Sayfa 1

Zincir bağlantı
Merkezi Olmayan Oracle Ağı
Steve Ellis, Ari Juels
†
ve Sergey Nazarov
4 Eylül 2017 (v1.0)
Öz
Akıllı sözleşmeler, birçok sektörde devrim yaratmaya hazırlanıyor.
hem geleneksel yasal anlaşmalara hem de merkezi olarak otomatikleştirilmiş dijital
anlaşmalar. Hem performans doğrulama hem de yürütme manuel işlemlere dayanır
sözleşme taraflarından birinden veya programlayan otomatik bir sistemden
ilgili değişiklikleri ically alır ve günceller. Maalesef onların yüzünden
temel fikir birliği protokolleri, akıllı sözleşmelerin üzerinde çalıştığı blok zincirleri
harici sistemlerle yerel iletişimi destekleyemez.
Bugün, bu sorunun çözümü, adı verilen yeni bir işlevsellik sunmaktır.
dış dünyaya bağlantı sağlayan bir oracle. Mevcut kahinler
merkezi hizmetler. Bu tür hizmetleri kullanan herhangi bir akıllı sözleşmenin tek bir noktası vardır
Başarısızlık, onu geleneksel, merkezi olarak çalışan bir dijitalden daha güvenli hale getirmez.
anlaşma.
Bu yazıda, merkezi olmayan bir oracle ağı olan ChainLink'i sunuyoruz. Biz ...
ChainLink'in sözleşmeler için sağladığı zincir üzerindeki bileşenleri kazanın
harici bağlantı ve ağ düğümlerine güç sağlayan yazılım. Biz
hem basit bir zincir üzerinde sözleşme veri toplama sistemi hem de daha fazlasını sunun
verimli zincir dışı fikir birliği mekanizması. Destekleyici itibarı da tarif ediyoruz
ve ChainLink için kullanıcıların bilgilenmesine yardımcı olan güvenlik izleme hizmetleri
sağlayıcı seçimleri ve agresif reklamların altında bile sağlam hizmet elde edin
sarial koşullar. Son olarak, ideal bir oracle'ın özelliklerini şöyle tanımlıyoruz:
güvenlik stratejimiz için rehberlik etmek ve gelecekteki olası iyileştirmeleri düzenlemek,
zengin özellikli oracle programlama, veri kaynağı altyapı modu dahil
kurgular ve gizli akıllı sözleşme yürütme.

Sayfa 2

İçindekiler

---

3. Sayfa

1. Giriş

Akıllı sözleşmeler, merkezi olmayan altyapı üzerinde çalışan uygulamalardır.
bir blok zinciri olarak. Hiçbir partinin (yarattıkları bile olsa) kurcalamaya dayanıklıdırlar.
ator) kodunu değiştirebilir veya çalıştırılmasına müdahale edebilir. Tarihsel olarak, sözleşmeler
kodda somutlaşan, onları al-
Ayrıcalıklı bir tarafça teration, fesih ve hatta silme. Aksine, akıllı
tüm tarafları yazılı bir sözleşmeye bağlayan sözleşmelerin icra garantileri,
güvene dayanmayan yeni ve güçlü bir güven ilişkisi türü yaratın.
herhangi bir parti. Çünkü kendi kendini doğrulayan ve kendi kendini çalıştıran (yani, kurcalanmaya karşı korumalı)
yukarıda açıklandığı gibi), akıllı sözleşmeler bu nedenle gerçekleştirmek için üstün bir araç sunar ve dijital anlaşmaları yönetmek.

Akıllı sözleşmelerin somutlaştırdığı güçlü yeni güven modeli yine de
yeni bir teknik zorluk: bağlantı. İlginç [27] 1 smart'ın büyük çoğunluğu
sözleşme uygulamaları, temel kaynaklardan gelen gerçek dünya hakkındaki verilere dayanır,
özellikle blok zincirinin dışındaki veri akışları ve API'ler. Yüzünden
bir blok zinciri olan blok zincirlerinin temelini oluşturan fikir birliği mekanizmalarının mekaniği
bu tür kritik verileri doğrudan getiremez.

Akıllı sözleşme bağlantı sorununa şu şekilde bir çözüm öneriyoruz:
ChainLink, güvenli bir oracle ağı. ChainLink'i diğer oracle'dan ayıran nedir?
çözümleri, tamamen merkezi olmayan bir ağ olarak çalışma yeteneğidir. Bu merkezi olmayan
yaklaşımı, herhangi bir tek tarafa olan güveni sınırlayarak, değer verilen kurcalamaya dayanıklı
kaliteyi mümkün kılar
akıllı sözleşmelerde akıllı sözleşmeler arasında uçtan uca operasyona genişletilecek
ve güvendikleri API'ler. Akıllı sözleşmelerin dışarıdan farkına varması, yani yetenekli olması
zincir dışı kaynaklarla etkileşime girme,
günümüzde kullanılan dijital anlaşmalar.

Bugün, geleneksel sözleşmeye dayalı anlaşmalardaki aslan payı,
Çeteleyle otomatikleştirilmiş, sözleşmeye dayalı performansı kanıtlamak için harici verileri kullanın
ve veri gerektirir
harici sistemlere gönderilecek çıktılar. Akıllı sözleşmeler bunların yerini aldığında
sözleşmeye dayalı mekanizmalar, aynı türlerin yüksek garantili versiyonlarını gerektireceklerdir
veri giriş ve çıkışlarının sayısı. Potansiyel yeni nesil akıllı sözleşmelerin örnekleri
ve veri gereksinimleri şunları içerir:

• Tahviller, faiz oranı türevleri gibi menkul kıymetler akıllı sözleşmeleri ve birçok
diğerleri piyasa fiyatlarını ve piyasa referansını bildiren API'lere erişim gerektirecektir
veriler, örneğin faiz oranları.
1 Günümüzde Ethereum'da akıllı sözleşmelerin ana kullanımı,
çoğu akıllı sözleşme ağında ortak işlevsellik. Şu anda jetonlara odaklanıldığına inanıyoruz
diğer birçok olası uygulamanın hariç tutulmasının nedeni, oracle hizmetlerinin yeterli olmamasından
kaynaklanıyorsa,
durum ChainLink özellikle bir çare bulmayı amaçlamaktadır.
3

---

4. sayfa

• Sigorta akıllı sözleşmelerinin, aşağıdakilerle ilgili IoT verileri hakkında veri beslemelerine ihtiyacı
olacaktır.
sigortalanabilir olay, örneğin: deponun manyetik kapısı kilitlendi mi?
ihlal zamanı, şirketin çevrimiçi güvenlik duvarı mıydı yoksa yaptığınız uçuş
zamanında varmak için sigorta.
• Ticaret finansmanı akıllı sözleşmelerinin gönderiler hakkında GPS verilerine,
tedarik zinciri ERP sistemleri ve sevk edilen mallarla ilgili gümrük verileri
sözleşme yükümlülüklerinin yerine getirildiğini teyit etmek için.
Bu örneklerde ortak olan bir başka sorun da akıllı sözleşmelerin yetersiz olmasıdır.
verileri zincir dışı sistemlere çıkarmak için. Bu tür bir çıktı genellikle bir ödeme şeklini alır
kullanıcıların halihazırda sahip olduğu geleneksel merkezi altyapıya yönlendirilen mesaj
örneğin banka ödemeleri, PayPal ve diğer ödeme ağları için hesaplar. ChainLink'ler
akıllı bir kullanıcı adına verileri API'lere ve çeşitli eski sistemlere güvenli bir şekilde gönderme
yeteneği
sözleşme, dışarıdan farkında olan kurcalamaya dayanıklı sözleşmelerin oluşturulmasına izin verir.
Teknik rapor yol haritası
Bu tanıtım belgesinde *, ChainLink mimarisini (Bölüm 2) inceliyoruz . Biz sonra
oracle'lar için güvenliği nasıl tanımladığımızı açıklayın (Bölüm 3). ChainLink'i tanımlıyoruz
oracle'ların ve veri kaynaklarının ademi merkeziyetçiliği / dağıtımı yaklaşımı (Bölüm 4) ,
ve ChainLink tarafından önerilen dört güvenlik hizmetiyle ilgili bir tartışma ile devam edin,
LINK token'larının oynadığı rol (Bölüm 5). Daha sonra önerilen bir
daha iyi gizlilik korumalarını içeren uzun vadeli kalkınma stratejisi,
güvenilir donanım kullanımı, altyapı değişiklikleri ve genel oracle programlanabilirliği
(Bölüm 6). Alternatif oracle tasarımlarını kısaca gözden geçiriyoruz (Bölüm 7) ve
ChainLink'e rehberlik eden tasarım ilkeleri ve felsefesinin kısa bir tartışması ile
geliştirme (Bölüm 8) .
2 Mimari Genel Bakış
ChainLink'in temel işlevsel hedefi, iki ortam arasında köprü oluşturmaktır: zincir içi ve dışı
Zincir. Her bir ChainLink bileşeninin mimarisini aşağıda açıklıyoruz. Zincir bağlantı
başlangıçta Ethereum [16] , [35] üzerine inşa edilecek , ancak tüm liderleri desteklemesi niyetindeyiz
hem zincir dışı hem de zincirler arası etkileşimler için akıllı sözleşme ağları. İkisinde de
Zincir üzerinde ve zincir dışı versiyonlarda, ChainLink modülerlik göz önünde bulundurularak
tasarlanmıştır.
ChainLink sisteminin her parçası yükseltilebilir, böylece farklı bileşenler
daha iyi teknikler ve rakip uygulamalar ortaya çıktıkça değiştirilmelidir.
4

5.Sayfa

2.1 Zincir Üzerinde Mimari
Bir oracle hizmeti olarak, ChainLink düğümleri veri taleplerine veya yapılan sorgulara yanıt verir
Sözleşme talep etme olarak adlandırdığımız bir kullanıcı sözleşmesi tarafından veya adına ve
USER-SC ile gösterilir. ChainLink'in sözleşme talep etmek için zincir içi arayüzü kendisidir
CHAINLINK-SC ile belirttiğimiz bir zincir içi sözleşme.

CHAINLINK-SC'nin arkasında, ChainLink, üç parçadan oluşan bir zincir üstü bileşene sahiptir.
ana sözleşmeler: bir itibar sözleşmesi, bir sipariş eşleştirme sözleşmesi ve bir toplama
sözleşme. İtibar sözleşmesi, oracle-hizmet-sağlayıcı performansının kaydını tutar
metrikler. Sipariş eşleştirme akıllı sözleşme, önerilen bir hizmet seviyesi sözleşmesini alır,
SLA parametrelerini günlüğe kaydeder ve oracle sağlayıcılarından teklifleri toplar. Daha sonra
teklifleri seçer
itibar sözleşmesini kullanarak ve oracle SLA'yı sonuçlandırır. Toplama sözleşmesi
Oracle sağlayıcılarının yanıtlarını toplar ve nihai toplu sonucunu hesaplar
ChainLink sorgusu. Aynı zamanda oracle sağlayıcı metriklerini itibara geri besler
sözleşme. ChainLink sözleşmeleri, modüler bir şekilde tasarlanır ve onlara izin verir.
gerektiğinde kullanıcılar tarafından yapılandırılmalı veya değiştirilmelidir. Zincir üzerindeki iş
akışında üç
adımlar: 1) oracle seçimi, 2) veri raporlama, 3) sonuç toplama.
Oracle Seçimi Bir oracle hizmetleri satın alan kişi,
bir hizmet seviyesi sözleşmesi (SLA) teklifi. SLA teklifi aşağıdaki gibi ayrıntıları içerir:
Sorgu parametreleri ve alıcı tarafından ihtiyaç duyulan oracle sayısı. Bunlara ek olarak,
alıcı, itibar ve toplu sözleşmeleri belirtir.
anlaşmanın geri kalanı.
Zincir üzerinde tutulan itibarın daha sağlam bir veri kümesiyle birlikte kullanılması
Geçmiş sözleşmelerin günlüklerinden toplanan alıcılar manuel olarak sıralayabilir, filtreleyebilir ve
seçebilir
zincir dışı listeleme hizmetleri aracılığıyla oracle'lar. Niyetimiz ChainLink'in bir
bu tür bir listeleme hizmeti, ChainLink ile ilgili tüm günlüklerin toplanması ve ikililerin doğrulanması
listelenen oracle sözleşmeleri. Listeleme hizmetini ve itibar sistemlerini daha ayrıntılı olarak
detaylandırıyoruz
Bölüm 5 . Listeleri oluşturmak için kullanılan veriler blok zincirinden çekilecek,
alternatif oracle listeleme hizmetlerinin oluşturulmasına izin verir. Alıcılar gönderecek
Zincir dışı oracle'lara SLA teklifleri ve SLA'yı sonuçlandırmadan önce anlaşmaya varın
zincir üzerinde.
Tüm durumlarda manuel eşleştirme mümkün değildir. Örneğin, bir sözleşme
yüküne yanıt olarak oracle hizmetlerini dinamik olarak talep etmesi gerekir. Otomatik çözüm
tionlar bu sorunu çözer ve kullanılabilirliği artırır. Bu nedenlerden dolayı otomatik oracle
eşleştirme, ChainLink tarafından sipariş eşleştirme kullanılarak önerilmektedir.
sözleşmeler.
Alıcı, ora ile iletişime geçmek yerine SLA teklifini belirttikten sonra
doğrudan, SLA'yı bir sipariş eşleştirme sözleşmesine sunacaklardır. Teslim
sipariş eşleştirme sözleşmesine teklifin, oracle sağlayıcılarının yapabileceği bir günlüğü tetikler

---

Sayfa 6

Yeteneklerine ve hizmet hedeflerine göre izleyin ve filtreleyin. ChainLink düğümleri
sonra teklif için teklif verilip verilmeyeceğini seçin, yalnızca sözleşme kabul eder
SLA'nın gereksinimlerini karşılayan düğümlerden teklifler. Oracle servis sağlayıcısı
bir sözleşmedeki teklifler, özellikle ceza tutarını ekleyerek taahhüt ederler
SLA'da tanımlandığı gibi, yanlış davranışları nedeniyle kaybolacak.
Teklifler, teklif verme penceresinin tamamı için kabul edilir. SLA,
yeterince nitelikli teklif aldı ve teklif verme penceresi sona erdi, talep edilen
teklif havuzundan seçilen oracle sayısı. Olan ceza ödemeleri
ihale sürecinde teklif edilen, seçilmeyen oracle'lara iade edilir ve
kesinleşmiş bir SLA kaydı oluşturulur. Kesinleşmiş SLA kaydedildiğinde, bir
seçilen oracle'ları bilgilendiren günlük. Kahinler daha sonra ayrıntılı görevi yerine getirir
SLA tarafından.
Veri Raporlama Yeni oracle kaydı oluşturulduktan sonra, zincir dışı oracle'lar
anlaşmayı yürütmek ve zincir üzerinde rapor vermek. Zincir dışı hakkında daha fazla ayrıntı için
etkileşimler için Bölüm 2.2 ve 4'e bakınız .

