Electronic Letters on Science & Engineering 16(2) (2020)
Available online at dergipark.org.tr/else

DESIGN AND IMPLEMENTATION OF A CRYPTOGRAPHIC METHOD ON MATHEMATICAL EXPRESSIONS

Bahar MİLANİ[1*], Muhammed MİLANİ[2], Özlem ORHAN[3]

[1]Department of Computer Engineering, Bandirma Onyedi Eylul University, Balikesir- Turkey
Email: bmilani@bandirma.edu.tr ORCID: 0000-0002-5295-4215
[2]Department of Computer Engineering, Bandirma Onyedi Eylul University, Balikesir- Turkey
Email: mmilani@bandirma.edu.tr ORCID: 0000-0003-2450-0280
[3]Department of Engineering Science, Bandirma Onyedi Eylul University, Balikesir- Turkey
Email: oorhan@bandirma.edu.tr ORCID: 0000-0003-0058-0431

Abstract: Data hiding has been a very interesting issue recently and various methods have been developed to deal with this issue. With the development of data hiding techniques, suitable steganalysis design to detect hidden data appears in studies. However, an approach to common valves followed by steganalyses to a new type of valve may be less questionable. In this study, a method for hiding messages in mathematical expressions is proposed. In the proposed method, the mathematical expression is used as a cover to convey the message with a secure text. In line with the study, we propose a methodology for problems that can be included in the computer algebra system. Its method involves developing a grammar-based interpreter who can recognize the syntax and meaning of mathematical expressions and turn it into an abstract syntax tree (AST).

Keywords: Data hiding, Stochastic grammar, Math expression, Abstract Syntax Tree, Parser, javaCC

MATEMATİKSEL İFADELER ÜZERİNE BİR KRİPTOGRAFİK YÖNTEMİN TASARIMI VE GERÇEKLEMESİ

Özet: Veri gizleme son zamanlarda çok dikkat çekici konulardan biri olmuştur ve bu konu ile ilgili çeşitli yöntemler geliştirilmiştir. Veri gizleme tekniklerinin gelişmesi ile birlikte gizli verileri saptamak için uygun steganaliz tasarlama çalışmalarda gözükmektedir. Ancak, steganalizler tarafından izlenilen genel kapaklar yeni tipte bir kapağa getirilen bir yaklaşım daha az şüpheli olabilir. Bu çalışmada, matematiksel ifadeler içerisinde mesaj gizleme için bir yöntem önerilmiştir. Önerilen yöntemde matematiksel ifade, mesajı güvenli bir metinle birlikte iletmek için bir kapak olarak kullanılmaktadır. Çalışma doğrultusunda, bilgisayar cebir sistemine dahil edilebilecek problemler için bir metodoloji öneriyoruz. Metodoloji matematiksel ifadelerin sözdizimini ve anlamını açıklayan biçimsel bir gramerin geliştirilmesini ve belirli bir matematiksel ifadeyi Özet Sözdizimi Ağacına (AST) dönüştürmeyi kapsar.

Anahtar Kelimeler: Veri gizleme, Stokastik gramer, Matematiksel ifade, Soyut Sözdizimi Ağacı, Parser, javaCC

Reference to this paper should be made as follows:
Milani B. ,Milani M. ,Orhan Ö. , 'Design And Implementatıon Of A Cryptographic Method On Mathematical Expressions', Elec Lett Sci Eng , vol. 16(2), (2020), 121-129.

1. Giriş

Veri güvenliği ve bilgi alışverişinin bütünlüğü bilgisayar ağları ve internetin geliştirilmesi ve sürekli artan kullanımıyla yetkili ve yetkisiz kişilerin aktarılan verilere kolay erişiminden dolayı araştırmacıların karşılaştığı önemli bir sorun haline gelmiştir. Veri paketleri internet ortamında birkaç ara katmandan geçerek hedeflerine ulaştıklarından, üçüncü şahıs bunları yakalayabilirler ve değerlendirip kullanabilirler. Bu nedenle, internet üzerinden güvenliği sağlamanın temel yapı taşı veri paketlerinin korunmasıyla ilişkilidir.
Yeni uygulamaların, özellikle ticari, askeri uygulamalar ve video konferansların geliştirilmesi gibi dijital ortamların aktarımı için hızlı ve güvenli sistemlerin kullanılması gereklidir. Verilerin şifrelenmesi, geleneksel veya modern kriptografi, DNA tabanlı ve kaos tabanlı teknikler vb.

