine karşı algoritmanın güvenliğine önemli bir katkı sağlar, çünkü her rotor, her seferinde farklı sonuçlar üretebilir.

4.5. S3 rotoru ve XOR işlemi olasılığı: S3 rotorunda da benzer şekilde mod 10 ile belirlenen kaydırma işlemi yapılır. Bu, 1/10 olasılığa sahiptir. Ancak bu rotorun farkı, mod 10 sonucuna göre elde edilen değerin ters çevrilip çevrilmeyeceğidir. Bu işlemde iki olasılık vardır: Değer ya aynı kalır ya da tersine çevrilir yani 1/2 değerini alır. Bu durumda her bir kaydırma sonucunun olasılığı S3 rotoru için 1/20 olarak bellidir.

Bu aşamada, mod 10 sonucuna bağlı olarak kaydırma sonrası elde edilen değerlerin bir kısmı tersine çevrilir. Ardından, XOR işlemi devreye girer ve bu kaydırmalar ile elde edilen değerler, algoritmanın sonraki adımlarında kullanılmak üzere işlenir.

4.6. Frekans analizine karşı direnç: Algoritmanın frekans analizine karşı direncini değerlendirmek için, şifreleme sürecindeki her adımın şifreli metin üzerindeki etkisini ve genel olasılık dağılımını incelemeliyiz:

Matris Çarpımı ve Mod 63 İşlemi: Matris çarpımının ardından uygulanan mod 63 işlemi, şifreli karakterlerin olasılık dağılımını homojen hale getirir. Her bir şifreli karakterin 0 ile 62 arasında eşit olasılıkla ortaya çıkması sağlanır. Bu homojen dağılım, şifreli metinde belirgin frekans paternlerinin oluşmasını engeller ve frekans analizini zorlaştırır.

Karıştırma Algoritması ve Permütasyonlar: Karıştırma algoritması, mod 10 sonucuna göre belirlenen ve her biri farklı permütasyonlara sahip yer değiştirmeleri içerir. Örneğin, 4 indeksin yer değiştirmesiyle oluşan permütasyonların sayısı 4!= 24 olarak bellidir. Bu yüksek permütasyon sayısı, şifreli metindeki karakter dizilimlerini büyük ölçüde çeşitlendirerek frekans analizine karşı ek bir koruma sağlar.

S1, S2 ve S3 Rotorları: Rotorlar, anahtar değerlerine bağlı olarak farklı kaydırma işlemleri uygularlar. S1 ve S2 rotorlarının kaydırma olasılıkları 1/10, S3 rotorunun ise 1/20 olarak tanımlanır. Bu rotorlar, şifreleme sürecine ek rastgelelik katarak kriptanaliz saldırılara karşı direnci artırır.

XOR İşlemi: Son aşamada gerçekleştirilen XOR işlemi, önceki adımlardan elde edilen değerleri anahtar ile birleştirerek şifreli metnin düzenini daha da karmaşık hale getirir. Bu işlem, düz metin ile şifreli metin arasındaki doğrudan ilişkileri gizleyerek frekans analizinin etkinliğini azaltır.

4.7. Toplam olasılık hesaplaması: Şifreleme algoritmasının toplam olasılık değerini hesaplayabilmek için 4.1, 4.3, 4.4 ve 4.5 alt bölümlerinde verilen olasılık değerlerinin çarpımı bize toplam olasılıkların değerini verecektir. Böylece toplam olasılık değeri 1.33×10^{-22} olarak hesaplanır.

Bu hesaplama sonucunda algoritmanın kırılma olasılığı oldukça düşük seviyelere ineceği açıkça ortaya konmuştur. Her adımın karmaşıklığı ve şifrelenmiş metnin frekans düzenlerinin bozulması, olası saldırılara karşı yüksek bir direnç seviyesi sağlar.

Sonuç olarak, algoritmanın her aşamasında uygulanan olasılık hesaplamaları, şifreleme yapısının frekans analizine karşı güçlü bir direnç sağladığını matematiksel olarak kanıtlamaktadır. Bu, algoritmanın güvenlik seviyesinin oldukça yüksek olduğunu göstermektedir.