Sonuç Toplama Oracle'lar, sonuçlarını oracle con-
yolu, sonuçları toplama sözleşmesine beslenecektir. Toplama sözleşmesi
toplu sonuçları hesaplar ve ağırlıklı bir cevap hesaplar. Her birinin geçerliliği
oracle yanıtı daha sonra itibar sözleşmesine bildirilir. Son olarak, ağırlıklı
yanıt, USER-SC'de belirtilen sözleşme işlevine döndürülür.
Dıştaki veya yanlış değerleri tespit etmek, her türe özgü bir sorundur
veri beslemesi ve uygulaması. Örneğin, dışarıdaki cevapları tespit etmek ve reddetmek
sayısal veriler için ortalamadan önce gerekli olabilir, ancak boole için gerekli olmayabilir. Bu yüzden,
belirli bir toplama sözleşmesi olmayacak, ancak yapılandırılabilir bir sözleşme adresi olacaktır
hangi alıcı tarafından belirtilir. ChainLink, standart bir ag-
gregating sözleşmeler, ancak özelleştirilmiş sözleşmeler de belirtilebilir,
standart hesaplama arayüzüne uygundur.
2.2 Zincir Dışı Mimari
Zincir dışı, ChainLink başlangıçta ağa bağlı bir oracle düğümleri ağından oluşur.
Ethereum ağı ve bunun tüm önde gelen akıllı sözleşme ağlarını desteklemesi niyetindeyiz.
İşler. Bu düğümler bağımsız olarak zincir dışı isteklere verilen yanıtları toplar. Biz gibi
aşağıda açıklayın, bireysel yanıtları olası birkaç taneden biri aracılığıyla toplanır
konsensüs mekanizmaları, talep eden bir uzlaşmaya döndürülen küresel bir yanıta dönüştürür.
kanal USER-SC. ChainLink düğümleri, standart açık kaynak çekirdek tarafından desteklenmektedir
standart blok zinciri etkileşimlerini, zamanlamayı ve kontrolleri yöneten uygulama
ortak dış kaynaklarla bağlantı kurma. Düğüm operatörleri yazılım eklemeyi seçebilir

---

7. Sayfa

Operatörlerin ek teklif sunmasına olanak tanıyan harici adaptörler olarak bilinen uzantılar
özel zincir dışı hizmetler. ChainLink düğümleri zaten birlikte konuşlandırıldı.
kurumsal ortamlarda hem genel blok zincirlerini hem de özel ağları destekleyin; etkinleştirmek
Merkezi olmayan bir şekilde çalışan düğümler, ChainLink ağı için motivasyon kaynağıdır.
ChainLink Çekirdeği. Çekirdek düğüm yazılımı,
blok zinciri, zamanlama ve dengeleme çalışmaları, çeşitli harici hizmetlerinde. İş
ChainLink düğümleri tarafından yapılır, atamalar olarak biçimlendirilir. Her ödev bir dizi
ardışık düzen olarak işlenen alt görevler olarak bilinen daha küçük iş özellikleri. Her biri
alt görevin sonucu bir sonrakine geçirmeden önce gerçekleştirdiği belirli bir işlem vardır.
alt görev ve nihayetinde nihai sonuca ulaşmak. ChainLink'in düğüm yazılımı ile birlikte gelir
HTTP istekleri, JSON ayrıştırma ve
çeşitli blok zinciri formatları.
Harici Adaptörler. Yerleşik alt görev türlerinin ötesinde, özel alt görevler de
bağdaştırıcılar oluşturarak tanımlanır. Adaptörler, minimum REST ile harici hizmetlerdir
API. Bağdaştırıcıları hizmet odaklı bir şekilde modelleyerek, herhangi bir programdaki programlar
Ming dili, küçük bir ara API eklenerek kolayca uygulanabilir
programın önünde. Benzer şekilde, karmaşık çok adımlı API'lerle etkileşimde bulunmak,
parametrelerle tek tek alt görevlere basitleştirilebilir.
Alt Görev Şemaları. Pek çok bağdaştırıcının açık kaynaklı olacağını tahmin ediyoruz, bu nedenle
bu hizmetler çeşitli topluluk üyeleri tarafından denetlenebilir ve çalıştırılabilir. Birçoğuyla
birçok farklı geliştirici tarafından geliştirilen farklı adaptör türleri,
adaptörler arasında uyumluluk önemlidir.
ChainLink şu anda JSON Schema [36] tabanlı bir şema sistemiyle çalışmaktadır ,
her bağdaştırıcının hangi girişlere ihtiyaç duyduğunu ve nasıl biçimlendirilmesi gerektiğini belirlemek
için. Sim-
Bağdaştırıcılar, her bir alt görevin biçimini açıklamak için bir çıktı şeması belirtir.
çıktı.
3 Oracle Güvenliği
ChainLink'in güvenlik mimarisini açıklamak için, önce neden ikinci olduğunu açıklamalıyız.
temizlik önemlidir ve ne anlama geldiği.
Kahinler neden güvende olmalı? Bölüm 1'deki basit örneklerimize dönersek,

bir akıllı sözleşme güvenliği yanlış bir veri beslemesi alırsa, yanlış tarafa ödeme yapabilir,
Akıllı sözleşme sigortası veri beslemeleri sigortalı taraf tarafından değiştirilebiliyorsa
7

---

8. Sayfa

Şekil 1: ChainLink iş akışı: 1) USER-SC, zincir üzerinde bir istekte bulunur; 2) CHAINLINK-SC
oracle'lar için bir olay kaydeder; 3) ChainLink çekirdeği olayı alır ve atamayı yönlendirir
bir adaptöre; 4) ChainLink adaptörü, harici bir API'ye bir istek gerçekleştirir; 5) Zincir Bağlantı
bağdaştırıcı yanıtı işler ve çekirdeğe geri iletir; 6) ChainLink temel raporları
CHAINLINK-SC'ye veri; 7) CHAINLINK-SC yanıtları toplar ve geri gönderir
USER-SC'ye tek bir yanıt olarak.
sigorta dolandırıcılığı olabilir ve bir ticaret finansmanı sözleşmesine verilen GPS verileri
veri sağlayıcısından ayrıldıktan sonra değiştirilebilir,
gelmedi.
Daha genel olarak, defteri veya ilan panosu ile iyi işleyen bir blok zinciri
soyutlama, çok güçlü güvenlik özellikleri sunar. Kullanıcılar blok zincirine güvenirler.
İşlemleri doğru bir şekilde doğrulayan ve verilerin
değişmiş. Buna fiilen güvenilir bir üçüncü taraf gibi davranıyorlar (tartıştığımız bir kavram)
aşağıdaki uzunluk). Destekleyici bir oracle hizmeti, uygun bir güvenlik düzeyi sunmalıdır.
desteklediği blockchain ile. Bir kahin de bu nedenle kullanıcılara şu şekilde hizmet etmelidir:
etkili ve güvenilir bir üçüncü taraf, çok doğru ve zamanında yanıtlar
yüksek olasılık. Herhangi bir sistemin güvenliği, ancak en zayıf halkası kadar güçlüdür,
Bu nedenle, bir kuyunun güvenilirliğini korumak için oldukça güvenilir bir kahin gereklidir.
tasarlanmış blok zinciri.
Oracle güvenliğinin tanımlanması: İdeal bir görünüm. Oracle se-
curity, önce onu tanımlamalıyız. Oracle hakkında akıl yürütmenin öğretici, ilkeli bir yolu
güvenlik aşağıdaki düşünce deneyinden kaynaklanmaktadır. Güvenilir bir üçüncü düşünün
parti (TTP) - her zaman talimatları yerine getiren ideal bir varlık veya işlevsellik
mektuba sadakatle - bir kehanet yürütmekle görevlendirildi. Bu kahini göstereceğiz
ORACLE tarafından (kullanıcıların tamamen güvendiği bir varlığı belirtmek için genel olarak tüm
büyük harfleri kullanarak) ve
TTP'nin verileri tamamen güvenilir bir Src veri kaynağından aldığını varsayalım .
Bu sihirli hizmet ORACLE göz önüne alındığında, ondan hangi talimatları yerine getirmesini isterdik?
Doğruluk özelliği olarak da anılan bütünlük özelliğini elde etmek için
erty [24], biz sadece ORACLE'ın aşağıdaki adımları gerçekleştirmesini isteriz:
8

Sayfa 9

Şekil 2: İdeal bir oracle ORACLE davranışı aşağıdaki adımlarla tanımlanır: 1) İsteği kabul et; 2)
Veri elde edin; 3) Verileri döndür. Ek olarak, bir talebin gizliliğini korumak için,
şifresini çözen ORACLE, Src sorgusu dışında içerdiği verileri asla kullanmaz veya açıklamaz .
1. İsteği kabul edin: USER-SC akıllı sözleşmeden bir istek alın Req = (Src , τ, q)
bir hedef veri kaynağı Src , bir zaman veya zaman aralığı ve bir sorgu belirten
q;
Verilerini elde etmek: 2. sorgu q gönder Src süresi x olarak;
3. Verileri döndür: a cevabını aldıktan sonra, a akıllı sözleşmeye dönün.
Doğru bir şekilde uygulanan bu basit talimatlar, güçlü, anlamlı, ancak
basit güvenlik kavramı. Sezgisel olarak, ORACLE'ın güvenilir bir
Src ve USER-SC arasında köprü .2 Örneğin, Src https://www.FountOfKnowledge.com ise,
τ 16:00 ve q = "fiyat kodu INTC", ORACLE bütünlüğünü garanti eder
USER-SC'ye saat 16: 00'da sorgulanan INTC fiyatını tam olarak sağlayacağını
https://www.FountOfKnowledge.com.
Gizlilik, kahinler için arzu edilen bir başka özelliktir. USER-SC, Talep gönderirken
ORACLE'a blok zincirindeki açıklıkta, Req halka açıktır. Birçok durum var

Req'in hassas olduğu ve yayınlanması zararlı olabileceği. USER-SC bir
örneğin uçuş sigortası sözleşmesi ve ORACLE'a bir
belirli bir kullanıcının uçuşu (q = "Ether Air Flight 338"), sonuç şu olacaktır:
kullanıcının uçuş planları tüm dünyaya açıklanır. USER-SC bir sözleşmeyse
2 Elbette burada birçok ayrıntı atlanmıştır. ORACLE, hem USER-SC ile iletişim kurmalıdır
ve Src'yi güvenli, yani kurcalamaya dayanıklı kanallar üzerinden kaynaklayın. (Src bir web
sunucusuysa, TLS gereklidir.
USER-SC ile iletişim kurduğunuzda, ORACLE doğru blok zincirini kazıyıp dijital olarak
imzaladığınızdan emin olmalıdır
Uygun bir şekilde.)