[*] *Milani B.; Tel.:* +905317203041, bmilani@bandirma.edu.tr
ISSN 1305-8614

gruplarda sınıflandırılan çok farklı şifreleme algoritmaları ve teknikleri ile gerçekleştirilir. Bazı teknikler Kuantum Fourier dönüşümleri ve eliptik eğri gibi matematiksel kavramlara dayanır [1,2]. Bazı uygulamada nadiren kullanılmış olan 1979'da Shamir tarafından önerilen gizli paylaşım kavramlarına dayanmaktadır [3]. Sıkıştırma tabanlı yöntemler, genellikle, SCAN dili [4] gibi sıkıştırmaya veya vektör nicelleştirmeye dayalı tekniklere dayanan dijital görüntüler için önerilen başka bir şifreleme yöntemidir [5].

Kriptografi gizli iletişim için yaygın olarak kullanılan bir tekniktir ve çeşitli şifreleme ve şifre çözme yöntemleri ile uygulanır [6]. Ancak, şifreli metin dikkat çekebileceğinden, şifreleme sadece güvenli bilgi aktarımı için yeterli değildir. Hatta son zamanlarda bazı optimizasyon yöntemlerde [7,8] bilgi gizleme işlemlerini daha güvenilir şekilde yapmak için kullanılmaktadır. Steganografi de bu problemi çözmek için en ufak bir şüphe uyandıran stego-metinleri kullanarak güvenli bir şekilde bilgi göndermenin yollarından biridir. Tipik olarak, bu sistemlerde, metinler çeşitli kapaklarda mesaj olarak gizlenebilir. Girilen mesaj sadece kapaklarda küçük değişikliklere neden olmalıdır. Genel olarak, kapaklar dört kategoriye ayrılabilir: metin, resim, video ve ses [9]. Bu dört kategoride yer almayan, yürütülebilir dosyalar gibi başka kapaklar da olabilir [10].

Bu kategoriler arasında, görüntü bazlı steganografi daha yaygın olarak kullanılmış ve görüntünün daha az tanınabilen kısımlarına veri dahil edilmiştir [11]. Video, mesajı eklemek için çok sayıda işlem gerektiren bir kapak olarak da bilinir [12]. Ancak, büyük hacimli mesajlar ekleme yeteneği nedeniyle, dikkat çekmişlerdir [13]. Sesli bir mesaj eklemek de mümkündür. Genellikle, insan kulağı 20HZ'den 20KHZ'ye kadar olan sesleri anlayabilir. Bu nedenle, bazı ses tipi kapaklarda, bu özellik tanınmayan bir şekilde veri eklemek için kullanılır [14]. Bazı steganografi yöntemlerinde de metinler kapak olarak kullanılmıştır [8]. Metin tabanlı yöntemlerin sayısı diğer yöntemlerden daha az olsa da bu yöntemlerde büyük miktarda metin veri türü onları çekici hale getirmiştir.

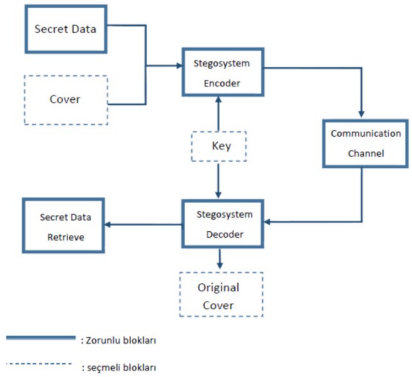
Steganografinin aksine, steganaliz [15] yerleştirilen mesajın yetkisiz olarak çıkarılması bilimidir. Steganaliz ve steganografinin saklambaç oyununa benzer olduğu söylenebilir [16]. Steganografi için çeşitli yöntemlerin geliştirilmesi ile Steganaliz için de çeşitli yöntemler geliştirilmiştir [17]. Açıkçası, yeni yöntemler güvenli bilgi aktarımında daha güvenilirdir, çünkü steganaliz henüz yeni kapaklara odaklanmamıştır.

Bu çalışmada, matematiksel ifadelerde veri gizleme olayının gerçekleştirilmesine bir yöntem önerilmiştir. Matematiksel ifadelerin özellikleri kullanarak daha gerçek gibi görünen veriler üretmek için çaba sarf edilebilir [18].

2. Steganografi

Bilginin gizlemesinin en önemli kısmı "Kapalı yazı" anlamına gelen Yunanca sözcüklerden türetilmiş steganografidir. Steganografi ile kriptografi arasındaki fark, mesajın sadece şifrelenmesi yerine mesajın varlığının gizlemesidir.

