Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, 34, S. 639-644, Mart 2022

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 34, pp. 639-644, March 2022

Research Article

SecureRandom Kütüphanesi Kullanarak Yazılımsal Trivium Oluşturma

Cemile İnce[1*], Kenan İnce[2], Davut Hanbay[3]

[1*] İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Malatya, Türkiye, (ORCID: 0000-0002-4638-8501), cemile.ince@inonu.edu.tr
[2] İnönü Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, Malatya, Türkiye (ORCID: 0000-0003-4709-9557), kenanince@gmail.com
[3] İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Malatya, Türkiye (ORCID: 0000-0003-2271-7865), davut.hanbay@inonu.edu.tr

(2nd International Conference on Applied Engineering and Natural Sciences ICAENS 2022, March 10-13, 2022)

(**DOI:** 10.31590/ejosat.1084005)

ATIF/REFERENCE: İnce, C., İnce, K. & Hanbay, D. (2022). SecureRandom Kütüphanesi Kullanarak Yazılımsal Trivium Oluşturma. *Avrupa Bilim ve Teknoloji Dergisi*, (34), 639-644.

Öz

Bu çalışmada yazılımsal trivium yapısı oluşturulmuştur. Trivium, 3 adet LFSR'nin (Doğrusal geri beslemeli kaydırmalı yazmaç) çeşitli mantık kapılarıyla birbirine bağlanmasıyla oluşturulan ve rasgele sayı üreteci olarak kullanılan yapılardan biridir. Trivium yapıları donanımsal veya yazılımsal oluşturulabilir. Yazılımsal trivium yapılarının ilk değerlerini (anahtar, başlangıç vektörü ve LFSR başlangıç içerikleri) belirlemek için literatürde önerilen çeşitli yöntem ve algoritmalar mevcuttur. Bu çalışmada mevcut çalışmalardan farklı olarak ilk şartlar Java SecureRandom kütüphanesi kullanlarak oluşturulmuştur. Çalışmada oluşturulan yazılımsal trivium yapısı kullanılarak üretilen sözde rasgele sayılar NIST SP 800-22 Rev.1a testleri ile analiz edilmiştir. Oluşturulan trivium yapısı kullanılarak üretilen ikili diziler rasgelelik testlerinden başarılı şekilde geçmiştir.

Anahtar Kelimeler: Trivium, LFSR (Linear Feedback Shift Register), SRSÜ (Sözde Rasgele Sayı Üreteçleri), NIST 800-22 Rev.1a, SHA (Secure Hash Algorithm).

Generating Software Trivium Using SecureRandom Library

Abstract

In this study, a software trivium structure was created. Trivium is one of the structures created by connecting 3 LFSRs (Linear feedback shift registers) with various logic gates and used as a random number generator. Trivium structures can be created in hardware or software. There are various methods and algorithms proposed in the literature to determine the initial values (key, initial vector and LFSR initial contents) of software trivium structures. In this study, unlike the existing studies, the first conditions were created by using the Java SecureRandom library. Pseudo-random numbers generated using the software trivium structure created in the study were analyzed with NIST SP 800-22 Rev.1a tests. The binary sequences produced using the generated trivium structure passed the randomness tests successfully.

Keywords: Trivium, LFSR (Linear Feedback Shift Register), PRNG (Presudo-Random Number Generators), NIST 800-22 Rev.1a, SHA (Secure Hash Algorithm).

[*] Sorumlu Yazar: cemile.ince@inonu.edu.tr

1. Giriş

İnternet teknolojilerinin yaygınlaşmasıyla birlikte aktarılan verilerin boyutları da aynı oranda artmaktadır. İletilen verilerin güvenliğinin sağlanması da veri iletişimi kadar önemli bir konudur. Kişisel verilerin korunmasının yasalarla güvence altına alındığı günümüzde retina, parmak izi, iris, DNA dizilimi gibi birçok kişisel veri iletim sırasında, veri tabanlarında şifreli bir şekilde tutulmalıdır. Şifrelemede anahtar alan oluşturmak için farklı yaklaşımlar mevcuttur. Bütün yaklaşımların temelinde rassallık kavramına rastlanır. Rasgele sayılar günlük hayatın her aşamasında kullanılır. Örneğin cep telefonumuz bir baz istasyonunu kaydederken baz istasyonu telefonun kimliğini doğrulamak için rasgele sayı gönderir. Ya da banka hesabınıza giriş yaparken kullanıcı tarayıcısı ile banka sunucusu kendi arasında rasgele sayı alışverişi ile haberleşir. Veri güvenliği söz konusu olan bütün alanlarda bir şekilde rassalık kavramı kullanılmaktadır. Bunun temel sebebi rassalık kavramının tahmin, taklit ve tekrar edilemez olmasıdır.

Hassas verilerin güvenliğinin sağlanması için birçok şifreleme yöntemi mevcuttur. Verilerin özelliklerine göre değişiklik gösteren bu yöntemler, genel olarak metin, resim, ses ve video şifreleme olarak kategorize edilebilir. Metin şifreleme yöntemleri genellikle resim ya da video şifrelemede çok başarılı değillerdir. Çünkü görüntü veya video verilerini oluşturan pikseller arasındaki korelasyon yüksektir (Özkaynak, 2011).