Sayfa 10

finansal ticaret, Req bir kullanıcının alım satımları ve portföyü hakkında bilgi sızdırabilir.
Elbette birçok başka örnek var.
Req'in gizliliğini korumak için, Req'teki verilerin şifrelenmesini isteyebiliriz
ORACLE'a ait bir (genel anahtar) altında. TTP doğasından yararlanmaya devam etmek
ORACLE için, daha sonra ORACLE'a bilgi akışı kısıtlamasını verebiliriz:
Req'in şifresini çözdükten sonra, Src'yi sorgulamak dışında Req'teki verileri asla açıklamayın veya
kullanmayın .
Kullanılabilirlik, sonuncusu gibi başka önemli oracle özellikleri de vardır.
klasik CIA (Gizlilik-Bütünlük-Kullanılabilirlik) üçlüsü. Gerçekten ideal bir hizmet VEYA-
Elbette ACLE asla düşmez. Kullanılabilirlik ayrıca daha incelikli
sansür direnci gibi özellikler: Dürüst bir ORACLE, par-
ticular akıllı sözleşmeler yapar ve isteklerini reddeder.
Güvenilir bir üçüncü taraf kavramı, ideal bir işlev kavramına benzer.
ality [7], belirli modellerde kriptografik protokollerin güvenliğini kanıtlamak için kullanılır. Biz
bir blok zincirini benzer terimlerle modelleyebilir ve onu bir TTP açısından kavramsallaştırabilir
ideal bir ilan panosu sağlayan. Talimatları işlemleri kabul etmektir,
doğrulayın, seri hale getirin ve ilan tahtasında kalıcı olarak saklayın,
yalnızca ekli bir veri yapısı.
Neden ideal oracle'a (ORACLE) ulaşmak zordur. Tabii ki hayır
mükemmel güvenilir veri kaynağı Src . Veriler tehlikesiz veya kötü amaçla bozulmuş olabilir
hatalı web siteleri, aldatma hizmeti sağlayıcıları veya dürüst hatalar nedeniyle.
Eğer Src ORACLE in gibi bir TTP gibi tam olarak işletmek bile o zaman, güvenilir değildir
yukarıda yapılandırılmış olsa da, yine de istediğimiz güvenlik kavramını tam olarak
karşılamıyor. Verilen bir
hatalı kaynak Src , yukarıda tanımlanan bütünlük özelliği artık bir oracle'ın
cevap a doğru. Intel'in gerçek fiyatı 40 $ ve https://www.FountOfKnowledge.com ise
örneğin 50 $ olarak yanlış bildirirse, ORACLE yanlış a = 50 $ değeri gönderir
USER-SC'ye. Tek bir kaynak Src kullanıldığında bu sorun kaçınılmazdır . ORACLE
Src'nin sorularına verdiği cevapların doğru olup olmadığını bilmenin hiçbir yolu yoktur .
Elbette daha büyük bir sorun, ORACLE için TTP'mizin sadece bir abstrac-
yon. Hiçbir hizmet sağlayıcı koşulsuz olarak güvenilir değildir. En iyi niyetliler bile
hatalı veya saldırıya uğramış olabilir. Dolayısıyla, bir kullanıcının veya akıllı sözleşmenin
ORACLE hizmetinin talimatlarını sadakatle yerine getireceğine dair mutlak güvence.
ChainLink, bu ideal işlevsellik açısından güvenlik protokolleri hakkında nedenler
ORACLE. ChainLink'teki amacımız, özelliklerle gerçek bir dünya sistemine ulaşmaktır.
gerçekçi güven varsayımları altında ORACLE'unkilere mümkün olduğunca yakın. Biz şimdi
nasıl olduğunu açıkla.
Aşağıda anlatılanları basitleştirmek için, şimdi CHAINLINK-SC ile tüm
ChainLink sözleşmeleri kümesi, yani tam zincir içi işlevselliği (yalnızca arayüzü değil

---

Sayfa 11

sözleşme talep etmek için). Böylelikle birden fazla bireysel sözleşmeyi soyutlarız
aslında sistem mimarisinde kullanılır.

4 ChainLink Merkeziyetsizleştirme Yaklaşımı

Hatalı düğümlere karşı koruma sağlamak için üç temel tamamlayıcı yaklaşım öneriyoruz:
(1) Veri kaynaklarının dağılımı; (2) Kahinlerin dağılımı; ve (3) Güvenilir kullanım
donanım. Ademi merkeziyetçiliği içeren ilk iki yaklaşımı,
bu bölüm. Farklı ve güvenilir donanımlar için uzun vadeli stratejimizi tartışıyoruz.
tamamlayıcı yaklaşım, Bölüm 6'da.

4.1 Kaynakları dağıtma

Hatalı tek bir kaynak Src ile başa çıkmanın basit bir yolu , birden çok kaynaktan veri elde etmektir.
kaynaklar, yani veri kaynağını dağıtır. Güvenilir bir ORACLE bir koleksiyonu sorgulayabilir
Kaynaklar için Src 1 , Src 2 , ..., Src k , yanıtları a 1 , a 2 , ..., a k elde edin ve bunları
tek bir cevap A = agg (a 1 , a 2 , ..., a k). ORACLE bunu herhangi bir sayıda yapabilir
Yollardan. Biri, örneğin, çoğunluk oylamadır. Kaynakların çoğu geri dönerse
özdeş değer a, agg işlevi a'yı döndürür; aksi takdirde bir hata döndürür. Bunda
durumda, çoğunluk (> k / 2) kaynaklarının doğru çalışması şartıyla, ORACLE
her zaman doğru bir A değeri döndürecektir.

Birçok alternatif fonksiyon agg, hatalı verilere karşı sağlamlık sağlayabilir veya
Veri değerlerindeki zaman içindeki dalgalanmaları ele alın (örneğin, hisse senedi fiyatları). Örneğin, agg
aykırı değerleri atabilir (örneğin, en büyük ve en küçük değerler a i) ve ortalama
kalanlardan.

Elbette, hatalar veri kaynakları arasında zayıflatacak şekilde ilişkilendirilebilir.
toplulaştırma tarafından sağlanan güvenceler. Src sitesi 1 = EchoEcho.com kendi verilerini alırsa
adlı Src 2 = TheHorsesMouth.com, bir hata Src 2 her bir hata anlamına gelecektir Src 1 .

Veri kaynakları arasında daha ince korelasyonlar da ortaya çıkabilir. Chainlink ayrıca
veri kaynaklarının bağımsızlığını haritalamak ve raporlamak için araştırma yapmak
Kahinler ve kullanıcıların istenmeyen korelasyonları önleyebilmesi için kolay sindirilebilir bir yol.

4.2 Oracle'ları dağıtma

Kaynakların dağıtılabildiği gibi, ideal hizmetimiz ORACLE'ın kendisi de yaklaşık olabilir
dağıtılmış bir sistem olarak çiftleşti. Bu, tek bir yekpare kehanet yerine
O düğümü , bunun yerine n farklı oracle düğümünden oluşan bir koleksiyonumuz olabilir { O 1 , O 2 ,
..., O n }.

Her oracle O i , kendi farklı veri kaynakları kümesiyle iletişim kurar veya olmayabilir.

Sayfa 12

Şekil 3: İstekler hem oracle'lara hem de veri kaynaklarına dağıtılır. Bu şekil gösterir
böyle iki seviyeli dağıtımın bir örneği.

diğer kahinlerinkilerle örtüşüyor. O i , veri kaynaklarından gelen yanıtları toplar ve
bir Req sorgusuna kendi farklı cevabını A i çıkarır.

Bu oracle'lardan bazıları hatalı olabilir. Yani açıkça tüm kahinlerin cevapları
A 1 , A 2 , ..., A n'nin güvenilir bir şekilde tek bir yazar olarak toplanması gerekecektir.
itatif değer A. Ancak hatalı oracle olasılığı göz önüne alındığında, bu nerede ve nasıl olacak?
ChainLink'te toplanma mı oluyor?

İlk çözüm: Sözleşme içi toplama. İlk önerdiğimiz çözümümüz
ChainLink, sözleşme içi toplama adı verilen basit bir bağlantı olacaktır. CHAINLINK-SC—
yine, ChainLink'in zincir üzerindeki kısmını ifade eder — kendisi oracle'ı bir araya getirecektir.
tepkiler. (Alternatif olarak, CHAINLINK-SC başka bir toplama sözleşmesi çağırabilir, ancak
kavramsal basitlik için iki bileşenin tek bir sözleşme oluşturduğunu varsayıyoruz.)

Başka bir deyişle, CHAINLINK-SC, bazı işlevler için A = Agg (A 1 , A 2 , ..., A n) hesaplayacaktır.
Agg (agg'ye benzer, yukarıda açıklandığı gibi) ve sonucu A USER-SC'ye gönderin.

Bu yaklaşım, küçük n için pratiktir ve birkaç farklı faydası vardır:

• Kavramsal basitlik: Kahin dağıtılmış olmasına rağmen, tek bir
varlık, CHAINLINK-SC, Agg.
• Güvenilirlik: CHAINLINK-SC'nin kodu halka açık bir şekilde incelenebildiğinden,
davranış doğrulanabilir. (CHAINLINK-SC nispeten küçük ve basit olacaktır.
kod parçası.) Ek olarak, CHAINLINK-SC'nin yürütülmesi,

Sayfa 13

Zincir. Böylece kullanıcılar, yani USER-SC'nin yaratıcıları, yüksek derecede güven elde edebilirler.
CHAINLINK-SC'de.
• Esneklik: CHAINLINK-SC en çok istenen toplama işlevlerini uygulayabilir
Agg - çoğunluk işlevi, ortalama vb.
Basit olduğu kadar, bu yaklaşım yeni ve ilginç bir teknik zorluk sunuyor.
yani serbest yükleme sorunu. Hile yapan bir kahin O z yanıtı gözlemleyebilir
Bir i başka kahin ait O i ve kopyalayın. Bu şekilde oracle O z ,
sorgu başına ücret alabilen veri kaynaklarını sorgulama. Serbest yükleme güvenliği zayıflatır
veri kaynağı sorgularının çeşitliliğini zayıflatarak ve ayrıca oracle'ları caydırarak
hızlı yanıt vermekten: Yavaş yanıt vermek ve serbest yükleme daha ucuz bir stratejidir.
Bu soruna iyi bilinen bir çözüm öneriyoruz, yani bir commit /
şemayı ortaya çıkarın. İlk turda oracle'lar CHAINLINK-SC kriptografik taahhüdü gönderir
cevaplarına yönlendirir. CHAINLINK-SC yeterli sayıda yanıt aldıktan sonra,
kahinlerin yanıtlarını ortaya çıkardığı ikinci bir tur başlatır.
Algoritma 1 , verilen kullanılabilirliği garanti eden basit bir sıralı protokolü gösterir.
3f + 1 düğüm. Serbest yüklemeyi önlemek için bir teslim etme / gösterme şeması kullanır. Oracle
yanıtlar kaldırılır ve bu nedenle potansiyel bir serbest yükleyiciye ancak sonuçta maruz kalır.
taahhütler verilmiş, böylece serbest yükleyicinin diğer
oracles'ın yanıtları.
Zincir üzeri protokoller, senkronize protokol çözümünü desteklemek için blok sürelerinden
yararlanabilir.
işaretler. Bununla birlikte, ChainLink'te, oracle düğümleri, sahip olabilecek kaynaklardan veri alır.
son derece değişken yanıt süreleri ve düğümlere göre taahhüt dışı bırakma süreleri aşağıdakilerden
dolayı değişebilir:
örneğin, Ethereum'da farklı gaz fiyatlarının kullanılması. Mümkün olan en hızlı protokolü sağlamak
için
duyarlılık, bu nedenle, Alg. 1 eşzamansız bir protokol olarak tasarlanmıştır.
Burada, Taahhüt r (A), tanık r ile A değerinin bir taahhüdünü belirtirken, SID
geçerli oturum kimlikleri kümesini gösterir. Protokol, kimliği doğrulanmış kanalları varsayar
tüm oyuncular arasında.
Alg'ı görmek kolaydır. 1 başarıyla sona erecek. İçinde 3f + 1 düğüm verildi
toplam, en fazla f hatalı, bu nedenle en az 2f + 1 4. Adımda taahhütler gönderecektir.
bu taahhütler, en çok f hatalı düğümlerden gelir, bu nedenle en az f + 1,
dürüst düğümler. Bu tür tüm taahhütler, er ya da geç kaldırılacaktır.
Ek olarak, A'nın Alg'de doğru olacağını görmek kolaydır . 1. f + 1'in
A tek değerindeki taahhütler, en az birinin dürüst bir düğümden gelmesi gerekir.
Alg aracılığıyla sözleşme içi toplama. 1, Chain tarafından desteklenen ana yaklaşım olacaktır.
Kısa vadede bağlantı. Önerilen ilk uygulama, daha çok
algoritmanın gelişmiş, eşzamanlı varyantı. Uzun vadeli teklifimiz yansıtılır
oldukça daha karmaşık OCA protokolünde (Zincir Dışı Toplama) belirtilen
Ek A'daki Algoritmalar 2 ve 3, OCA,

Sayfa 14

Algoritma 1 InChainAgg ({ O i } n
i = 1) (CHAINLINK-SC kodu)