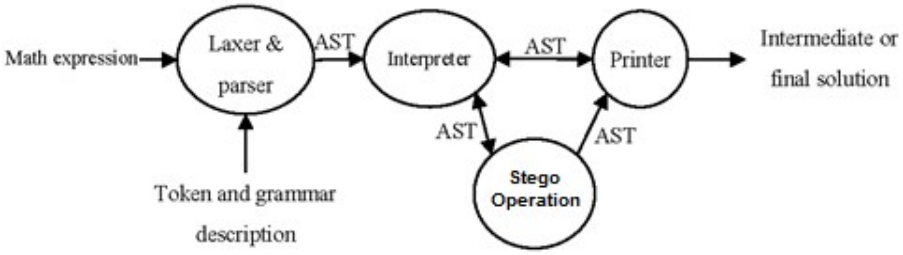
Steganografinin temel modelinde bir Kapak (Cover) seçilerek (veya üretilerek), gizli mesaj kapak içerisine özel bir algoritma ile birlikte bir stego-nesne oluşturulur. Diğer tarafta, kodlama sürecinde yapılanları tersine çevirerek, gizli mesaj Stego nesnesinden alabilir. Bu arada başka bir güvenlik seviyesi ekleyebilmek için, gizli mesaj gizlenmeden önce şifrelenebilir. Genel Steganografi Modeli Şekil 1'de gösterilmektedir.



Şekil 1. Steganografinin Genel Modeli

3. Matematiksel İfadelerle Çalışma

Bu çalışmada kriptoloji biliminde olan veri gizleme işlemi ele alınmıştır. Veri gizleme işlemi farklı medyalar üzerinden yapılması amacı ile matematiksel ifadeler için literatürde boşluk hissedildiğinden dolayı, bu konu matematiksel ifadeler üzerinden yapılmaktadır. Önerilen yöntemde var olan bir ifadeyi ağaç üzerine taşıyıp, daha sonra istenilen mesaj ifadeye gömülmüştür. Dolayısıyla, çalışmada matematiksel ifadeler üzerine gereken metotlar geliştirilmiştir. Şekil 2 çalışmanın genel yapısını göstermektedir.



Şekil 2. Önerilen yöntemin genel yapısı

3.1. Matematiksel İfadelerin Değerlendirilmesi

Bu çalışmada, çok taraflı bir sistem şeklinde tasarlanan ve matematiksel ifadelerin üzerinde çalışabilen sistem için bir metodoloji önerilmiştir. Sistem girdi verilerini sözdizimi olarak ayrıştırması (parse) için genişletilmiş bir gramer kullanır ve daha sonra belirli bir matematiksel ifadeyi kolayca işlemek için uygun modeli sağlayan bir ağaç veri yapısına dönüştürür. Ağacın her düğümü, giriş ifadesinin bir sembolünü veya terimini ve sembollerin operatör, sayı vb. olabileceği diğer semboller veya terimlerle ilişkilerini içerir. Ağaç gösterimi ayrıca bir düğüm ve çocukları üzerinde tekrarlanan işlemlerin uygulanmasına yardımcı olur. Örneğin, bir düğümün değerlendirilmesi öncelikle o düğümün çocuklarını değerlendirmeyi gerektirir. Belirli bir düğüm kombinasyonunun yanı sıra, diğer belirli işlemlerin uygulanması gereken bir model tanımlayabilir.

3.2. BNF Gramer Tanımı

BNF notasyonu, özellikle programlama dilleri olmak üzere, biçimsel diller için bağlam-içermeyen gramerler tanımlamak için kullanılır. Basit notasyonlara ve özyinelemeli yapılara sahiptir. YACC [19], LEX ve JavaCC [20, 21] gibi birçok derleyici oluşturma aracı bir kaynak dilin BNF benzeri tanımını kullanır. Matematiksel ifadeler *toplama* veya *çıkarma*, *sin* veya *cos*