Hayatın dijitalleşmesi ile birlikte, kişisel verilere saldırı oranları yüksek oranda artmıştır. HIPA journal tarafından yapılan araştırmaya göre kişisel verilere yönelik saldırılar, diğer verilere yapılan saldırılara göre iki kat fazla artış göstermiştir(Arrachid vd., 2014). Kişisel verilerin korunmasında şifre, örüntü, parmak izi ve retina gibi anahtar alanları kullanılmaktadır. Parmak izi ve retina taklit edilemez olduğundan güvenli gibi görünse de bu verilerin saklanması farklı teknikleri gerektirir. Şifre veya örüntü kullanımında ise, rassallık kavramı ön plana çıkmaktadır.

Kriptoloji işlemlerinde anahtar alan olarak sıklıkla rasgele sayı üreteçleri (RSÜ) kullanılmaktadır. RSÜ'ler iki ana kategoride incelenir:

1. Gerçek Rasgele Sayı Üreteçleri (GRSÜ)
2. Sözde Rasgele Sayı Üreteçleri (SRSÜ)

GRSÜ'ler genel olarak fiziksel olaylarla veya özel donanımlar kullanılarak oluşturulurlar. SRSÜ'ler ise, yazılımsal olarak oluşturulan sistemlerdir. GRSÜ'leri kullanmak SRSÜ'lere göre daha zor ve maliyetlidir. Ancak güvenlik analizlerinde SRSÜ'lerden daha iyi performans gösterirler. Fakat erişim ve kullanım kolaylığından SRSÜ'ler daha yaygın kullanım alanına sahiptirler.

SRSÜ'ler yazılım tabanlı üreteçlerdir. Genel olarak farklı algoritmalar olmakla birlikte, bir tohum değeri ile başlayan ve algoritmik olarak ihtiyaç duyulan uzunlukta sayı dizisini oluştururlar.

LFSR'ler genellikle devrelerin olası girdilerine test örüntüleri üretmek için kullanılır. Şifrelemede anahtar alan için üretilen rasgele sayıların güvenli olabilmesi için doğrusallık göstermemesi beklenir. LFSR'ler matematiksel olarak ilkel polinomlarla hesaplanır (Beletsky, 2021). LFSR doğrusal bir yapıya sahiptir. Bu sebeple LFSR ile üretilen ikili dizilerin istatistiksel rassallık testlerinden başarılı sonuç üretmesi mümkün değildir. Bu sebeple farklı uzunluklardaki LFSR yapıları kullanılarak karmaşıklık arttırılır. Hesaplanma kolaylığından dolayı ihtiyaç duyulan uzunlukta LFSR 'ler üretilebilir.

Bütün bilimsel çalışmalarda olduğu gibi, RSÜ'lerin de kabul edilmiş ve tekrarlanabilir kriterleri olmalıdır. Bu anlamda rassallık test ortamları bulunmaktadır. Zaman içerisinde önerilmiş olan 15 istatistiksel test NIST tarafından SP 800-22 Rev.1a (NIST test ortamı olarak kullanılacaktır) adı altında bir test ortamı haline getirilmiştir. Literatürde en çok kullanılan ortam SP 800-22 Rev.1a ortamıdır.

NIST test ortamının bazı alt testlerinde bir milyon bit üzerinde uzunluk gereksinimi bulunmaktadır. Bu anlamda, RSÜ olarak LFSR kullanmak için, çok uzun LFSR yapıları kurulabilir. Ancak bu durum uygulanabilir olmayacaktır. Kısa boyutlu LFSR'ler, doğrusal yapısından ötürü tahmin ve tekrar edilebilir olacaktır. Bu durumda kriptografik olarak güvenli bir RSÜ olmayacaktır. Bu sebeple LFSR yapılarının kullanıldığı güvenli bir SRSÜ elde edebilmek için birden fazla LFSR farklı karıştırma ve birleştirme mekanizmaları ile birlikte kullanılır. Hatta bazı çalışmalarda en temel seviyede XOR 'lama ile de yeterli güvenlik seviyelerinin sağlandığı belirtilmiştir (Manikya vd., 2021). Eşzamanlı şifreleme sistemlerinin gerekli olduğu alanlarda işlem hızı büyük önem taşımaktadır. LFSR'ler hızlı sistemler olduğundan yaygın olarak kullanılırlar. Güçlü şifreleme için LFSR'nin kombinasyon halinde kullanıldığı ve literatürde trivium diye adlandırılan kombine LFSR'ler de vardır. RC4, A5/3, Seal vb. algoritmalar güncel olarak kullanılmaktadırlar (İslam&Hak, 2016). LFSR yapıları özellikle akış şifrelemede kullanıma uygundurlar. Çünkü doğrusal yapısı ile, doğrusal veri akışı üzerinde uygulanması kolaydır. Ayrıca FPGA geliştirme kartları kullanılarak gerçekleştirilebilir ve bu RSÜ'in istatistiksel rassalık testlerinden geçtiği çalışmalar mevcuttur (Ravichandran vd., 2018).