1: USER-SC'den Talep alınana kadar bekleyin.
2: sid ← $ SID
3: Yayın (istek, sid).
4: 2f +1 mesajlarının C'sine (commit, c i = Commit r i (A i), sid) farklı olana kadar bekleyin.
O ben aldım.
5: Yayın (taahhüt edildi, sid).
6: f + 1 farklı geçerli geri bildirimlerin D ayarına kadar bekleyin (decommit, (r i , A i), sid)
Bazı A için tüm A i = A nerede alınır .
7: USER-SC'ye (Cevap, A, sid) gönderin.
zincir içi işlem maliyetlerini en aza indirir. Bu protokol ayrıca oracle'a yapılan ödemeyi de içerir.
düğümler ve freeloader'lara yapılan ödemelere karşı güvence sağlar.
Orta vadeli strateji: Zincir dışı toplama. Sözleşme içi toplama var
önemli bir dezavantaj: Maliyet. Üzerinde iletme ve işleme maliyetini doğurur.
zincir O (n) oracle mesajları (A 1 , A 2 , ..., A n için taahhüt eder ve ifşa eder). İzinli
blok zincirleri, bu ek yük kabul edilebilir. On- ile izinsiz blokzincirlerde
Ethereum gibi zincir işlem ücretleri, eğer n büyükse, maliyetler engelleyici olabilir. Bir
daha uygun maliyetli bir yaklaşım, zincir dışı oracle yanıtlarını toplamak ve bir
CHAINLINK-SC A'ya tek bir mesaj. Bu yaklaşımın uygulanmasını öneriyoruz.
orta ila uzun vadede zincir dışı toplama.
Potansiyel olarak hatalı düğümler karşısında bir fikir birliği değeri A elde etme sorunu
blok zincirlerinin temelini oluşturan fikir birliği sorununa çok benzer. Verilen bir
önceden belirlenmiş kehanet kümesi, klasik bir Bizans Fayı Tol-
A.Klasik BFT protokollerini hesaplamak için erant (BFT) fikir birliği algoritması,
bir protokol çağrısının sonunda tüm dürüst düğümlerin aynı şeyi depolamasını sağlamayı hedefleyin
Örneğin bir blok zincirinde tüm düğümlerin aynı taze bloğu depoladığı değer. Bizim kehanetimizde
hedef biraz farklıdır. CHAINLINK-SC'nin (ve
daha sonra USER-SC) katılımcı olmadan toplu yanıt A = Agg (A 1 , A 2 , ..., A n) alır
fikir birliği protokolünde ve birden çok kişiden yanıt almaya gerek kalmadan
kahinler. Dahası, serbest yükleme sorununun hala ele alınması gerekmektedir.
ChainLink sistemi, eşik değeri içeren basit bir protokolün kullanılmasını önerir.
eski imzalar. Bu tür imzalar, herhangi bir sayıda imza kullanılarak gerçekleştirilebilir.
şemaları, ancak Schnorr imzaları [4] kullanılarak uygulanması özellikle basittir . Bunda
yaklaşımı, oracle'ların toplu bir açık anahtar pk ve buna karşılık gelen bir özel anahtara sahip
Bu O 1 , O 2 , ..., O n arasında a (t, n) eşik olarak paylaşılır [3]. Böyle bir paylaşım
her düğüm O i'nin ayrı bir özel / genel anahtar çiftine sahip olduğu anlamına gelir (sk i , pk i). O i can

---

Sayfa 15

Şekil 4: Sig sk [A], oracle'ların herhangi bir n / 2 + 1'i ile elde edilebilir.
kısmi bir imza oluştur σ i = Sig sk i
[A i] pk i'ye göre doğrulanabilir .
Bu kurulumun temel özelliği, aynı A değerindeki kısmi imzaların
tek bir geçerli toplu imza elde etmek için herhangi bir grupta toplanmalıdır
Σ = Bir yanıt A için Sig sk [A]. Bununla birlikte, hiçbir t - 1 oracle dizisi geçerli bir
herhangi bir değerde imza. Tek imza Σ böylece dolaylı olarak kısmi
en azından kehanetlerin imzaları.
Eşik imzaları, açıkça bir kümeden oluşmasına izin verilerek safça gerçekleştirilebilir.
bireysel düğümlerden gelen geçerli, bağımsız imzaların sayısı. Eşik imzaları var
bu naif yaklaşıma benzer güvenlik özellikleri. Ama önemli bir
zincir performansının iyileştirilmesi: Doğrulamanın boyutunu ve maliyetini bir
t faktörü.
Bu kurulumla, oracle'ların yalnızca kısmi
t Bu tür kısmi imzalar, Σ'nin hesaplanmasını sağlayana kadar imzalar. Yine de,
serbest yükleme sorunu ortaya çıkar. Bu nedenle, oracle'ların
A i'yi aldatmak ve kopyalamak yerine, belirlenen kaynaklardan veri elde edin .

başka bir kehanet. Çözümümüz bir finansal mekanizma içeriyor: Bir kuruluş PROVIDER (akıllı sözleşme olarak gerçekleştirilebilir) yalnızca orijinal verilerden elde edilen oracle'ları ödüllendirir
kısmi imzaları için.
Dağıtılmış bir ortamda, hangi oracle'ların ödeme için uygun olduğunun belirlenmesi zor olmak. Kahinler zincir dışı iletişim kurabilir ve artık bir günahımız yok.
gle yetkili tüzel kişiliği (CHAINLINK-SC) yanıtlar alıyor ve bu nedenle

---

Sayfa 16

uygun alacaklıları doğrudan katılan oracle'lar arasında belirleyebilme. Sonuç olarak, PROVIDER, kehanetlerin kendisinden yanlış davranışlara dair kanıt elde etmelidir.
güvenilmez olabilir. Fikir birliğine benzer mekanizmaların kullanımını öneriyoruz
ChainLink çözümümüzde PROVIDER'ın ücretsiz yükleme ücreti ödememesini sağlamak için kahinler.
ChainLink için önerdiğimiz zincir dışı toplama sistemi, beraberinde
Güvenlik geçirmez skeçler, Ek bulunabilir A. Bir dağıtılmış kullanır
serbest yüklemeye karşı direnç sağlayan eşik imzalarına dayalı protokol
f <n / 3 oracles. Serbest yüklemeye karşı direncin ilginç ve yeni bir teknik olduğuna inanıyoruz sorun.

5 ChainLink Güvenlik Hizmetleri

Bir önceki bölümde, ChainLink'te açıkladığımız protokoller sayesinde
hatalı oracle'lara kadar kullanılabilirliği ve doğruluğu sağlamayı önerir.
Ek olarak, Bölüm 6'da tartışıldığı gibi, güvenilir donanım aktif olarak değerlendirilmektedir
yanlış sağlayan bozuk oracle'lara karşı korumaya yönelik güvenli bir yaklaşım olarak
tepkiler. Bununla birlikte, güvenilir donanım, üç kişi için kesin koruma sağlamayabilir
nedenleri. İlk olarak, ChainLink ağının ilk sürümlerinde konuşlandırılmayacaktır.
İkinci olarak, bazı kullanıcılar güvenilir donanıma güvenmeyebilir (tartışma için Ek B'ye bakın).
Son olarak, güvenilen donanım, düğüm kesintilerine karşı koruma sağlayamaz, yalnızca düğüme karşı koruma sağlar
yanlış davranış. Bu nedenle kullanıcılar, en çok tercih edebileceklerinden emin olmak isteyeceklerdir.
Güvenilir oracle'lar ve USER-SC'nin hatalı oracle'lara güvenme olasılığını en aza indirin.
Bu amaçla, dört temel güvenlik hizmetinin kullanımını öneriyoruz: Doğrulama Sistemi,
bir İtibar Sistemi, bir Sertifika Hizmeti ve bir Sözleşme Yükseltme Hizmeti. Hepsi
bu hizmetler başlangıçta,
ChainLink ağı, ancak ChainLink'in ağına kesinlikle uygun şekilde çalışmak üzere tasarlanmıştır.
merkezi olmayan tasarım felsefesi. ChainLink'in önerilen güvenlik hizmetleri,
oracle düğüm katılımını engelleme veya oracle yanıtlarını değiştirme. Yalnızca ilk üç hizmet
Sözleşme Yükseltme Hizmeti tamamen
kullanıcılar için isteğe bağlı. Ek olarak, bu hizmetler, bağımsız
kullanıcıların sonunda sahip olabilmesi için katılımı teşvik edilmesi gereken sağlayıcılar
Aralarından seçim yapabileceğiniz birden fazla güvenlik hizmeti.

5.1 Doğrulama Sistemi

ChainLink Doğrulama Sistemi, zincir üzerindeki oracle davranışını izleyerek bir
Kullanıcıların oracle seçimine rehberlik edebilecek objektif performans ölçütü. Arayacak
oracle'ları izlemek için:

Sayfa 17

• Kullanılabilirlik: Doğrulama Sistemi, hataların giderilmesi için bir oracle tarafından kaydedilmelidir.
sorgulara zamanında cevap vermek. Devam eden çalışma süresi istatistiklerini derleyecektir.
• Doğruluk: Doğrulama Sistemi, görünen hatalı yanıtları kaydetmelidir
akranlar tarafından sağlanan yanıtlardan sapmalarla ölçülen bir oracle tarafından.[3]
ChainLink'teki ilk zincir üstü toplama sistemimizde, bu tür bir izleme

basittir, çünkü tüm oracle etkinliği CHAINLINK-SC tarafından görülebilir.
Bununla birlikte, ChainLink için tasarlanan zincir dışı toplama sisteminde,
toplama gerçekleştiren kahinlerin kendisidir. Sonuç olarak, CHAINLINK-SC
oracle yanıtlarına doğrudan görünmez ve kendi başına uygunluğu izleyemez
yetenek ve doğruluk.
Neyse ki, oracle'lar yanıtlarını dijital olarak imzalarlar ve bu nedenle, bir yan etki olarak, gen-
cevaplarına dair reddedilemez kanıtlar ortaya koyar. Önerdiğimiz yaklaşım bu nedenle
doğrulama hizmetini kahinleri ödüllendirecek akıllı bir sözleşme olarak gerçekleştirmek
sapan yanıtların kanıtlarını sunmak için. Başka bir deyişle, kahinler
görünüşte hatalı davranışları bildirmek için teşvik edildi.
Elbette oracle'lar,
cevap vermede başarısızlık. Bunun yerine, önerilen bir protokol iyileştirmesi oracle'ları
gerektirecektir.
diğerlerinden aldıkları yanıtlara dijital olarak onayları imzalamak
kahinler. Doğrulama sözleşmesi daha sonra aşağıdakilerin gönderimini kabul eder (ve tekrar
ödüllendirir)
bir altta yatan kişinin tutarlı yanıt vermediğini gösteren onay dizileri
akranlarına oracle oluşturuyor.
Hem zincir içi hem de zincir dışı durumlarda, kullanılabilirlik ve doğruluk istatistikleri
kahinler zincir üzerinde görünür olacaktır. Kullanıcılar / geliştiriciler böylece onları
görüntüleyebilecek
Ethereum'da bir Dapp veya bir
izinli bir blok zinciri için eşdeğer uygulama.
5.2 İtibar Sistemi
ChainLink için önerilen İtibar Sistemi, kullanıcı derecelendirmelerini kaydedecek ve yayınlayacaktır.
oracle sağlayıcıları ve düğümleri, kullanıcıların oracle performansını değerlendirmeleri için bir yol
sunar
bütünsel. Doğrulayıcı Sistem raporları, muhtemelen
oracle itibarları ve bu itibarları sağlam bir güven temeli üzerine yerleştirmek. Faktörler
zincir üzerindeki geçmişin ötesinde, oracle düğümü hakkında önemli bilgiler sağlayabilir
3 "Sapma", verilere özel bir şekilde tanımlanmalıdır. Basit mantıksal yanıtlar için - örneğin -
örneğin, bir uçuşun zamanında gelip gelmediği - sapma, basitçe,
çoğunluk. Örneğin, sensörler ve kaynaklar arasında yasal olarak değişiklik gösterebilen bir şehrin
sıcaklığı,
sapma, önemli sayısal sapma anlamına gelebilir. Tabii ki, çeşitli nedenlerle, örneğin, kırık sen-
Sors, iyi işleyen bir kahin bile çoğunluğun yanıtından sapabilir.
zaman.

Sayfa 18

güvenlik profilleri. Bunlar, kullanıcıların oracle markalarına olan aşinalıklarını içerebilir.
varlıklar ve mimariler. ChainLink İtibar Sisteminin şunları içermesini öngörüyoruz:
kullanıcıların derecelendirmelerinin diğer akıllı cihazlar için de mevcut olacağı temel bir zincir üstü
bileşen
referans sözleşmeleri. Ek olarak, itibar ölçütlerine kolayca erişilebilmelidir
daha büyük miktarda verinin verimli ve daha esnek bir şekilde işlenebileceği zincir dışı
ağırlıklı.
Belirli bir oracle operatörü için İtibar Sistemi başlangıçta destek olarak önerilmiştir.
Aşağıdaki ölçümleri, her ikisi de belirli atama türlerinin ayrıntı düzeyinde taşıma
(bkz.Bölüm 2) ve ayrıca genel olarak bir düğüm tarafından desteklenen tüm türler için:
• Atanan toplam istek sayısı: Bir
oracle kabul etti, hem yerine getirildi hem de yerine getirilmedi.
• Tamamlanan toplam istek sayısı: Bir
oracle yerine geldi. Bu, atanan istek sayısı üzerinden ortalaması alınabilir
tamamlanma oranını hesaplar.