5. Sonuç

Bu çalışmada, şifreleme ve deşifreleme süreçlerinin detaylı incelenmesi, kullanılan algoritmaların mantığının ve adımlarının açıklanması amaçlanmıştır. Şifreleme ve deşifreleme işlemleri, modern kriptolojinin temelini oluşturan matris çarpımı, modüler aritmetik ve rotor sistemleri gibi matematiksel ve teknik bileşenleri içerir. İlk aşamada, şifrelenecek metin belirli bir anahtar kullanılarak şifrelenmiştir. Şifreleme işlemi için ilk aşamada anahtar matris kullanımı için belirlenen kelimenin ASCII değerleri ve bunların belirli bir tabloya göre mod 10'na göre sonuçları ile işleme tabi tutularak rotorlar vasıtasıyla kaydırma işlemi uygulanarak yeni değerler elde edilmiştir. Daha sonra elde edilen yeni değerler, belirli bir algoritma ile karıştırılarak ve mod 63'e göre şifrelenmiş metin oluşturulmuştur. Bu süreçte XOR işlemleri sayesinde verilerin güvenli bir şekilde şifrelenmesini sağlamıştır. Daha sonra ise çalışmanın en önemli kısmını oluşturan şifrelenmiş metni geri çözmek için deşifreleme işlem kısmıdır. Bu kısım birçok önemli aşamalardan oluşmaktadır. İlk olarak deşifreleme sürecindeki anahtar matrisinin

determinantı hesaplanmalıdır. Çünkü bu matrisin determinantının modüler tersinin kontrolü ve determinant ile mod 63 arasındaki asal ilişkiyi ortaya koyması açısından çok önemlidir. Böylece anahtar matrisin tersinin bulunması formülü kullanılarak, şifrelenmiş metni geri çözme işlemi başlatılmıştır. Daha sonra da modüler ters aritmetik uygulanması ile şifrelenmiş değerler düz metne dönüştürülmek üzere, mod 63'e göre yeniden düzenlenmiştir.

Bu çalışmamızda, hesaplamalar için Python 3.8.10 sürümü kullanılmıştır ve mevcut şifreleme yöntemlerinin zayıf yönlerinin işlemlere etkisini azaltarak oluşturulan yeni yöntem sayesinde daha sağlam ve güvenilir bir yapı oluşturulmuştur. Bu sayede, güvenlik ve performans gereksinimlerini karşılayan, yenilikçi ve etkili bir şifreleme yöntemi sunulmuştur. Geliştirdiğimiz karma teknik, özellikle büyük ölçekli veri güvenliği, iletişim güvenliği ve kaynak kısıtlı sistemlerde etkili çözümler sunarak hem teorik hem de pratik açıdan literatüre değerli bir katkı sağlamaktadır. Ayrıca çalışmamız kriptografi alanında önemli bir yenilik olarak değerlendirilebilir ve bu yönü ile de ileriye dönük araştırmalara da ilham verecektir.

Kaynaklar

[1] Akleylek S, Akyıldız E ve Çimen C. Şifrelerin matematiği: Kriptografi. ODTÜ Yayıncılık, 2007.
[2] Beşkirli A, Özdemir D ve Beşkirli M. Şifreleme yöntemleri ve RSA algoritması üzerine bir inceleme. Avrupa Bilim ve Teknoloji Dergisi 2019; (Özel Sayı), 284-291.
[3] Topaloğlu N, Calp MH ve Türk B. Bilgi Güvenliği Kapsamında Yeni Bir Veri Şifreleme Algoritması Tasarımı ve Gerçekleştirilmesi. Bilişim Teknolojileri Dergisi 2016; 9(3) 291-300.
[4] Birer GC. Kriptoloji. Bilim ve Teknik 2022; 17-31.
[5] Gümüş E. Kuantum kriptografi ve anahtar dağıtım protokolleri. Akademik Bilişim'11- XIII. Akademik Bilişim Konferansı Bildirileri; 2-4 Şubat 2011; İnönü Üniversitesi, Malatya. 547-552.
[6] Avaroğlu E. Bilgi güvenliğinin temel yapı taşı: Kriptoloji. Düşünce Dünyasında Türkiz 2017; 8(43), 53-65
[7] Freeman P. Julius Caesar. İstanbul: Kronik Kitap, 2019.
[8] Çiftçi H. Her Yönüyle Siber Savaş. Ankara: TÜBİTAK Yayınları, 2013.
[9] Chindhe AD. and Kiwne S. Application of Natural Transform in Cryptography. Journal of New Theory, 2017; (16), 59-67.
[10] Gençoğlu MT. Kriptolojide İntegral Dönüşümünün Kullanımı. Fırat Üniversitesi Mühendislik Bilimleri Dergisi 2016; 28(2), 217-220.
[11] Gencoglu MT. Embedded image coding using laplace transform for Turkish letters. Multimed Tools Appl 78 2019; 17521–17534.
[12] Gençoğlu MT. Cryptanalysis of a new method of cryptography using Laplace transform hyperbolic functions. Communications in Mathematics and Applications 2017; 8(2), 183–189.
[13] Güney Duman M and Duman M. Encryption and Decryption of the Data by Using the Terms of the Lucas Series. Duzce University Journal of Science and Technology 2021; 9(3), 1-7.
[14] Zengin M and Albayrak Z. Designing a new data encryption algorithm using a genetic code method.Acta Polytechnica Hungarica 2022; 19(2), 235-252.