LFSR görüntü şifreleme uygulamasında yine rasgele sayı üreteci olarak kullanılmaktadır. (Mondal vd., 2016), çalışmalarında, permütasyon-ikame mimarisi ile LFSR kullanarak oldukça güvenli olduğunu belirtilen bir şifreleme algoritması önermektedir. Görüntü pikselleri LFSR yardımı ile karıştırılır. Elde edilen ara görüntü piksel değerleriyle RC4 algoritmasının ürettiği değerler XOR işlemine tabi tutularak üretilen rasgele sayı dizisinin güvenlik analizleri gerçekleştirilmiş ve önerilen yöntemin güvenli olduğu belirtilmiştir.

LFSR'nin matematiksel temeli ilkel polinomlardır. Galois ve Fibonacci matrisleri kullanılarak üretilen PRNG çalışmasında, üreteç çeşitliliğini arttırarak kriptografik açıdan güçlü LFSR'ler elde edildiği belirtilmiştir (Goresky & Klapper, 2002). Böylece ilkel polinomdan ilkel olmayan polinom üretilmiş ve indirgenemez polinom kullanılmıştır. Sonuç olarak indirgenemez polinom ile Galois üreteci uygulanmıştır. Galois ve Fibonacci matrisleri kullanılarak oluşturulan bir başka LFSR çalışmasında (Chakrobotry vd., 2014), iki modun (Fibonacci ve Galois) aynı kayıtta gerçekleştirilmesini öneren çalışma kaotik lojistik haritası kullanılarak oluşturulmuştur. Ayrı modlara sahip rasgele sayılara kıyasla daha iyi sonuçlar verdiği açıklanmıştır. Sonuç olarak üretilen 149 bitlik anahtarın büyük boyutlu olması sayesinde her türlü kaba kuvvet saldırılarına karşı dirençli olduğu belirtilmiştir.

Medikal görüntü şifrelemede kullanılmak üzere rasgele sayı üreten bir başka çalışmada ise şifreleme için kaotik haritaları ve LFSR'yi birlikte kullanarak sözde iki rasgele sayı dizisi oluşturulmuştur (Zode & Deshmukh, 2019). Daha sonra bu iki dizi XOR işlemine tabi tutularak güvenli anahtar elde edilmiştir. Kaotik haritalar, algoritmaya rasgele davranış sağlarken, başlangıç koşullarına aşırı hassas olması da sisteme ek güvenlik katkısında bulunmuştur.

LFSR yapıların birbilerine and, or, xor gibi mantıksal kapılarla bağlandığı yapılar da mevcuttur. Bunlar, trivium yapılar

olarak adlandırılır. (Zode, Deshmukh, 2019) çalışmasında, FPGA temelli trivium ile rasgele üretilen sayılar, NIST test ortamından geçirilerek başarılı sonuçlar elde edildiği belirtilmiştir. (Garipcan vd.,2017), çalışmasında sonuç olarak Nist test ortamında 13 alt testten başarılı bir şekilde geçirilmiş ancak NIST testlerinin tamamını kapsayan bir çalışma olmamıştır. Nist test ortamı, 15 alt testten oluşmakta ve bir testin geçirilmemiş olması, üretilen sayıların yeterli güzel istatistiksel özellik göstermemesi anlamını taşımaktadır.

LFSR yapılarındaki ortak sorun sayılar arasında deterministik bir yapının bulunmasıdır. Üretilen sayılar arasındaki korelasyonu ortadan kaldırmak için trivum yapısını kullanan (Kaya, 2020), çalışmasında, GRSÜ üretmek için memristor ve trivium tabanlı elde edilen sayılar NIST test ortamı ile test edilmiştir. Sonuçta üretilen rasgele sayıların birçok alanda kullanılabileceği ifade edilmiştir.

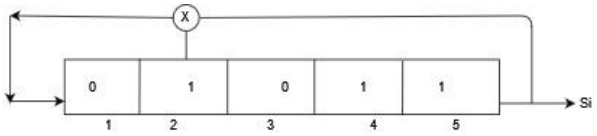
Görüntü şifrelemede anahtar alan oluşturmak için kullanılan trivium yapısını kullanan bir diğer çalışmada ise (Etem&Kaya, 2020), SRSÜ tasarımı için LCG algoritması oluşturulduktan sonra trivium ile son işlemden geçirilerek elde edilen sayılar NIST test ortamından geçirilmiştir. Sonuçlar NIST test ortamının 15 alt testinden başarı ile geçememiştir. Başarılı rasgele sayılar elde etmek için LCG algorimasından geçirilmiştir. Bu çalışmanın dezavantajı göreceli fazladan işlem yüküdür. Zira trivium başlı başına karmaşık işlemler içeren bir yapıdır ve ekstra LCG algoritması da eklenince görüntü şifreleme gibi senkron yapılı şifrelemenin önemli olduğu sistemlerde kullanımı kısıtlı olabilecektir.

2. Materyal ve Yöntem

Bu çalışmada, rasgele sayı üreten LFSR ve trivium yapıları kıyaslanmıştır. Doğrusal yapılar olan LFSR'lerin doğrusalliği, SRSÜ'lerin tahmin edilmez olması özelliğine aykırıdır. Bu durumu düzeltmek için LFSR'ler mantık kapılarıyla birleştirilerek trivium yapıları oluşturulmuştur. Doğrusal olmayan triviumlar, iyi istatistiksel özelliklere sahiptir.