• Kabul edilen toplam istek sayısı: Kabul edilen toplam istek sayısı
akran yanıtları ile karşılaştırıldığında sözleşmelerin hesaplanmasıyla kabul edilebilir olarak
değerlendirilir.
Bu, toplam atanan veya toplam tamamlanan isteklerin ortalaması alınabilir.
doğruluk oranları hakkında bilgi.
• Ortalama yanıt verme süresi: Oracle yanıtlarına zaman vermek gerekli olabilir
Onay için, yanıtlarının zamanında olup olmadığı belirlenmesinde yardımcı olacaktır.
gelecekteki zamanındalık. Ortalama yanıt süresi, tamamlananlara göre hesaplanır
istekleri.
• Ceza ödemelerinin miktarı: Ceza ödemeleri güvence altına almak için kilitlendiyse
bir düğüm operatörünün performansı, sonuç bir finansal metrik olacaktır.
oracle sağlayıcısının bir "çıkış dolandırıcılığı" saldırısına girmeme taahhüdü,
sağlayıcı kullanıcıların parasını alır ve hizmet sağlamaz. Bu metrik,
hem zamansal hem de finansal boyut içerir.
Yüksek itibara sahip hizmetler, herhangi bir pazarda düzgün davranmaya kuvvetle teşvik edilir.
doğrudan ve yüksek kullanılabilirlik ve performans sağlar. Olumsuz kullanıcı geri bildirimi ortaya
çıkacak
Yanlış davranışla ilişkili cezalar gibi marka değeri için önemli bir risk.
Sonuç olarak, iyi işleyen kahinlerin de-
iyi itibar ve iyi itibar geliştirmek,
yüksek performans.

Sayfa 19

5.3 Sertifikasyon Hizmeti
Doğrulama ve İtibar Sistemlerimiz geniş bir yelpazeye hitap etmeyi amaçlasa da
oracles tarafından hatalı davranışlar ve sistem bütünlüğünü sağlamanın bir yolu olarak önerilmiştir.
çoğu durumda, ChainLink ayrıca
bir Sertifikasyon Hizmeti. Amacı, nadir görülen ancak felaketle sonuçlanmayı önlemek ve / veya
düzeltmektir.
olaylar, özellikle de Sybil şeklinde hile yapma ve yansıtma saldırıları,
şimdi açıklıyoruz.
Sybil ve yansıtma saldırıları. Hem basit hem de sözleşme içi toplama protokolümüz
cols, dürüst düğümleri kopyalayan dürüst olmayan düğümler anlamında serbest yüklemeyi önlemeye
çalışır '
Yanıtlar. Ancak hiçbiri Sybil saldırılarına karşı koruma sağlamaz [9]. Bu tür saldırılar bir reklam
içerir
birden çok, görünüşte bağımsız oracle'ı kontrol eden versary. Bu düşman olabilir
oracle havuzuna hükmetme girişimi, f oracle'lardan daha fazlasının katılmasına neden olur
toplama protokolü ve stratejik zamanlarda yanlış veriler sağlamak, örneğin
Yüksek değerli sözleşmelerde büyük işlemleri etkiler. Hile yapan oracle sayısı,
aynı zamanda sadece tek bir düşmanın kontrolü altında değil, aynı zamanda gizli anlaşma yoluyla da
ortaya çıkar.
birden fazla düşman arasında. F oracle'ları içeren saldırılar veya hatalar özellikle
yalnızca zincir içi davranıştan tespit edilememeleri bakımından zararlıdır.
Ek olarak, operasyonel maliyetleri azaltmak için, bir Sybil saldırganı bir davranış benimseyebilir
yansıtma olarak adlandırılır, burada oracle'ların verilere dayalı olarak bireysel yanıtlar göndermesine
neden olur
tek bir veri kaynağı sorgusundan elde edilir. Başka bir deyişle, yanlış davranan kahinler
verileri zincir dışı paylaşır, ancak verileri bağımsız olarak kaynakladığını iddia eder. Yansıtma
faydaları ve
yanlış veri göndermeyi seçip seçmemesi konusunda rakip. Çok daha az ciddi
güvenlik tehdidi, veri sahteciliğinden daha fazladır, ancak güvenliği biraz düşürür.
belirli bir kaynağa yönelik çeşitli sorgulardan kaynaklanan hata düzeltmesini ortadan kaldırır
Src . Örneğin, https://www.datasource.com, örneğin,

ara sıra tetiklenen bir hata, birden fazla sorgulayıcı yine de doğru bir çoğunluğu elde edebilir sonuç.
Yanlış veri, yansıtma ve genel olarak gizli anlaşma ile sonuçlanan Sybil saldırıları, uzun vadeli stratejimizde güvenilir donanım kullanımıyla ortadan kaldırılmıştır (bkz. Bölüm 6) .
Sertifikasyon Hizmeti tasarımı. ChainLink Sertifikasyon Hizmeti, tespit etmek ve önlemeye yardımcı olmak için genel bütünlük ve kullanılabilirlik güvencesi sağlamak kısa ve orta vadede oracle yeter sayılarını yansıtma ve bir araya getirme. The Certifica- Hizmet, yüksek kaliteli oracle sağlayıcılarının onaylarını yayınlayacaktı. Vurguluyoruz yine, yukarıda belirtildiği gibi, hizmet sağlayıcıları yalnızca kullanıcıların yararı için derecelendirecektir.
Oracle düğüm katılımını veya sisteme katılmamayı dikte etmesi amaçlanmamıştır. Sertifikasyon Hizmeti, or- acle dağıtım ve davranış. Doğrulama Sistemi istatistiklerini izler

---

Sayfa 20

oracles üzerinde ve zincir üzerindeki yanıtlarda post-hoc nokta kontrolünü gerçekleştirin - özellikle yüksek değerli işlemler - bunları doğrudan temsilciden alınan cevaplarla karşılaştırarak kullanılabilir veri kaynakları. Oracle sağlayıcısının verileri için yeterli talep olması durumunda, Oracle sağlayıcılarının zincir dışı denetimlerini haklı çıkarmak için yeterli ekonomik teşvik olması, aşağıdaki ilgili kontroller gibi ilgili güvenlik standartlarına uygunluğun doğrulanması Cloud Security Alliance (CSA) Cloud Controls Matrix [26] ve ayrıca oracle'ların kaynağının uygun denetimlerini yaptıklarına dair yararlı güvenlik bilgileri ve akıllı sözleşmeleri için bayt kodu.
İtibar ölçümlerine ek olarak, otomatikleştirilmiş zincir içi ve otomatikleştirilmiş kapalı dolandırıcılık tespiti için zincir sistemleri, Sertifikasyon Hizmeti bir araç olarak planlanmıştır Zincir içi sistemleri otomatikleştiren Sybil saldırılarını ve diğer suistimalleri tespit etmek için olumsuz. Örneğin, tüm düğümler ayın yeşil peynirden yapıldığını kabul ederse, USER-SC'nin bu yanlış gerçeği yutmasına neden olabilir. AY BİLEŞENLERİ = {YEŞİL PEYNİR} ancak blok zincirine kaydedilecek ve post-hoc incelemede görülebilecek.

5.4 Sözleşme Yükseltme Hizmeti

Son zamanlardaki akıllı sözleşme saldırılarının gösterdiği gibi, kurşungeçirmez akıllı sözleşmeleri kodlamak, son derece zorlu egzersiz [1], [20], [22]. Akıllı bir sözleşme olsa bile doğru programlanmış, çevresel değişiklikler veya hatalar yine de savunmasız kalmaya neden olabilir. bağlar, örneğin [2] .
Bu nedenle, bir Sözleşme Yükseltme Hizmeti öneriyoruz. Bu kullanımı vurguluyoruz bu hizmetin tamamı isteğe bağlıdır ve kullanıcıların denetimindedir.
Kısa vadede, güvenlik açıkları keşfedilirse, Sözleşme Yükseltme Hizmeti ChainLink'te yeni bir destekleyici oracle sözleşmeleri seti oluşturacaktır.
Yeni oluşturulan talep eden akıllı sözleşmeler daha sonra yeni kümeye taşınabilecek oracle sözleşmelerinin.
Ne yazık ki, yine de, mevcut olanlar, potansiyel olarak eski ile takılıp kalacaktı. savunmasız küme. Bu nedenle, uzun vadede CHAINLINK-SC bir bayrağı destekleyecektir (MIGFLAG) oracle çağrılarında, bir çağrı olup olmadığını belirten sözleşme talebinden yeni bir CHAINLINK-SC'ye iletilmelidir. Ayarlamak varsayılan olarak (yani, bayrak eksikse) false, MIGFLAG istek yapmayı etkinleştirir otomatik yönlendirmeden yararlanacak sözleşmeler ve dolayısıyla yeni sürüme geçiş CHAINLINK-SC. Yönlendirmeyi etkinleştirmek için bir kullanıcı, ChainLink isteklerini MIGFLAG = true ile yayınlamak için sözleşme. (Kullanıcılar kendi akıllı sözleşmeler, böylece bir talimat aldıktan sonra bu bayrağı değiştirirler yetkili bir sözleşme yöneticisinden zincir üzerinde.)
Kullanıcıların yeni oracle sözleşmelerine geçişi bir tür "kaçış yolu" işlevi görür. bir mekanizma olarak blok zinciri araştırmacıları tarafından uzun süredir savunulan bir şey (bkz., ör. [23])

hataları düzeltmek ve saldırıları düzeltmek için
20

Sayfa 21

whitehat hacking [1] veya hard forklar. Güncellenen sözleşmelere geçiş görünür olacak
blok zincirinde ve kullanıcıların yükseltmeden önce gözden geçirmesi için denetlenebilir.
Yine de, bazı kullanıcıların herhangi biriyle rahat hissetmeyeceğinin farkındayız
göç / yönlendirme şeklinde bir kaçış kapısını kontrol eden grup. Zorla
geçiş, geçiş sözleşmesinin denetleyicisini veya bir bilgisayar korsanı
oracle'ı değiştirmek gibi kötü niyetli faaliyetlerde bulunmak için ilgili kimlik bilgilerini vaat ediyor
tepkiler. Bu nedenle, talepte bulunan sözleşmeler, for-
koruma özelliği ve böylece kaçış kapısı aktivasyonunu devre dışı bırakabilir. Ek olarak
ChainLink'in ademi merkeziyetçiliğe odaklanmasına uygun olarak, sağlayıcıların
topluluk tarafından geliştirilen CHAINLINK-SC'nin birden çok sürümünü destekleyebilme.
5.5 LINK belirteci kullanımı
ChainLink ağı, ChainLink Node operatörlerine ödeme yapmak için LINK jetonunu ** kullanır
zincir dışı veri akışlarından verilerin alınması, verilerin blok zincirine biçimlendirilmesi için
okunabilir formatlar, zincir dışı hesaplama ve çalışma süresi garantileri, op-
erler. Ethereum gibi ağlarda bir akıllı sözleşmenin ChainLink kullanması için
düğüm, seçtikleri ChainLink Düğüm Operatörüne LINK jetonlarını kullanarak ödeme yapmaları
gerekecek,
Zincir dışı kaynak talebine göre fiyatlar düğüm operatörü tarafından belirlenir
ChainLink, ve diğer benzer kaynakların tedarikini sağlar. BAĞLANTI
ken, ek ERC223 "aktar ve ara" işlevselliğine sahip bir ERC20 belirtecidir
transferin (adres, uint256, bayt), tokenlerin alınıp işlenmesine izin verir.
tek bir işlemdeki sözleşmeler.
6 Uzun Vadeli Teknik Strateji
Bu tanıtım belgesinde önerilen ChainLink için uzun vadeli teknik strateji şunları içerir:
üç temel yön: Oracle gizliliği, altyapı değişiklikleri ve zincir dışı şirket
varsayım.
6.1 Gizlilik
Dağıtılmış bir oracle ağı, arızalara karşı yüksek derecede koruma sağlamayı amaçlamaktadır.
kahinler. Çoğu dağıtım senaryosunda,
f Bizans faylarının yüzü (basit toplama protokolümüzde f <n / 2 için). Güvenilir
donanım çok daha fazlasını sunabilir ve güvenlik için daha iyi bir yaklaşım olarak önerilmiştir
ChainLink ağı. Güvenilir donanım, Town Crier'ın (TC) temel taşıdır
oracle [24], şu anda Ethereum ana ağında [33] çalışan ve
içerik oluşturucular, TC lansmanında SmartContract ile ortaklık kurdu.
21

Sayfa 22

Bazı güvenilir donanım biçimleri, özellikle Intel'in son Yazılım Koruması
eXtensions (SGX) komut seti mimari uzantıları kümesi [12] - [15], [18], arama
dağıtılmış güven biçimlerine güçlü bir ek sağlamak. Kısaca, SGX, bir
iki kritik olduğunu iddia eden bir yerleşim bölgesi olarak adlandırılan bir ortamda yürütülecek uygulama
güvenlik özellikleri. İlk olarak, enklavlar uygulamanın bütünlüğünü korur, yani
diğer süreçler tarafından bozulmaya karşı verileri, kodu ve kontrol akışı. İkincisi, bir
enclave, bir uygulamanın gizliliğini korur, yani verileri, kodu,
ve yürütme durumu diğer işlemlere karşı opaktır. SGX mahsur kalmışlığı korumaya çalışıyor
uygulamalar kötü niyetli bir işletim sistemine karşı ve dolayısıyla
bir uygulamanın çalıştığı ana bilgisayarın yöneticisi.
ARM TrustZone gibi alternatif güvenilir donanım biçimleri,
bir süredir var olan SGX, bunlardan yoksun ek bir anahtar özellik sağlar.