gibi genel işlevler veya *integral* gibi özel semboller gibi işlemleri içerebilir. Belirli bir matematiksel ifade için geliştirilmiş bir gramer verildiğinde, tüm operatörler, fonksiyonlar, semboller, değişkenler ve sayılar terminal setinin üyeleri olacaktır. Non-terminal küme gramerin üretim kurallarına göre belirlenir. Tasarlanan gramer, bir operatör ve onun işlenenleriyle (operands) veya argümanlarla bir matematiksel işlevle aritmetik ifadeler üretmelidir. Bir işlenenin (operand) veya argümanın kendisi bir sayı, bir değişken veya başka bir matematiksel ifade olabilir. Gramerin üretim kuralları özyinelemeli olabilir, çünkü "expression" türünde işlenenler, kısa bir süre "<E>" olarak adlandırılır. Ek olarak, ondalık (veya tamsayı) sayılar istenilen basamak sayısına kadar oluşturulabilir. Her farklı matematiksel ifadesi farklı gramerin kullanılmasını gerektirir. Liste 1'de gösterilen [20] 'te sunulan birtakım değişiklikler ile uzatılmış bir BNF gramer geliştirilmiştir.

Liste 1. Matematiksel ifade için Genişletilmiş-BNF grameri

```
<E>::= <E> <O> <E>  |(<E>)
          |<F>(<E>)    |<V>      |<N>
<O>::='+' | '-' | '*' | '/' | '^'
<F>::= 'Abs' | 'Sqrt' | 'Power' | 'Sin' |'Sec' | 'Tan'
<V>::='x'
<N>::= '-' ? <D> + ('.' <D> +)?
<D>::=['0'-'9']
```

Liste 1'de verilen gramer, basit matematiksel ifadede yer alabilen beş operatör ve altı fonksiyona sahiptir. Ancak, diğer bazı operatörlerin, fonksiyonların ve simgelerin eklenmesiyle değiştirilebilir. Gramer tanımı operatörlerin önceliğini göz önünde bulundurmadığı için, ifadeye matematikte yapılandan farklı bir anlam verilecek ve bu nedenle operasyonların doğru düzeninde değerlendirilmeyecektir. Bu çalışmada JavaCC gibi derleyici-derleyici aracı kullanmak için grameri LL (1) şeklinde yazılması gerekiyor. Liste 2, operatörün önceliği ve ilişkilendirmesi dikkate alınarak değiştirilen eşdeğer LL (1) gramerini göstermektedir.

Liste 2. Matematiksel ifadeler için LL (1) grameri

```
<E>⟶<U> <T> [("+"|"-") <T> ]*
<U>⟶("+"|"-")?
<T>⟶<P> [("*"|"/")<P>]*
<P>⟶<E'> ("^"<P>)?
<E'>⟶<F> (<E>) | <N> | "x"
<F>⟶"Abs" | "Sqrt" | "Power" | "Sin" | "Sec" | "Tan"
<N>⟶"-"? <D> + ("."<D>+)?
<D>⟶["0"-"9"]
```

Liste 2'deki gramer, normal LL (1) gramerin optimize edilmiş bir versiyonu olup, tüm ekstra terminaller, orijinal olmayan terminale geri birleştirilir.

3.3. Sözdizimi Sınıflarının Tanımı

Sözdizimi bir dilin yazma kurallarıdır. Matematiksel ifadeler diğer dillerin yanı sıra, bileşenlerinin sınırlı bir kombinasyonu olduğundan belirli bir sözdizimine sahiptir. Matematiksel ifadelerinde sayılar, değişkenler, işlemler, fonksiyonlar, gruplama sembolleri ve diğer sözdizimi semboller kullanılır, her bileşen tanımlanması gereken belirli bir yapıya sahiptir. Tablo 1, bazı sözdizimi sınıflarını ve çalışmalarımıza uyarlanmış niteliklerini göstermektedir.

Tablo 1'de olduğu gibi, diğer operatörler, fonksiyonlar veya semboller için ek sözdizimi sınıfları tanımlanabilir. Nesne yönelimli programlamada, her kural genellikle bir sınıf tarafından tanımlanır, daha sonra ifadeleri değerlendirmek için kullanılır.

Tablo 1. Bazı matematiksel bileşenler için sözdizimi sınıfları

	Sözdizimi	**Sınıf**	**Özellikler**
+	E + E	Sum	exp1, exp2
-	E - E	Sub	exp1, exp2
*	E * E	Mul	exp1, exp2
Abs	Abs(E)	Abs	exp
Sqrt	Sqrt(E)	Sqr	exp