2.1. LFSR (Doğrusal Geri Beslemeli Kaydırmalı Yazmaç)

LFSR, maksimum peryotlu ikili diziler üreten doğrusal yapılardır. Peryot, kendini tekrar etmeye başlamadan önce üretilen bit dizisi uzunluğudur. Geri beslemeli denilmesinin sebebi ise kaydedicinin bazı bitlerinin özel mantıksal işlemlere tabi tutularak çıktı bitlerinin elde edilmesidir. LFSR'ler ile donanımsal yada yazılımsal olarak SRSÜ kaynağı olarak kullanılabilir. Donanımsal olarak flip flop ve xor kapıları ile üretilen SRSÜ, yazılımsal olarak çeşitli yöntem ve algoritmalarla elde edilirler. LFSR tarafından türetilen diziler, geleneksel Berlekamp Massey algoritması kullanılarak oluşturulur (Berlekamp, 2015). Bu algoritma temel olarak, belirli bir ikili dizi çıkışı için en kısa LFSR yi bulan bir algoritmadır. Ayrıca algoritma rasgele dizi üreten yapı için minimal matematiksel polinomu da bulur. Bu algoritma sıfır olmayan tüm öğelerin bir çarpımsal tersi olması gerektiğini de söyler (Reeds & Sloone, 1985). Temel seviyede bir LFSR'nin görüntüsü Şekil 1 'deki gibi gösterilebilir. Ayrıca şekildeki başlangıç değerlerine göre 10 bit üretim sonucu Tablo 1'de hesaplanarak sunulmuştur.



Şekil 1. Örnek bir LFSR

Şekil 1 deki örnek bitler için LFSR Tablo 1'de sunulduğu şekilde sonuçlar üretir.

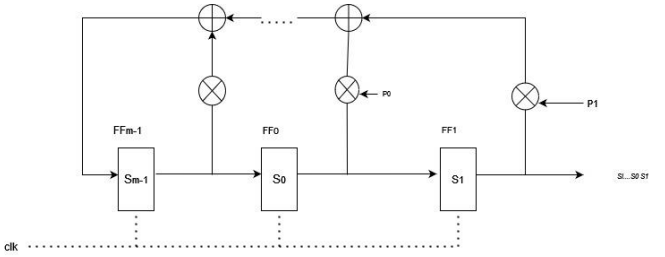
Tablo 1. LFSR 'nin 14 adım çalıştırılması sonucu elde edilen ikili bit çıkış dizisi

İşlem 1	İşlem 2	Bit Dizisi	Çıkış Biti	Toplam Çıkış Bit Dizisi (Si)
Tüm bitler bir bit sağa	S_1=$S_2 \oplus S_5$	01011	1	1
Tüm bitler bir bit sağa	S_1=$S_2 \oplus S_5$	00101	1	11
Tüm bitler bir bit sağa	S_1=$S_2 \oplus S_5$	10010	1	111
Tüm bitler bir bit sağa	S_1=$S_2 \oplus S_5$	01001	0	0111
Tüm bitler bir bit sağa	S_1=$S_2 \oplus S_5$	00100	1	10111
Tüm bitler bir bit sağa	S_1=$S_2 \oplus S_5$	00010	0	010111
Tüm bitler bir bit sağa	S_1=$S_2 \oplus S_5$	00001	0	0010111
Tüm bitler bir bit sağa	S_1=$S_2 \oplus S_5$	10000	1	10010111
Tüm bitler bir bit sağa	S_1=$S_2 \oplus S_5$	01000	0	010010111
Tüm bitler bir bit sağa	S_1=$S_2 \oplus S_5$	10100	0	0010010111

LFSR'ler den faydalanarak sistemin ilkel polinomu da elde edilebilir. Şekil 1 deki örnek LFSR de hangi bitlerle işlem yapılmısa ilkel polinom da ona göre oluşturulur. Şekil 1 de 1, 2 ve 5. Bitler ile işlem yapıldığından derecesi 5 olan ilkel polinom $x^5 + x^2 + x^1$ 'dir. Bu ilkel polinomda dizilerin maksimum uzunluğu $2^n - 1$ 'dir. Tablo 1 için üretilecek maksimum bit uzunluğu $2^5 - 1 = 31$ bittir. İlkel polinom hesaplama için genel denklem Denklem 1'deki gibi elde edilir:

$$P(x) = x^m + P_{m-1}x^{m-1} + \cdots + P_1 x + P_0 \quad (1)$$

Bir LFSR nin donanım mantığı Şekil 2 deki gibidir. Şekil 2 incelendiği zaman geri besleme yolunun aktif olup olmaması P_0 , P_1 , P_2 , P_3 ... ... P_{m-1} katsayılarının belirlediği görülür.