teknolojileri. Bir platformun, bir
belirli uygulama (hash durumunun bir yapısı ile tanımlanır). Bu tasdik,
uzaktan doğrulanabilir ve belirli bir uygulama örneğinin herkese açık olarak bağlanmasına izin verir
anahtar ve böylece diğer taraflarla doğrulanmış ve gizli kanallar kurmak.
Bir mahallede bir oracle çalıştırmak ve onayları dağıtmak,
oracle'ın özellikle belirli bir uygulamayı yürüttüğüne dair güçlü güvence
ChainLink ekosistemindeki geliştiriciler tarafından oluşturulan veya onaylanan biri. Bunlara ek olarak,
HTTPS aracılığıyla bir veri kaynağına bağlanabilen bir mahallede çalışan bir oracle,
aldığı verilerin tahrif edilmediğine dair güçlü bir güvence sağlar.
(Ayrıntılar için [24], [33] 'e bakın.) Bu özellikler,
veri bozulması, Sybil saldırıları vb. anlamında oracle yanlış davranışı.
Yine de daha büyük bir fırsat, güvenilir donanımın
güçlü bir gizlilik sağlar. Gizlilik ihtiyacı genel olarak aşağıdakilerden biridir:
Blockchain dağıtımının önündeki ana engeller. Gizliliği koruyan oracle'lar,
problemi çözmede yardımcı olur.
Neden dağıtılan oracle'lar gizliliği sağlamaz. Gizlilik eğlencelidir.
Herhangi bir oracle sisteminde elde edilmesi çok zor. Bir oracle bir blockchain cephesine sahipse
akıllı bir sözleşme gibi sonlandırılırsa, oracle'a yönelik herhangi bir sorgu kamuya açık olacaktır.
ible. Sorgular zincir üzerinde şifrelenebilir ve oracle hizmeti tarafından şifresi çözülebilir, ancak
o zaman oracle hizmetinin kendisi onları görecek. Güvenli gibi ağır araçlar bile
Şifrelenmiş veriler üzerinden hesaplamaya izin veren çok partili hesaplama çözülemiyor
mevcut altyapı göz önüne alındığında bu sorun. (Uygulama odaklı bir uygulama için bkz. Ör. [11]
perspektif.) Bir noktada, bir sunucunun bir hedef veri kaynağına bir sorgu göndermesi gerekir.
sunucu. Bu nedenle, sorgunun gizliliği ne olursa olsun sorguyu görmelidir.
daha önce zevk aldım. Ayrıca sorguya verilen yanıtı da görecektir.

Sayfa 23

SGX aracılığıyla güvenliliği koruyan oracle'lar. SGX kullanan bir oracle,
ve özünde bütünlük için güvenilen bir TTP gibi davranarak verileri bir mahfaza içinde işleyin
ve gizlilik. Başlangıç olarak, böyle bir oracle, kendi içindeki sorguların şifresini çözebilir.
yerleşim bölgesi. Daha sonra bunları başka herhangi bir işleme maruz bırakmadan işleyebilir (veya
herhangi bir insan). Enklav ayrıca kaynaklardan gelen verileri gizli bir şekilde işleyebilir
ve kullanıcı kimlik bilgileri gibi hassas bilgileri güvenli bir şekilde yönetebilir.
yetenek, aşağıda gösterdiğimiz gibi.
Town Crier sistemi, gizli uçuş veri sorgularını destekler. Uçuş bilgisi
mation, kamu tarafından şifrelenmiş bir TC akıllı sözleşmesi ön ucuna geçirilebilir
TC hizmetinin anahtarı. TC sorgunun şifresini çözer ve ardından bir veri kaynağıyla iletişim kurar (ör.
flightaware.com) HTTPS üzerinden. Sorgulayan akıllı sözleşmeye basit bir
evet / hayır "Bu uçuş rötar yaptı mı?" sorusuna cevap ve başkalarını açığa çıkarmaz
zincir üzerindeki bilgiler.
Daha da ilginç bir TC yeteneği, Steam'de alım satım desteğidir.
oyun platformu. TC, kullanıcı kimlik bilgilerini (şifreleri) güvenli bir şekilde alıp,
oyun sahipliği bir alıcıdan bir satıcıya aktarılmıştır. Böylece yaratabilir
Yüksek güvenceli adil, aksi takdirde ulaşılamayacak güvenli bir pazar yeri
dijital ürünler için kripto para takası. (Bunun aksine, basit bir dağıtılmış kahin,
kullanıcıların şifrelerini onlar adına güvenli bir şekilde yönetemedi.)
TC aynı zamanda birden fazla kaynaktan gelen verilerin güvenilir zincir dışı toplanmasını da
gerçekleştirebilir,
birden fazla kaynaktan gelen veriler üzerinden güvenilir hesaplamanın yanı sıra (örneğin, ortalama) ve
veri kaynaklarının etkileşimli sorgulanması (örneğin, bir kaynağın veritabanında arama
başka birinin cevabına cevap).
Güvenilir donanım, blok zincirlerinin ölçeklenebilir kullanımına heyecan verici yeni bir yaklaşım
sunar [24] ,
[29], akıllı sözleşmeler dahil olmak üzere blockchain altyapısının büyük bölümlerinin,

enklavlarda yürütmek. Böyle bir mimari, şeffaflık faydalarını birleştirir.
zincir dışı yürütmenin gizlilik özelliklerine sahip ve güvenilir blok zincirleri
donanım. Benzer fikirler başka teknikler kullanılarak önerilmiş olsa da, örneğin
zk-SNARKs [21], güvenilir donanım çok daha pratiktir (ve daha az karmaşıktır). bizim
mevcut araştırma gündemi, katalizör olarak oracle'larla birlikte bu geniş vizyonu içermektedir.
hizmet.
Bir güven kaynağı olarak Intel konusunu Ek B'de kısaca tartışıyoruz .
SGX verilen güvenliği tanımlama. Güvenilir donanım kullanımı göz önüne alındığında mümkündür,
Intel SGX'in biçimciliğinden başlayarak oracle doğruluğunu daha resmi olarak tanımlamak için
[32] 'de önerilmiştir . Bu biçimcilik, SGX'in küresel bir Evrensel olarak ele alınmasını sağlar
Tertip edilebilir (UC), [6] işlevselliği, F = SGX (Σ SGX-) [PROG Encl , R] '. Burada ve sonrasında
Σ imzalama ve doğrulama işlevlerine sahip bir imza şemasını belirtir Σ. İmza ve
Σ. Doğrula. F bir örneği SGX (Σ SGX) [PROG Encl , R] 'grubu imza parametreli

---

Sayfa 24

şema Σ sgx . Prog encl argümanı , bir enklavda çalışan programı belirtir, yani,
donanım tarafından korunan ortam. R, üzerinde çalışan güvenilmeyen kodu gösterir.
bir SGX ana bilgisayarı, yani yerleşim bölgesinde çalışan uygulamayı çağıran yazılım.
Şekil 5 ([24] 'den alınmıştır) F sgx işlevselliğinin çalışmasını gösterir . Başlangıçta
= Prog tialization, bu Outp çalışır Encl kodu üreten ve tasdik) (.Initialize
prog ait encl ve outp. Bir onaylama σ att , platform tarafından dijital olarak imzalanmış bir beyandır.
Bu prog Encl bir yerleşim bölgesi içinde çalışan ve çıkış outp vermiştir. Tipik kullanımda,
PROG Encl .Initalize () oluşturmak için kullanılabilecek bir örneğe özgü ortak anahtar üretir
uygulama örneğine güvenli bir kanal. (İd, params) ile bir devam çağrısı üzerine,
F SGX yürütmeye devam eder ve PROG sonucunu verir Encl .Resume (id, parametreler),
id, bir oturum tanımlayıcısını belirtir ve parametreler, prog encl'ye girilen parametreleri belirtir .
F SGX [PROG Encl , R] ': SGX için soyutlama
Sabit kodlu : sk sgx (Σ sgx için özel anahtar)
Varsayalım: prog encl Initialize ve Resume giriş noktalarına sahip
Başlat:
Alındığında (init) R'den:
= Prog: outp Let Encl) (.Initalize
σ att : = Σ SGX .Sign (sk SGX- , (Prop Encl , outp))
Çıktı (çıkış, σ att)
Devam et:
Alındığında (özgeçmiş, id, parametreler) R'den:
Outp edelim: = prog Encl .Resume (id, parametreler)
Çıkış çıkışı
Şekil 5: SGX özelliklerinin bir alt kümesini yakalayan SGX uygulaması için resmi soyutlama.
Şekil 5 formalizm göz önüne alındığında, tam da bütünlüğünü tanımlamak mümkündür
bir kehanet. Tanım 1 , [24] 'te verilen tanımın hafif bir genellemesiyle bunu yapar.
buna Oracle Authenticity diyoruz.
Tanım 1 (Oracle'ın Orijinalliği). Bir oracle O çalışan bir program olduğunu söylüyoruz.
PROG Encl F kullanılarak SGx ve örnek anahtar pk çıkış O tatmin olmadığını Oracle Orijinallik,
F sgx ile keyfi olarak etkileşime girebilen herhangi bir polinom-zaman rakibi A için , A,
dürüst bir doğrulayıcının veriyi kabul etmesini sağlayın (pk O , σ att , params: = (url ,, T), data, σ)
değil T zamanında kamu anahtarı ile url içeriği (prog Encl .Resume (id, parametreler) içinde

---

Sayfa 25

bizim modelimiz). Daha resmi olarak, herhangi bir olasılıksal polinom zamanlı rakibi A için,
Pr
□

□
□
□
(pk O , σ att , id, parametreler, veri, σ) ← A F sgx (1 λ):
(Σ SGX- .Verify (pk SGX- , σ att , (Prop Encl , pk O)) = 1) ∧
(Σ.Verify (pk O , σ, (id, parametreler, veri)) = 1) ∧
Veri = PROG Encl .Resume (id, parametreler)
□
□
□
□
≤ negl (λ),
güvenlik parametresi için λ.

6.2 Altyapı değişiklikleri

Güvenli oracle'lar oluşturmadaki zorlukların çoğu, mevcut
veri kaynakları, sundukları verileri dijital olarak imzalamazlar. Yaptılarsa, kahinler
verileri kurcalamaktan kaçınmak için güvenilir olması gerekmez. HTTPS, protokol
güvenli web iletişimi, veri imzalamayı etkinleştirmez. Bununla birlikte, bir
Sunucuların sertifikalara sahip olmasını gerektiren temeldeki genel anahtar altyapısı (PKI)
bu prensipte veri imzalamayı destekleyebilir.
Bu gözlem, HTTPS'ye izin veren bir TLS uzantısı olan TLS-N'nin temelidir.
sunucular, oturumlarının bölümlerini istemcilerle imzalayacak. Seçici doğası
imzalama, müşterilerin siteden hariç tutması gibi başka güzel özellikler de sağlar.
imzalı transkriptler ve böylece kimlik bilgilerinin gizliliğini korur (ör. şifreler)
sunuculara bağlanmak için kullanırlar.
TLS-N gibi altyapı değişikliklerinin gelecek vaat eden yaklaşımlar olduğuna inanıyoruz.
Oracle güvenliğini destekliyor. Muhtemelen başkalarıyla birlikte kullanılmaları gerekecek
SGX gibi teknolojiler, ancak aşağıdaki sınırlamalar nedeniyle:
1. Altyapı değişiklikleri: Ne yazık ki, TLS-N bir
standart olarak, veri kaynakları bunu istemcilerin yararlanabilmesi için açıkça dağıtmalıdır. Birkaç veri kaynakların yakın gelecekte olması muhtemeldir.
2. Toplama ve hesaplama: TLS-N, toplama veya diğer
veri kaynaklarından gelen veriler üzerinden güvenilir hesaplama biçimleri, bu nedenle bazıları güvenilir
Bu görevleri yerine getirmek için yine de mekanizma gerekli olacaktır.
3. Maliyet: TLS-N imzalı verilerin doğrulanması, nispeten yüksek zincir içi maliyetlere neden olur basit imza doğrulama ile karşılaştırıldığında.
4. Gizlilik: TLS-N bant dışı gizli yönetimi destekleyemez
kullanıcı kimlik bilgileri veya sorguları, ancak bunun yerine kullanıcıların bir veri kaynağını sorgulamasını gerektirir
bu amaç için kendileri. Örneğin, gizli uçuş bilgileri,
bir web sitesinin daha sonra gizli otomatik sorgulanması için akıllı bir sözleşmede saklanabilir.

Sayfa 26

6.3 Zincir dışı hesaplama

Kimlik bilgilerine bağlı API'lerin kullanımı gibi oracle'ların bazı ilgi çekici kullanımları,
bir oracle veri iletmekten çok daha fazlasını yapar. Yönetmesi gerekebilir
kimlik bilgileri, verileri kazımak için hesaplarda oturum açma vb. Gerçekten, gerçekten güven verildiğinde-
değerli ve gizli kahinler, Town Crier'de SGX destekli sistemler
sıfır bilgi ispatı [21] gibi teknikler , sınırların elde edilmesine yardımcı olabilir.
oracle'lar ve akıllı sözleşmeler arasında akıcı hale gelebilir.
ChainLink, kullanıcıları etkinleştiren sorgular için zaten normal ifade tabanlı bir dili desteklemektedir.

zincir dışı verilerin işlenmesini esnek bir şekilde belirtmek için. Bununla birlikte, uzun vadeli stratejimiz,
oracle'ların kullanılan anahtar zincir dışı hesaplama kaynağı olduğu bir dünya yaratmaya çalışıyor çoğu akıllı sözleşmeye göre. Bunun bir hedefe doğru inşa edilerek sağlanacağına inanıyoruz. sonuçları olan oracle'lar içinde tamamen genel, özel zincir dışı hesaplama modeli akıllı sözleşmeler tarafından tüketilir. Bu, bizim gibi güçlü bir güvenlikle başarılabilirse, İnanın, pahalı ve hassas hesaplama mantığını oracle'lara itmek, daha iyi gizlilik, daha düşük sözleşme yürütme maliyetleri ve daha esnek mimariler.