3.4. Soyut Sözdizimi Ağacı Oluşturma (SSA-Abstract Syntax Tree-AST)

Bir parser (sözdizimi analizcisi), girdi verilerini bir dizi oluşturulmuş token'e dayanarak üretkenlik açısından analiz eder. Token dizisinin oluşturulup oluşturulmadığını kontrol eder ve gramer kurallarına göre inceler. Bu nedenle, token'lerin doğrulanması için bir mekanizmaya ve dil kurallarına göre onların görünüş sırasına ihtiyacımız var. Elbette, analiz sürecinde, sistem değerlendirmeden önce kabul edilebilir girdi verileri için anlamsal ya da semantik bir analiz yapmalıdır. Ancak, parser matematiksel ifadelerin yapıları nedeniyle bu verilerin değerlendirme yeteneğini garanti edebilir. Parser elle yazılmış fonksiyonlar veya ayrıştırıcı üretici araçları kullanılarak tasarlanabilir. Parser oluşturma aracının kullanımı, bu parser'in açıklama koşullarına dayalı olarak istenen gramerin geliştirilmesini içerir. Örneğin, Yacc için bir gramer LR'de ve JavaCC için LL biçiminde geliştirilmelidir. Bu çalışmada girdilerin parser üretimi ve geçerlilik testi için JavaCC kullanıyoruz. JavaCC bildirimi içindeki işlevlerin veya metotların adı, Liste 2'deki gramer içindeki non-terminal kümeye göre belirlenir. Genel olarak, bir gramerde her non-terminal için bir metot tanımlanmalıdır. Bazı durumlarda, non-terminal birkaç sembolü birleştirmek yararlı olabilir, sadece onlar için bir metot tanımlar. Örneğin, Liste 1'deki non-terminal <E> düşünün. Bu non-terminal için, <E> ⟶ <T> <E'> olarak bir kural var. Bu nedenle, parser'de bir metot tanımlanabilir. Ancak ilgili non-terminal <E'> için parse ağacı üretiminde bazı zorluklar ortaya çıkabilir. Bu gibi durumlar için, bu terminaller birleştirilebilir, bir non-terminal ile sonuçlanabilir ve parse üreticisi aracında sadece bir metot ile temsil edilebilir. Tablo 2 birleştirme işleminin tipik örneklerini göstermektedir.

Tablo 2. Birleştirme işleminin tipik örnekleri

Kurallar	Birleşik Kurallar
$< E > \longrightarrow < T > < E' >$ $< E' > \longrightarrow$("+" | "-") $< T > < E' >$ $< E' > \longrightarrow \lambda$	$< E > \longrightarrow < T >$ { ("+" | "-") $< T >$ }*
$< T > \longrightarrow < U > < T' >$ $< T' > \longrightarrow$("*" | "/") $< U > < T' >$ $< T' > \longrightarrow \lambda$	$< T > \longrightarrow < U >${ ("*" | "/") $< U >$}*

Matematiksel ifadeler üzerinde bazı işlemleri yapabilmek için, sadece doğruluğu değil, aynı zamanda istenen veri yapısını da oluşturmak gereklidir. Matematiksel ifadeler için yararlı yapılardan biri, sözdizimi ağacıdır. Bir sözdizimi ağacı, hiyerarşik bir yapıda birbirine bağlanan birçok düğümden oluşur. Her düğüm bir sözdizimi sınıfından türetilir ve kaynak verilerinin bir ifadesini veya ifadesini temsil eden bir nesne tarafından oluşturulur. Sözdizimi ağacı genellikle bir süper sınıf türüyle oluşturulur. Aynı süper sınıftan miras alan alt sınıflar, bir nesne ağacının

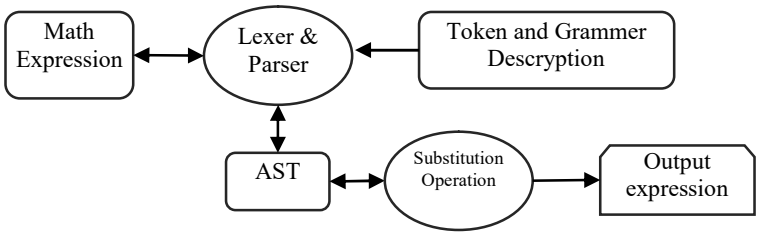
düğümlerini oluşturmak için kullanılabilir, ancak ağaç üzerindeki bu düğümler, süper sınıfın referansıyla temsil edilebilir.

4. Önerilen Veri Gizleme Yöntemi

Bilgi saklama konusuna son zamanlarda çok dikkat çekilmiştir. Steganografi için çeşitli yöntemler geliştirilmiştir ve aynı zamanda gizli verilerden kuşkulanmak ve bulunması için uygun steganaliz tasarlanmıştır. Ancak, yeni tipte bir kapağa getirilen bir yaklaşım daha az şüpheli olabilir. Bu bölümde, bir mesajı matematiksel bir ifadeye dönüştürebilecek bir yöntem önerilmektedir. Oluşturulan matematiksel ifade, mesajı güvenli bir metinle birlikte iletmek için bir kapak olarak kullanılabilir.