Şekil 2. m dereceli LFSR yapısı

$$P_i = \begin{cases} 1\,; kapalı\ anahtar \\ 0\,; açık\ anahtar \end{cases}$$

$$S_m = S_{m-1}P_{m-1} + \cdots + S_1P_1 + S_0P_0 \quad (2)$$

$$S_{m+1} = S_mP_{m-1} + \cdots + S_2P_1 + S_1P_0 \quad (3)$$

$$S_{i+m} = \sum_{j=0}^{m-1} P_j \, . S_{i+j} \quad (4)$$

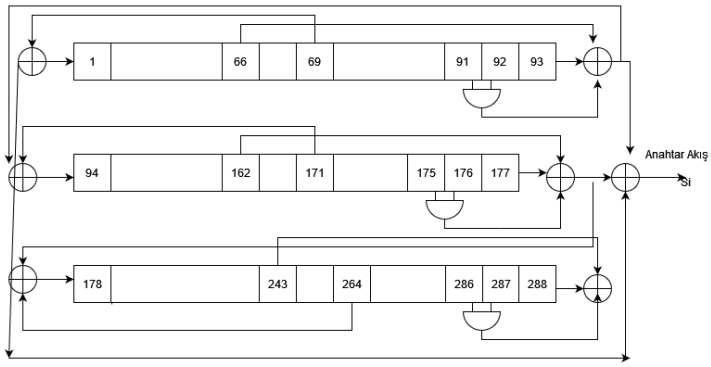
NIST test ortamı Linear Complexity (doğrusal karmaşıklık) testi LFSR yapıları ile üretilmiş bit dizilerinin bu doğrusallığını tespit edecek ve başarısız sonuç verecektir. Yani bir anahtar alan bit dizisi LFSR ile üretilmişse NIST testlerinden geçirildiğinde en azından bu testi geçmeyeceği beklenen bir durumdur. LFSR nin doğrusallik dezavatajını ortadan kaldırmak için literatürde eSTREAM yapıları önerilmiştir. eSTREAM yapılarından biri de trivium yapılarıdır.

2.2. Trivium

Trivium yapısı aslında donanım odaklı senkron akış şifreli bir yapıdır. LFSR gibi yapılar doğrusallik özellik gösterdiğinden ve akış şifresi oluşturmada kullanım uygunluğunun geliştirilmesi fikriyle ortaya çıkmıştır. Basit tasarımların üretilmesi kolay olsa da saldırılara karşı direnci düşük olur. LFSR'nin çeşitli kullanım versiyonları denenerek önerilen trivium yapıları gelişim sürecinde LFSR'lerden daha güvenli olduğu ortaya konulmuştur.

Trivium yapıları, 80 bitlik başlangıç değeriyle 2^{64} bite kadar anahtar akışı oluşturmak üzere tasarlanmış senkron bir yapıdır. Doğrusal LFSR'nin doğrusal olmayan hale dönüştürülmüş versiyonu olan trivium, toplamda 3 farklı LFSR'nin birbirine 3 and kapısı, 7 xor kapısı ile bağlanması sonucu elde edilir. Toplamda 288 kaydediciye sahiptir. Birinci LSFR 'de 93, ikincide 84, son olarak üçüncüde 111 kaydedici mevcuttur. Bu yapılar LFSR'lerden farklı olarak doğrusal değildirler. Trivium yapıları, aynı zamanda eş zamanlı akış şifrelemeye örnektir. Saldırganlar saldırılarını gerçekleştirirken anahtar akış bitleri arasındaki korelasyona bakmak en temel yöntemlerdendir. Bu yapıda açık bir şekilde Denklem 5'te verilen işlem tekrarlama adımından ötürü sonuçlar arası korelasyon mevcuttur. Trivium yapısı Şekil 3'teki gibidir.

$$z_i = S_{66} + S_{93} + S_{162} + S_{177} + S_{243} + S_{288} \quad (5)$$



Şekil 3. Genel Trivium Yapısı

2.3. Akış Şifresi Oluşturma

Trivium yapısı 288 bitik kaydediciden oluşur ($S_1 \ldots \ldots \ldots \ldots S_{288}$). Şekil 3' teki trivium yapısının sözde kodunu Algoritma 1'deki gibi tanımlamak mümkündür.

Algoritma 1: Trivium yapısı

1. $T_1 \leftarrow S_{66} + S_{93}$
2. $T_2 \leftarrow S_{162} + S_{177}$
3. $T_3 \leftarrow S_{243} + S_{288}$
4. $Z_i \leftarrow T_1 + T_2 + T_3$
5. $T_1 \leftarrow T_1 + S_{91}.S_{92} + S_{171}$
6. $T_2 \leftarrow T_2 + S_{175}.S_{176} + S_{264}$
7. $T_3 \leftarrow T_3 S_{286}.S_{287} + S_{69}$
8. $(S_1, S_2, S_3, \ldots, S_{93}) \leftarrow (T_3, S_1, S_2, \ldots, S_{92})$
9. $(S_{94}, S_{95}, S_{96}, \ldots, S_{177}) \leftarrow (T_1, S_{94}, S_{95}, \ldots, S_{176})$
10. $(S_{178}, S_{179}, S_{180}, \ldots, S_{177}) \leftarrow (T_2, S_{178}, S_{179}, \ldots, S_{287})$

3. Uygulama

3.1. İstatistiki Testler

Şifrelemede anahtar alan belirleme, sistemin temel olarak güvenliğinin sağlanmasındaki en önemli araçtır. Şifrelemede RSÜ yaygın olarak kullanılmaktadır. Bir RSÜ, donanımsal ya da yazılımsal olarak üretilmiş olsa da istatistiki olarak üretilen bu sayıların güvenli sayılabilmesi için asgari olarak literatürde kabul görmüş istatistiki testlerden geçmesi gerekmektedir. Literatürde kullanılan birçok istatistiki test mevcuttur. Bunlardan en yaygın olarak kullanılan testler NIST test ortamı, ENTU01, Diehard testleridir (İnce, 2021). NIST test ortamı 15 alt testten oluşmaktadır. Bu testi oluşturan alt testlerin tamamının test edilebilmesi için test edilecek rasgele bit stringinin bir milyon bit olması gerekmektedir (Bassham vd., 2010). NIST test ortamı Tablo 2'deki alt testlerden oluşmaktadır.