7 Mevcut Oracle Çözümleri

ChainLink, akıllı cihazlarda yeni oracle teknolojisine yönelik yaygın bir ihtiyacı karşılamak için tasarlanmıştır.
sözleşme sistemleri. Maalesef bugün çok sınırlı miktarda yüksek güvenlikli ve esnek oracle sistemleri. Bu güvenilir kahinlerin yokluğunun önemli olduğuna inanıyoruz. akıllı sözleşmelerin gelişimine engel.

Günümüzde oracle hizmetleri için en yaygın kullanılan seçenek, merkezi oracle'dır sağlayıcılar. Bu yaklaşım, merkezi bir kontrol noktası oluşturduğu için sorunludur, ve bu nedenle, güvenilmez akıllı olan yüksek kurcalamaya karşı dayanıklılık standartlarını karşılamıyor sözleşmeler gerektirir. Bu türden bazı sistemler, örneğin [31], bu sorunu çözmek için Doğru davranışı "kanıtlamak" için noter onayına güvenmek. Noter tasdik hizmetlerinin bu şekilde kullanılması bu hizmetlerle ilgili belgelenmiş sorunlar [37] ve onaylarının zincir üzerinde uygun şekilde doğrulanamayacağını ve bunun sonucunda bir (potansiyel olarak özyinelemeli) daha fazla doğrulama ihtiyacı.

Güvenilir oracle verilerini sağlamaya yönelik bir başka yaklaşım, manuel insan kaynaklarına güvenmektir.
yapılandırılmamış verilerin girişi. Bu "manuel giriş oracle" genellikle tahmin piyasalarında kullanım [17] , [25], [28]. Uygun finansal riskler yaratarak ve hile yapmak için sınırlı mali teşviklere sahip ekonomik olarak rasyonel oyuncular varsayarak, bu tür kahinler, doğru kitle kaynaklı yanıtlar için yüksek bir güvence sağlar. Bu yaklaşım merkezi olmayan ve esnektir. Manuel giriş oracle'ları yanıtlarını insanlar, yapılandırılmış verilerin zor olduğu sorulara yanıt verebilirler. güvenilir bir şekilde bulmak veya çıkarmak zor, örneğin doğal dil işleme gerektirir

---

Sayfa 27

haber olayları. Ne yazık ki, insan bilişi maliyetli ve yavaş olduğu için, manuel giriş oracle'ları kaynak yoğun olup gerçek zamanlı değildir ve yalnızca bir herhangi bir zamanda sınırlı soru seti. ChainLink'in de olabileceğine inanıyoruz tahmini piyasa sözleşmelerini hızlı ve otomatik olarak çözmek için çok kullanışlıdır. yapılandırılmış verilerle çözülebilir.

Son bir yaklaşım, kaynaktaki verilerin biçimini değiştirmektir. Bir veri kaynağı dijital Sağladığı verileri tally imzaladı, ardından geçiş sunucusuna güvenilmesi gerekmeyecekti. USER-SC, aldığı verilerdeki imzaları basitçe kontrol edebilir. Mükemmel, genel bu tür bir yaklaşım, yukarıda tartışıldığı gibi TLS-N tarafından sağlanır. Maalesef daha önce bahsedildiği gibi, TLS-N mevcut altyapıda değişiklik yapılmasını gerektirir.

8 Sonuç

Akıllı sözleşmeler için merkezi olmayan bir oracle ağı olan ChainLink'i tanıttık blok zincirinin dışındaki kaynaklarla güvenli bir şekilde etkileşim kurmak için. Ana hatlarıyla Hem zincir içi hem de zincir dışı bileşenleri açıklayan ChainLink mimarisi. Tanımladıktan sonra Oracle bağlamında güvenlik, ChainLink'in çok katmanlı yaklaşımını anlattık. ademi merkeziyetçilik. Koruma gibi yeni özelliklere sahip yeni bir protokol önerdik serbest yüklemeye karşı (kağıtta ek protokoller ve güvenliğe dayanıklı eskizler ile)

ek). Ayrıca, ChainLink'in teknolojik gelişmelerden nasıl yararlanabileceğine dair bir yol haritası hazırladık.
ve güvenilir donanım ve verilerin dijital olarak imzalanması gibi altyapısal gelişmeler kaynaklar. Son olarak, mevcut oracle çözümlerini ve eksikliklerini inceledikten sonra, ChainLink gibi bir sisteme olan ihtiyacı bugün ortaya çıkardık.
Tasarım ilkeleri. ChainLink üzerinde çalışmalarımıza devam ederken, öncekilere bakacağız- Aşağıdaki temel değerleri belirleyin:
• Güvenli ve açık sistemler için ademi merkeziyetçilik. Ademi merkeziyet sadece blok zincirlerinin kurcalamaya dayanıklı özelliklerinin temeli, ancak bunların temeli izinsiz doğa. Merkezi olmayan sistemler kurmaya devam ederek, ekosistem içinde izinsiz gelişimi daha da mümkün kılmak. İnanıyoruz ademi merkeziyetin küresel olarak gelişen bir ekosistem için çok önemli bir bileşen olduğunu uzun vadeli sürdürülebilirlik ile.
• Basit, esnek sistem tasarımı için modülerlik. Felsefesini takdir ediyoruz bir şeyi iyi yapan küçük araçlar oluşturmak. Basit bileşenler kolaylıkla yapılabilir mantıklı ve böylece güvenli bir şekilde daha büyük sistemlerle birleştirildi. İnanıyoruz modülerlik yalnızca yükseltilebilir sistemleri mümkün kılmakla kalmaz, aynı zamanda merkezi olmayan
izasyon. ChainLink'in kilit parçalarının bağlı olduğu veya onlar tarafından yönetildiği her yerde

---

Sayfa 28

birkaç taraf, rekabete izin veren bir ekosistem tasarlamaya çalışacağız kullanılacak uygulamalar.
• Güvenli, genişletilebilir sistemler için açık kaynak. ChainLink, birçok açık kaynak projesinin omuzlarında duruyor. Biz topluma değer veriyoruz nity ve ChainLink'i açık kaynak olarak geliştirerek katkıda bulunmaya devam edecek tavır. Geliştiriciler, akademisyenler ve güvenlik görevlileri ile sürekli olarak etkileşim kurmayı planlıyoruz.
akran değerlendirmesi için zengin uzmanlar. Test, denetim ve resmi kanıtları teşvik ediyoruz. güvenlik, tümü sağlamlığı ve güvenliği olan bir platform oluşturmak amacıyla gelecekteki yenilikleri destekleyebilir.
Bu ilkeleri göz önünde bulundurarak, erişim ve etkiyi genişletmeyi dört gözle bekliyoruz oracle'ları güvenli bir dönüm noktası haline getirerek blok zincirlerinin ve akıllı sözleşmelerin ekosistem.
Referanslar
[1] Parite. Multi-sig hack: Bir ölüm sonrası. https://blog.ethcore.io/the-multi-sig-hack-a-postmortem /. 20 Temmuz 2017.
[2] Gun Sirer. Çapraz Zincir Tekrar Saldırıları. Hacking, Dağıtılmış blog. 17 Temmuz 2016.
[3] Adi Shamir. "Bir sır nasıl paylaşılır". İçinde: ACM 22.11 İletişimleri (1979), s. 612–613.
[4] Claus-Peter Schnorr. "Akıllı kartlarla verimli imza oluşturma". In: Journal of cryptology 4.3 (1991), s. 161–174.
[5] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, et al. "Güvenli dağıtılmış ayrık günlük tabanlı şifreleme sistemleri için anahtar üretimi ". İçinde: Eurocrypt. Cilt 99. Springer. 1999, s. 295–310.
[6] R. Canetti. "Evrensel Olarak Oluşturulabilir Güvenlik: Kripto için Yeni Bir Paradigma-grafik Protokoller ". İçinde: FOCS. 2001.
[7] Ran Canetti. "Evrensel olarak oluşturulabilir güvenlik: Kripto için yeni bir paradigma grafik protokoller ". In: Foundations of Computer Science, 2001. Proceedings. 42. IEEE Sempozyumu. IEEE. 2001, s. 136–145.
[8] Douglas R Stinson ve Reto Strobl. "Makul bir şekilde güvenli dağıtılmış Schnorr imzası-türler ve örtük sertifikalar için bir (t, n) eşik şeması ". İçinde: ACISP. Cilt 1. Springer. 2001, sayfa 417–434.
[9] John R Douceur. "Sybil saldırısı". In: Uluslararası Eşler Arası Çalıştayı

Eş Sistemler. Springer. 2002, s. 251–260.

Sayfa 29

[10] Aniket Kate ve Ian Goldberg. "İnternet için dağıtılmış anahtar üretimi". İçinde: Dağıtılmış Hesaplama Sistemleri, 2009. ICDCS'09. 29. IEEE Uluslararası Konferans devam ediyor. IEEE. 2009, s. 119–128.
[11] Claudio Orlandi. "Çok partili hesaplama pratikte herhangi bir işe yarıyor mu?" İçinde: Acoustikler, Konuşma ve Sinyal İşleme (ICASSP), 2011 IEEE International Conference on. IEEE. 2011, s. 5848–5851.
[12] Ittai Anati, Shay Gueron, Simon Johnson, vd. "Yenilikçi teknoloji CPU tabanlı tasdik ve mühürleme ". In: 2. Uluslararası Bildiriler Güvenlik ve Gizlilik için Donanım ve Mimari Destek Çalıştayı. Cilt 13. 2013. url: https: / / software. intel. com / tr - bize / makaleler / cpu tabanlı tasdik ve mühürleme için yenilikçi teknoloji (vis- 05/23/2016 tarihinde yayınlanmıştır).
[13] Matthew Hoekstra, Reshma Lal, Pradeep Pappachan, vd. "Yenilikçi Kullanmak Güvenilir Yazılım Çözümleri Oluşturma Yönergeleri ". In: Tutanaklar 2Nd International Workshop on Hardware and Architectural Support for Se- Güvenlik ve Gizlilik. HASP '13. Tel-Aviv, İsrail: ACM, 2013, 11: 1–11: 1. isbn: 978-1-4503-2118-1. doi : 10.1145 / 2487726.2488370 . url: http: //doi.acm. org / 10.1145 / 2487726.2488370.
[14] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, et al. "Yenilikçi eğitim- İzole çalıştırma için yazılım ve yazılım modeli. " In: 2. Bildiriler Uluslararası Güvenlik için Donanım ve Mimari Destek Çalıştayı ve Gizlilik. 2013, s. 10. url: http://css.csail.mit.edu/6.858/2015/ okumalar / intel-sgx.pdf (23.05.2016 tarihinde ziyaret edildi).
[15] Intel. Intel Yazılım Koruma Uzantıları Programlama Referansı. 2014. (Ziyaret 05/23/2016).
[16] Gavin Wood. "Ethereum: Güvenli bir merkezi olmayan genelleştirilmiş işlem defteri". In: Ethereum Projesi Sarı Kağıt (2014).
[17] Jack Peterson ve Joseph Krug. "Augur: merkezi olmayan, açık kaynaklı bir platform tahmin piyasaları için ". İçinde: arXiv ön baskı arXiv: 1501.01042 (2015).
[18] Victor Costan ve Srinivas Devadas. "Intel SGX Açıklaması". İçinde: Kriptoloji ePrint Arşivi (2016). url: https: / / eprint. iacr. org / 2016/086. pdf (05/24/2016 tarihinde ziyaret edildi).
[19] Victor Costan, Ilia A Lebedev ve Srinivas Devadas. "Sanctum: Minimal Zor- Güçlü Yazılım İzolasyonu için ware Uzantıları. " İçinde: USENIX Security Symposium. 2016, sayfa 857–874.