4.1. Yer Değiştirme Yöntemi

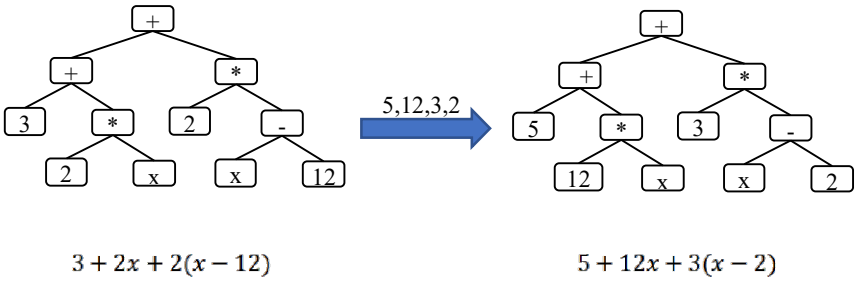
Bu yaklaşımda mesaj verilerini bir ifadenin sayı değerleri veya işlevleri içine gömebiliriz. Gömme işlemleri çeşitli metin yöntemlere benzer yöntemler ile gerçekleşebilir. Metin ile matematiksel ifadeler arasındaki fark aslında metinlerin karakterleri ardışık ve matematiksel ifadelerin özel yapıları olmasıdır. Dolayısıyla matematiksel ifadelerin üzerine işlem yapılmadan önce, AST ağacına taşımamız gerekiyor. Şekil 3 matematiksel ifadelerin üzerine veri gizleme metodolojisini göstermektedir.



Şekil 3. Yer Değiştirme Yöntemin genel modeli

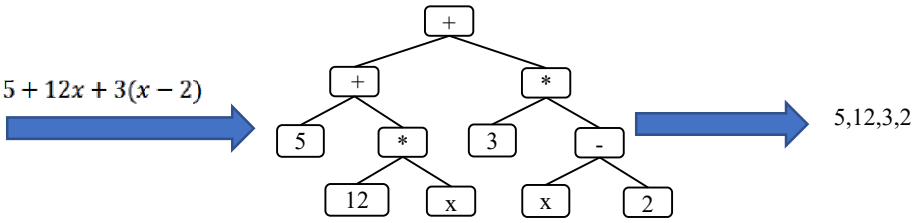
Şekil 3 de göründüğü gibi matematiksel ifadeler belli bir kurallar üzerinden AST ağacına dönüşü lup, üzerinde farklı farklı yer değiştirme yöntemleri uygulanır. Aşağıda birkaç örnek verilmiştir:

Örnek 1: Bu yöntemde matematiksel ifade AST ağaca taşındıktan sonra, AST ağacı pre-order, in-order veya post-order şeklinde okunur ve rastladığı her sayı, mesaj verileri ile yer değiştirilir. Şekil 4 bu yöntemi bir örnek üzerinden göstermektedir.



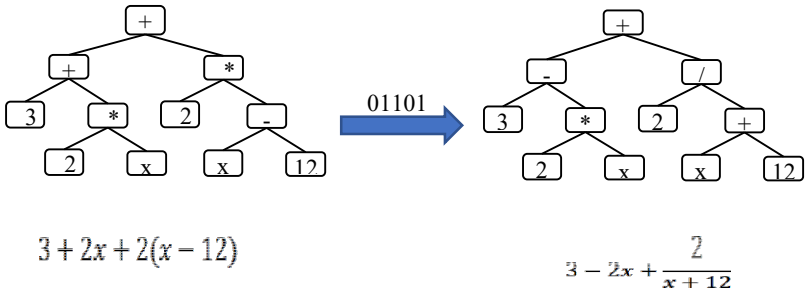
Şekil 4. In-order trace ile yer değiştirme yönteminden örnek

Şekil 4'deki örnekte mesaj verileri ile ağaçtaki sayılar eşit olduğundan herhangi bir sıkıntı yaşanmadı. Eğer ağaç sayıları mesaj verilerinden daha fazla olsaydı, ağacın geriye kalan sayıları *extract* aşamasında mesaj verilerine eklenebilir. Bu sıkıntıyı gidermek için mesaj verilerin sonuna -1 (veya herhangi eksi değer) eklenir. *Extract* aşamasında bu veri okunduktan sonra mesaj verilerin sonu olduğu anlaşılır. Şekil 5, önceki örneğin extraction (veri çıkarma) işlemini göstermektedir.