3.2. LFSR Yöntemi Kullanılarak Oluşturulan Bit Dizilerinin NIST Test Ortamı Analizi

Bir milyon bitlik ikili diziler LFSR ile üretilip, NIST testlerinden geçirildiği zaman, doğrusal karmaşıklık testinden geçmediği görülmüştür. LFSR ile üretilen bir milyon bitlik rasgele ikili dizinin NIST test ortamı testlerinden geçirildikten sonra sonuçları aşağıdaki gibi elde edilmiştir:

Tablo 2. LFSR mimarisi değiştirilerek üretilen bit dizisinin NIST testlerinden geçirilmesi

Testler	LFSR1 p-value	LFSR2 p-value
Frekans Testi	0.827603	0.712746
Blok Frekans Testi	0.515098	0.228180
Akış Testi	0.725929	0.037259
Bir Blok İçerisinde En Uzun Bitler Akış Testi	0.215875	0.033557
İkili matris derece Testi	0.536087	0.866045
Ayrık Fourier Dönüşüm Testi	0.00000	0.000000
Örtüşmeyen Şablon Eşleştirme Testi	0.225253	0.211102
Örtüşen Şablon Eşleştirme Testi	0.357972	0.067854
Maruer'in Evrensel İstatistik Testi	0.170459	0.770143
Doğrusal Karmaşıklık Testi	0.000000	0.761597
Seri Test	0.724430	0.037247
Yaklaşık Entropi Testi	0.448508	0.124680
Birikimli Toplamlar Testi	0.975575 0.842289	0.534422 0.866787
Rasgele Gezinimler Testi	0.340662 0.929416	0.771751 0.834199
Rasgele Gezinimler Değişken Testi	0.572479 0.484014	0.765025 0.623404

LFSR2, LFSR1 yapısında kullanılan mantık kapılarının değiştirilmesiyle elde edilmiş yeni LFSR yapısıdır. Kullanılan mantık kapıları değiştirildiği zaman doğrusallik testini geçebilen LFSR2, ayrık fourier testini geçememiştir. Bu durum sonrası geliştirilen trivium yapıları doğrusalliği bozmak için geliştirilmiş yapılardır. Dezavantaj olarak fazla işlem yükü gerektiriler ancak

LFSR'lerin tek başına kullanılmalarından çok daha güvenli sonuçlar üretirler.

3.3. Trivium Kullanılarak Oluşturulan Bit Dizilerinin NIST İstatistik Testlerinden Geçirilmesi

Şifreleme işlemleri yapılırken genel olarak mantıksal elemanladan ve matematiksel işlemlerden faydalanılır. Trivium yapıları 288 bitlik kaydediciye sahiptir. Bu yapılar e STREAM olarak adlandırılan şifreleme algoritmalarının irdelendiği ve fazlara ayrıldığı yapılardan donanım temelli bir yapı olup hem faz1 hem de faz 3 de kullanılan yapılardandır (Soto & Bassam, 2020). Trivium yapısı kullanılarak elde edilen bir milyon ikili bit dizisi NIST testlerinden geçirildiği zaman elde edilen sonuçlar Tablo 3' teki gibi değişmiştir

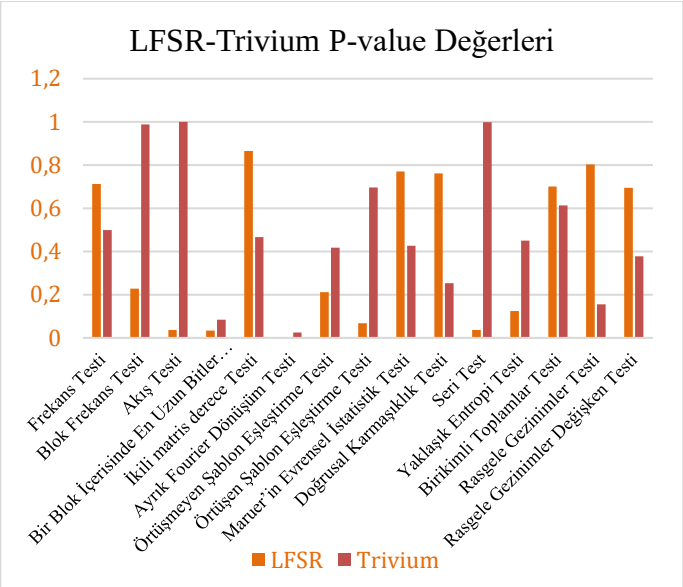
Tablo 3. Trivium yapıları kullanılarak üretilen bit dizisinin NIST testlerinden geçirilmesi