Şekil 5. In-order trace ile yer değiştirme yöntemindeki örneğin extraction işlemleri

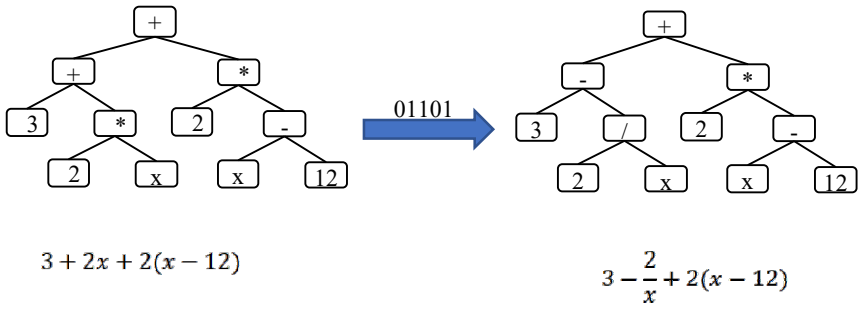
Örnek 2: Bu yöntemde AST üzerinde olan ifade belli bir kuralla taranır, örneğin pre-order veya satır-satır soldan sağa) ve mesaj bitlerine göre 0 ise rastladığı işlev değişmez ve 1 ise + işlevi – ile ve * işlevi / ile yer değiştirilir. Şekil 6 bu yöntemi bir örnek üzerinden anlatmaktadır.



Şekil 6. İşlev değiştirerek mesaj gömmeye örnek

Şekil 6'daki örnekte AST satırları soldan sağa sıra ile taranmıştır. Bu yöntem ile gömülen mesajları çıkarması için karşı taraf da orijinal ifadenin bulunması gerekiyor. *Extraction* işlemi yaparken, her iki ifadenin AST ağacı üretilir, değişilen işlevler 1, değişmeyen işlevler ise 0 olarak kayıt edilir. Bitlerin sırası ise TRACE yöntemine tabi tutulur. Karşı taraf de ifadenin olma gerekçesi ortadan kalksın diye aşağıdaki yöntemi kullanabiliriz.

Örnek 3: Bu yöntem bir önceki yönteme benzer, ancak burada + ve * işlevi 0,- ve / işlevi ise 1 anlamında kullanılır. Bu yöntemi kullanarak karşı taraf de orijinal ifadenin olması gerekmiyor. Yalnız işlevlere karşı gelen bitleri tarama yöntemine göre sıralanırsa mesaj elde edilebilir. Şekil 7 İşlev değiştirme yöntemini bir örnek üzerinden göstermektedir.



Şekil 7. İşlev değiştirerek mesaj gömmeye başka örnek

5. BULGULAR VE İRDELEME

Bu yöntem genel olarak matematiksel ifadeleri uygun biçime dönüştürüp yerine koyma yöntemleri uygulanabilecek vaziyete getirme amacıyla yapılmıştır. Görüntü veya metin kapakları üzerinde işlenen yöntemler AST ağacı üzerine taşınabilir. Bu çalışmada birkaç örnek verilerek bunun açıklaması yapıldı şeklinde düzeltilmelidir.
Genelde matematiksel ifadeler özel yapıya sahip olduğundan herhangi sıra söz konusu olmuyor. Ancak önerilen metodoloji ile ve belli bir dolaşma yöntemi kullanarak sıralı biçimde olaya bakılabilir.
Bu çalışmada, matematiksel problemlerin kullanılması için gramer tabanlı bir metodoloji önerilmiştir. Temel yapı matematik problem çözme sistemlerinin geliştirilmesi için kullanılabilir. İfadelerin biçimi, kesin bir yön sergileyen LL (1) gramerlerle temsil edilir. JavaCC aracını kullanarak, her zaman matematiksel ifadeler için eşsiz (*unique*) bir türevi izleyen LL (1) parser'leri geliştiririz. Bu parser'lerin uygulanması, gramer kurallarının ilgili metotlara eşlenmesini içerdiğinden nispeten kolaydır. Hiyerarşik bir yapıya sahip Soyut Sözdizimi Ağacı (SSA) kullanarak bir giriş ifadesi modellenmiştir. Ağaç, operatörlerin önceliğini ve birliği temsil eder ve böylece düğümleri arasındaki anlamsal ilişkiyi kurar. Uygulanan dönüşümleri ve ortaya çıkan ifadeyi yazdırmak için çeşitli belgeler veya raporlar üretilebilir. AST içeriğinin yazdırıldığı bir belge formatlama sistemini kullanmayı tercih ediyoruz. AST'nin değerlendirmesi oldukça basittir. Ağaçta değişkenler ve sayılar içerir, bu nedenle *x*, *a* ve *b* gibi değişkenleri bazı değerlere başlatılmasıyla, sayısal sonuç hesaplanabilir.