Testler	P-value Değerleri	Sonuç
Frekans Testi	0.499682	Başarılı
Blok Frekans Testi	0.988852	Başarılı
Akış Testi	0.999566	Başarılı
Bir Blok İçerisinde En Uzun Bitler Akış Testi	0.083540	Başarılı
İkili matris derece Testi	0.466400	Başarılı
Ayrık Fourier Dönüşüm Testi	0.025473	Başarılı
Örtüşmeyen Şablon Eşleştirme Testi	0.418305	Başarılı
Örtüşen Şablon Eşleştirme Testi	0.696056	Başarılı
Maruer'in Evrensel İstatistik Testi	0.426185	Başarılı
Doğrusal Karmaşıklık Testi	0.253527	Başarılı
Seri Test	0.999202	Başarılı
Yaklaşık Entropi Testi	0.449624	Başarılı
Birikimli Toplamlar Testi	0.764517 0.461968	Başarılı
Rasgele Gezinimler Testi	0.115473 0.193994	Başarılı
Rasgele Gezinimler Değişken Testi	0.225414 0.529061	Başarılı

Tablo 5 değerleri incelendiğinde trivium yapısı ile üretilen bir milyon bitlik ikili dizinin NIST testlerinin tamamından başarı ile geçtiği görülmüştür. Ayrıca LFSR mimarisine sahip triviumun doğrusallık özelliği göstermediği, böylece şifrelemede kullanılabileceği görülmüştür. Trivium üç farklı uzunlukta LFSR'nin mantık kapıları ile birbirine bağlanarak doğrusallik bozma yöntemidir denilebilir. Triviumun 288 bitlik ilk değerlerinin belirlenmesinde çeşitli yöntem ve algoritmalar mevcuttur. Bizim çalışmamızda Java SecureRandom kütüphanesinden elde edilen rasgele sayılar ile başlangıç değerleri belirlenmiştir. Litetürde RC4, LCG, A5/1, A5/3 vb. gibi algoritmalar kullanılarak geliştirilen trivium yapıları mevcuttur (Etetm & Kaya, 2020) (Cannieree & Preneel, 2008) (Deb & Bhuyon, 2021). Çalışmamızın diğer yöntemlerden fark iki ana başlıkta toplanabilir:

1. Yazılmsal olarak bir programlama dilinin içerisinde standart olarak sunulan bir kütüphane ile desteklenmiş olması. Böylede trivium yapılarını daha ulaşılabilir hale getirilmesi.
2. Trivium yapısının tamamen yazılım alana çekildiğinden ötürü daha hızlı kullanılabilmesi ve daha hızlı çalışması.

LFSR ve trivium için p-value değerlerinin kıyaslaması Şekil 5'deki gibidir.



Şekil 5. LFSR ve Trivium yapılarının NIST testleri P-value değerlerinin kıyaslanması

Tablo 3 trivium yapısının yazılımsal olarak da başarılı olduğunu gösterir. Ayrıca LFSR'deki doğrusallik testini geçemeyen ikili sayı dizisinin trivium yapısında başarılı olarak testi geçtiği, ayrık fourier testini de geçebilecek rastsallıkta sayılar üretilebildiği görülmüştür.

4. Bulgular

Trivium, LFSR yapılarındaki doğrusallik dezavantajlarını ortadan kaldıran bir yapıdır. Literatürde trivium başlangıç değerlerini üreten birçok algoritma ve yöntem mevcuttur. Bu çalışmada Java SecureRandom kütüphanesi kullanılarak elde edilmiş rasgele değerler başlangıç değerleri olarak kullanılarak bir milyon bitlik ikili veriler elde edilmiş ve rassallık analizi yapılmıştır. Sonuç olarak önerilen sistem NIST test ortamında başarı sağlamıştır. Java SecureRandom kütüphanesi kullanılırken SHA1PRNG algoritması seçilerek üretim gerçekleştirilmiştir. Bu çalışmada SecureRandom kütüphanesi kullanılarak yazılımsal uygulamalarda daha hızlı performans elde edilmesi hedeflenmiştir. Ayrıca yazılımsal olarak üretilen yapıların donanımsal olarak üretilen yapılara göre maliyetsiz olması da büyük avantajlardandır.

5. Tartışma

Bu çalışmada trivium yapılarındaki başlangıç değeri elde etme yöntemlerinden daha önce denenmediğini gözlemlediğimiz Java SecureRandom kütüphanesi denenmiştir. Sonuçlar başarılıdır ve işlem yükü azaltıldığından daha hızlıdır. Özellikle senkron şifreleme gereken alanlarda işlem yükünü azaltıp hızlı sonuçlar elde edileceği kanaatindeyiz.

6. Sonuçlar

NIST 800-22 Rev 1a. istatistiki testlerini başarılı şekilde geçen yazılımsal trivium yapıları, donanımsal trivium yapılarına göre tercih sebebi olabilir. Maliyetsiz ve hızlı olması özellikleri ile tercih sebebi olarak kullanılabilir.

Teşekkür

Bu çalışma İnönü Üniversitesi Bilimsel Araştırma Projeleri Daire Başkanlığı'nın (İnönü BAP) FBG-2020- 2143 numaralı projesi ile desteklenmiştir. Değerli destekleri için İnönü Üniversitesi İnönü BAP'a teşekkürlerimi sunarım.