Referanslar

[1] Gong, L.-H., et al., Single Channel Quantum Color Image Encryption Algorithm Based on HSI Model and Quantum Fourier Transform. International Journal of Theoretical Physics, 2018. 57(1): p. 59-73.
[2] Wu, J., X. Liao, and B. Yang, Color image encryption based on chaotic systems and elliptic curve ElGamal scheme. Signal Processing, 2017. 141: p. 109-124.
[3] Wu, X., J. Weng, and W. Yan, Adopting secret sharing for reversible data hiding in encrypted images. Signal Processing, 2018. 143: p. 269-281.
[4] Maniccam, S. and N.G. Bourbakis, Lossless image compression and encryption using SCAN. Pattern Recognition, 2001. 34(6): p. 1229-1245.
[5] Yan, B. and S. Bai. Design of image confusion-diffusion cryptosystem based on vector quantization and cross chaotic map. in Image, Vision and Computing (ICIVC), 2017 2nd International Conference on. 2017. IEEE.
[6] Gupta, B., D.P. Agrawal, and S. Yamaguchi, Handbook of research on modern cryptographic solutions for computer and cyber security. 2016: IGI Global .

[7] Rahkar Farshi, T.: Battle royale optimization algorithm. Neural Comput. Appl. 1–19 (2020)
[8] Orujpour, M., Feizi-Derakhshi, M. R., & Rahkar-Farshi, T. (2019). Multi-modal forest optimization algorithm. Neural Computing and Applications, 1-15.
[9] Bhattacharyya, S., I. Banerjee, and G. Sanyal, A novel approach of secure text based steganography model using word mapping method (WMM). International Journal of Computer and Information Engineering, 2010. 4(2): p. 96-103.
[10] Lockwood, R. and K. Curran, Text based steganography. International Journal of Information Privacy, Security and Integrity, 2017. 3(2): p. 134-153.
[11] Li, J., et al., Color image watermarking scheme based on quaternion Hadamard transform and Schur decomposition. Multimedia Tools and Applications, 2018. 77(4): p. 4545-4561.
[12] Balaji, R. and G. Naveen. Secure data transmission using video Steganography. in Electro/Information Technology (EIT), 2011 IEEE International Conference on. 2011. IEEE.
[13] Nikam, G., et al., A Survey of Video Steganography Techniques. Journal of Network Communications and Emerging Technologies (JNCET) www. jncet. org, 2017. 7(5).
[14] Mishra, S., et al., Audio Steganography Techniques: A Survey, in Advances in Computer and Computational Sciences. 2018, Springer. p. 581-589.
[15] Johnson, N.F. and S. Jajodia. Steganalysis of images created using current steganography software. in International Workshop on Information Hiding. 1998. Springer .
[16] Provos, N. and P. Honeyman, Hide and seek: An introduction to steganography. IEEE security & privacy, 2003. 99(3): p. 32-44.
[17] Czaplewski, B., Current trends in the field of steganalysis and guidelines for constructions of new steganalysis schemes. Przegląd Telekomunikacyjny+ Wiadomości Telekomunikacyjne, 2017: p. 1121-1125.
[18] Wayner, P., Mimic functions. Cryptologia, 1992. 16(3): p. 193-214.
[19] Johnson, S.C., Yacc: Yet another compiler-compiler. Vol. 32. 1975: Bell Laboratories Murray Hill, NJ.
[20] Kodaganallur, V., Incorporating language processing into java applications: A JavaCC tutorial. Software, IEEE, 2004. 21(4): p. 70-77.
[21] Viswanadha, S. and S. Sankar, Java compiler compiler (JavaCC)-The java parser generator. Java. net, https://javacc. dev. java. net/, accessed Aug, 2009. 23.
[22] Ryan, C., M. O'Neill, and J. Collins. Grammatical evolution: solving trigonometric identities. in Proceedings of Mendel. 1998. Citeseer