Kaynakça

Özkaynak F. (2018). *Brief review on application of nonlinear dynamics in image encryption*, Nonlinear Dynamics, 1573-269X, https://doi.org/10.1007/s11071-018-4056-x

Arrachid K. Et. Al (2014). Arrachid K, Mohamed Mejri M., Sadio T.E., AVTAC: A Framework for Automatic Auditing of Access Control in Windows and Linux Systems, New Trends in Software Methodologies, Tools and Techniques page. 672-691 doi: 10.3233/978-1-61499-434-3-672

Beletsky A. (2021). *Generalized Galois-Fibonacci Matrix Generators Pseudo-Random Sequences,* Computer Network and Information Security, 6,57-69, DOI: 10.5815/ijcnis.2021.06.05

Manikya D.M. (2021). D. M. Manikya, M. Jagruthi, R. Anjum and A. K. K, "Design of Test Compression for Multiple Scan Chains Circuits," 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), 2021, pp. 1-5, doi: 10.1109/ICSCAN53069.2021.9526387.

S. Islam and I. U. Haq (2016). "Cube attack on Trivium and A5/1 stream ciphers," 2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 409-415, doi: 10.1109/IBCAST.2016.7429911.

Ravichandran, D., Rajagopalan, S., Upadhyay, H. N., Rayappan, J. B. B., & Amirtharajan, R. (2018). Encrypted Biography of Biomedical Image - a Pentalayer Cryptosystem on FPGA. Journal of Signal Processing Systems, 91(5), 475–501. https://doi.org/10.1007/s11265-018-1337-z

Mondal, B., Sinha, N., & Mandal, T. (2016). A secure image encryption algorithm using LFSR and RC4 key stream generator. In Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics, ICACNI 2015 (pp. 227-237). (Smart Innovation, Systems and Technologies; Vol. 43). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-81-322-2538-6_24

Goresky M. and Klapper A. M. (2002). "Fibonacci and Galois representations of feedback-with-carry shift registers," in IEEE Transactions on Information Theory, vol. 48, no. 11, pp. 2826-2836, Nov. 2002, doi: 10.1109/TIT.2002.804048.

Chakraborty A., Mazumdar B., Mukhopadhyay D. (2014) *Fibonacci LFSR vs. Galois LFSR: Which is More Vulnerable to Power Attacks?*. In: Chakraborty R.S., Matyas V., Schaumont P. (eds) Security, Privacy, and Applied Cryptography Engineering. SPACE 2014. Lecture Notes in Computer Science, vol 8804. Springer, Cham. https://doi.org/10.1007/978-3-319-12060-7_2

P. Zode, R. Deshmukh (2019). "FPGA Based Novel True Random Number Generator using LFSR with Dynamic Seed," 2019 IEEE 16th India Council International Conference (INDICON), 2019, pp. 1-3, doi: 10.1109/INDICON47234.2019.9029049.

Garipcan, A. M. , Erdem, E. & Tuncer, T. (2017). Donanım Tabanlı Trivium Akış Şifreleme Algoritmasının FPGA Ortamında Gerçekleştirilmesi. Fırat Üniversitesi Mühendislik Bilimleri Dergisi , 29 (2) , 119-130

Kaya, T. (2020). "Memristor and Trivium-based true random number generator", Physica A Statistical Mechanics and its Applications, vol. 542, 2020. doi:10.1016/j.physa.2019.124071.

Etem T., Kaya T. (2020). *Görüntü Şifreleme için Trivium Doğrusal Eşlenik Üreteci Tabanlı Bit Üretimi* , Fırat Üniversitesi Mühendislik Bilimleri Dergisi. 32(1), 287-294

Berlekamp, Elwyn R. (2015). Algebraic Coding Theory - Revised Edition, World Scientific Publishing Co., Inc. USA, ISBN: 9789814635899

Reeds, J. A.; Sloane, N. J. A. (1985), *"Shift-Register Synthesis (Modulo n)*", SIAM Journal on Computing, 14 (3): 505–513, CiteSeerX 10.1.1.48.4652, doi:10.1137/0214038

İnce, K. (2021). *Security Analysis of Java SecureRandom Library* , Avrupa Bilim ve Teknoloji Dergisi, Ejosat Special Issue 2021 (Araconf), 157-160. DOI: 10.31590/ejosat.900956.

Bassham III, L. E., Rukhin, A. L., Soto, J., Nechvatal, J. R., Smid, M. E., Barker, E. B., ... & Vo, S. (2010). Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications.

https://www.ecrypt.eu.org/stream/ (Link erişim:23.01.2022 14:17)

Soto, J., & Bassham, L. (2000). *Randomness testing of the advanced encryption standard finalist candidates*. Booz-Allen And Hamilton Inc Mclean Va.

Cannière, C. D., & Preneel, B. (2008). Trivium. In *New stream cipher designs* (pp. 244-266). Springer, Berlin, Heidelberg.

Deb, S., Bhuyan, B. (2021). Chaos-based medical image encryption scheme using special nonlinear filtering function based LFSR. Multimed Tools Appl 80, 19803–19826. https://doi.org/10.1007/s11042-020-10308-7