Sayfa 30

[20] Kevin Delmolino, Mitchell Arnett, Ahmed Kosba, vd. "Adım adım Güvenli bir akıllı sözleşme oluşturma: Bir kripto para biriminden dersler ve bilgiler laboratuar ". In: Uluslararası Finansal Kriptografi ve Veri Güvenliği Konferansı. Springer. 2016, s. 79–94.
[21] Ahmed Kosba, Andrew Miller, Elaine Shi, vd. "Hawk: Blockchain modeli kriptografi ve gizliliği koruyan akıllı sözleşmeler ". Giriş: S & P'16. IEEE. 2016.
[22] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, et al. "Akıllı sözleşmeleri daha akıllı yapmak". İçinde: 2016 ACM SIGSAC Bilgisayar ve Bilgisayar Konferansı Bildirileri iletişim Güvenliği. ACM. 2016, s. 254–269.
[23] Bill Marino ve Ari Juels. "Zekayı değiştirmek ve geri almak için standartlar belirleme sözleşmeler ". İçinde: Kurallar ve Kural Biçimlendirme Dilleri Uluslararası Sempozyumu

Anlamsal Web için. Springer. 2016, s. 151–166.
[24] Fan Zhang, Ethan Cecchetti, Kyle Croman, ve diğerleri. "Kasaba Crier: Bir yetkili akıllı sözleşmeler için özel veri beslemesi ". İçinde: 2016 ACM SIGSAC Tutanakları Bilgisayar ve İletişim Güvenliği Konferansı. ACM. 2016, s. 270–282.
[25] Augur proje sayfası. https://augur.net. 2017.
[26] CSA Bulut Kontrolleri Matrisi. URL: https://cloudsecurityalliance.org/group/cloud-kontrol matrisi. 2017.
[27] Mark Flood ve Oliver Goodenough. Otomat Olarak Sözleşme: Computa-Mali Sözleşmelerin Genel Temsili . https: //www.financialresearch . gov / çalışma kağıtları / dosyalar / OFRwp-2015-04_Contract-as-Automaton-The-Computational-Representation-of-Financial-Agreements.pdf. Nın-nin-Finansal Araştırma Müdürlüğü, 2017.
[28] Gnosis proje sayfası. https://gnosis.pm. 2017.
[29] Hyperledger Sawtooth. https://intelledger.github.io/introduction.html. 2017.
[30] Abhiram Kothapalli, Andrew Miller ve Nikita Borisov. "SmartCast: Bir In-Akıllı Sözleşmelerin Kullanıldığı Yüzde Yüz Uyumlu Konsensüs Protokolü ". İçinde: Finans Kriptografi ve Veri Güvenliği (FC). 2017.
[31] Oraclize proje sayfası. http://www.oraclize.it. 2017.
[32] Rafael Pass, Elaine Shi ve Florian Tramer. "Onaylananlar için biçimsel soyutlamalar yürütme güvenli işlemciler ". İçinde: Eurocrypt. Springer. 2017, s. 260–289.
[33] Town Crier Ethereum hizmeti. http://www.town-crier.org/. 2017.

Sayfa 31

[34] Florian Tramer, Fan Zhang, Huang Lin, et al. "Mühürlü cam provaları: Trans-bilgiyi kanıtlamak ve satmak için ebeveyn yerleşim bölgeleri ". İçinde: Güvenlik ve Gizlilik (Eu-roS & amp; P), 2017 IEEE Avrupa Sempozyumu. IEEE. 2017, s. 19–34.
[35] Vitalik Buterin ve ark. Ethereum teknik raporu . https://github.com/ethereum/wiki / wiki / White-Paper.
[36] JSON Şeması. http://json-schema.org/.
[37] Hubert Ritzdorf, Karl Wüst, Arthur Gervais, vd. TLS-N: Reddetmeme üzerinden TLS Aracıların Kesilmesi için Her Yerde Bulunan İçerik İmzalamasını Etkinleştirme. IACR ePrint raporu 2017/578. URL: https://eprint.iacr.org/2017/578.

Sayfa 32

Açıklamalar
† Ari Juels, Cornell Tech'teki Jacobs Enstitüsü'nde öğretim üyesidir. Ortak yazardı bu, SmartContract ChainLink Ltd. için bir danışman olarak ayrı sıfatıyla çalışmaktadır. finansal bir çıkarı var.
* Bu teknik inceleme, SmartContract ChainLink, Ltd. ("SCCL") tarafından sağlanmaktadır, ChainLink Platformunu destekleyen bir İngiliz Virgin Adaları şirketi. Güvenli Asset Exchange, Inc. ("SAE") dba SmartContract.com, yönetim sağlar, SCCL'nin LINK Token satışının desteği dahil olmak üzere SCCL'ye teknik ve diğer destek, ve SCCL'den tazminat alıyor. Teknolojik, sosyal ve ticari Blockchain teknolojisini kullanan yapılar sürekli olarak gelişiyor ve gelişecek öngörülebilir gelecek için. Buna göre planlar, stratejiler ve uygulama tanımları Bu teknik incelemede açıklanan kuyruklar büyük olasılıkla gelişecek ve buna göre asla evlat edinilmek. SCCL ve SAE, ek veya al-ChainLink ile ilişkili üçlü planlar, stratejiler veya uygulama ayrıntıları Platform.
** LINK Tokenları, Şartlar ve Koşullar uyarınca SCCL tarafından satılmaktadır. Token satış koşulları https://link.smartcontract.com/terms adresinde mevcuttur. Tamamlamak için

ayrıntılar, şartları gözden geçirin. LINK Tokenları menkul kıymetler, yatırımlar veya para birimi değildir,
bu şekilde satılmaz veya pazarlanmaz. Ayrıca: satışa katılmak önemli
teknolojik ve sistemik riskler oluşturamaz; satış ikamet eden kişilere açık değildir
Amerika Birleşik Devletleri veya Kanada vatandaşları. Satış dönemi, süresi, fiyatlandırma,
ve diğer hükümler, jeton satış şartlarında belirtildiği gibi değişebilir. BAĞLANTI
satış, bilinen ve bilinmeyen riskleri, belirsizlikleri ve diğer faktörleri içerir.
LINK Token'ların gerçek işlevselliğinin, yardımcı programının veya kullanım seviyelerinin önemli olmasına neden olur
öngörülen gelecekteki sonuçlardan, kullanımdan, işlevlerden veya yardımcı programdan farklı olarak veya
SCCL tarafından ifade edilen terimler.

Sayfa 33

Zincir Dışı Bir Toplama
Hem geçerli, imzalı yanıtları sağlamak hem de serbest yüklemeyi önlemek için zincir dışı toplamamız
Bölüm 4.2'de tartışılan gation protokolü, basit bir dağıtılmış protokole dayanacaktır.
eşik imzalarına dayalı [8] . Bu yaklaşımın yararı, belirli bir
sorgu, tek bir imza bir dizi n oracle düğüm tarafından zincir dışı oluşturulabilir.
Sonuç olarak, bunun yerine yalnızca tek bir kimliği doğrulanmış mesajın zincir üzerinde işlenmesi gerekir.
farklı oracle düğümlerinden gelen O (n) mesajlarının sayısı. Bu yaklaşım, maliyetleri büyük ölçüde düşürür.
Algoritma 1'de ortaya çıkanlarla karşılaştırma 1. Fikir daha da genişletilebilir,
gibi [30], tek bir eşik içinde birden fazla sorgu cevaplarını bir araya getirmek için
imza, burada keşfetmediğimiz ancak mimarimizde dikkate alabileceğimiz bir fikir.
F <n / 3 oracle'larının hatalı olduğunu ve t = f + 1 olduğunu varsayalım. Hatalı düğümler deneyebilir
serbest yükleme ve / veya diğer dürüst olmayan davranışlardan herhangi birini gerçekleştirmek için, örneğin
geçersiz cevapların imzalanması.
Zincir dışı toplama için eksiksiz protokolümüz bir çift algoritmadan oluşur
OCA = (DistOracle, RewardOracles) A = değerinde bir imza Sig sk [A] hesaplamak için
Agg (A 1 , A 2 , ..., A n), çoğunluk işlevi için Agg. Basitleştirilmiş, tek aşamalı bir versiyon
Bu protokollerden biri, sırasıyla Algoritmalar 2 ve 3'te gösterilmiştir. Birincisi yürütülür
katılan oracle'lar tarafından, ikincisi ise PROVIDER tüzel kişiliği tarafından yürütülür.
bu, yukarıda belirtildiği gibi akıllı bir sözleşme şeklini alabilir.
OCA'yı sunmadan önce, Schnorr imzaları ve
bunları hesaplamak için eşik şeması [8] 'de verilmiştir .
Schnorr imzaları. Schnorr imza şeması, bir G grubunu kullanır.
ayrık log probleminin varsayıldığı g jeneratörlü asal mertebesi p
zor ol. Bir kullanıcının anahtar çifti, x ← $ Z için (sk, pk) = (x, y = g x) biçimini alır
×
p , nerede
Z
×
p = Z p −0. Genelliği kaybetmeden, grup işlemlerini çarpımsal olarak ifade ederiz.
Schnorr imzaları eliptik eğri grupları üzerinden hesaplanabilir ve aslında tipik olarak
modern kripto uygulamalarında. Şekil 6 , Schnorr imza şemasını göstermektedir.
Eşik Schnorr imza şeması. Eşik imzasını kullanıyoruz
[8] şeması . Bu şema, küresel bir özel / genel anahtar çifti (sk, pk) oluşturur. O
istenen bir mesaj m için bir tam imza Sig sk [m] eşik üretimini mümkün kılar .
İlk anahtar oluşturma protokolünde, her oyuncuya bir x i = sk i anahtar paylaşımı atanır .
Bu kurulum tek seferlik bir işlemdir ve dağıtılmış anahtar oluşturma kullanılarak yapılabilir.
protokol, ör. [5] veya zaman uyumsuz bir ayar için [10].

Bir imza oluşturmak için, oyuncular (bizim ortamımızdaki oracle'lar) önce bir dağıtılmış kurulumda anahtar paylaşımı oluşturmada olduğu gibi anahtar oluşturma protokolü. Bunun çıktısı protokol genel bir geçici gizli anahtardır e. Her oyuncu (O i) bir (gizli) pay alır
e i of e.

Sayfa 34

Schnorr imzası: İmzalayan girişi (m, sk = x); Doğrulayıcı girişi pk = y
İmzalayan
Doğrulayıcı
r ← $ Z
x
p
e ← g r
c = H (m || e)
s = cx + e
(e, s), m
−−−− →
c = H (m || e)
g s? = ry c
Şekil 6: Schnorr imza şeması.
Geçici anahtar e verilen O i'nin kısmi imzası, σ biçimini alır.
(e)
ben
= cx i + e i ,
burada c = H (m || e), tam imzadaki gibi. Her oyuncu için O i de var
kısmi imzasını doğrulayan geçerli bir i (σ i ; (pk, e)) işlevi . Kısmi atıfta bulunuyoruz
için geçerli olarak imza O ı geçerli altında doğru doğrularsa i .
Gösterimi biraz değiştirdik ve şemayı büyük ölçüde yoğunlaştırdık.
burada sergi. Detaylar için okuyucuya [8] başvururuz .
A.1 OCA protokolü
Şimdi OCA'yı oluşturan DistOracle ve RewardOracles algoritmalarını sunuyoruz.
Bunlar, aşağıda Algoritmalar 2 ve 3'te belirtilmiştir.
Algoritma 1'de olduğu gibi, tanıkları açıkça dahil etmektense) Commit'in bir com-
azaltma işlevi.
Tüm oyuncuların CHAINLINK-SC tarafından alınan mesajları açık olduğu gibi görebileceğini unutmayın.
Zincir. CHAINLINK-SC'ye gönderilen ilk geçerli imza Σ ∗ olsun . Alg. 3, izin verdik
PS * kimin kısmi imzaları SAĞLAYICISINDAN tarafından alınan decommitments kümesi belirtmek
verim Σ ∗ . (PS ∗ , Σ ∗ gönderen kehanetten gelebilir , ancak gerekli değildir. Herhangi bir kehanet
PS'de kısmi imzayla ∗ , PS ∗ göndermeye teşvik edilir .)

Sayfa 35

Algoritma 2 DistOracle (f, n, i, sk i = x i , pk, Src) (O i için kod)
Birlikte geçici anahtar oluşturun:
1: Dağıtılmış anahtar üretme protokolünü yürütün ve alın (e i , e).
Verileri elde edin:
2: A elde edilir i dan Src .
Kısmi imza oluşturun:
3: σ hesaplayın
(e)
ben
(= cx i + e i , c = H (m || e) için, burada m = A i).

Kaydetme turu:
4: Yayın taahhüdü comm i = (Kaydetme (σ
(e)
i , A i); i).
5: Farklı oracle'lardan geçerli taahhütlerden oluşan bir set C i alınana kadar bekleyin .
6: C i'yi PROVIDER'a gönderin.
Turu hazırlayın:
7: Yayın hazırlandı.
8: Farklı hazırlanmış mesajlar alınana kadar bekleyin.
Turu aç / kapat:
9: (σ
(e)
i , A i) iletişim için i .
10: Bir set PS'si olan geçerli degerlendirmeler alınana kadar bekleyin.
Tam imza hesaplaması:
11: CHAINLINK-SC tarafından henüz geçerli bir Σ alınmadıysa
12:
PS'deki kısmi imzaları Σ = Sig sk [A] olarak toplayın .
13:
CHAINLINK-SC'ye Σ gönderin.
14:
PS'yi PROVIDER'a gönderin.
15: eğer bitirmek

Sayfa 36

Algoritma 3 RewardOracles (PROVIDER kodu)
1: Farklı oracle'lardan n − f taahhüt kümelerinin (C i) ve PS ∗ ' ların C setine kadar bekleyin.
Alınan.
2: Her kehanet için O j do
3:
C'deki σ j ∈ PS ∗ ve> 2f kümeleri O j'den σ j'ye taahhütler içeriyorsa , o zaman
4:
Gönder $ ödül için Ç j .
5:
eğer biterse
6: için son
A.2 Kanıt eskizleri
OCA'nın temel özelliklerinin kanıt taslaklarını sunuyoruz. Bunu en fazla varsayımla gösteriyoruz
Hatalı düğümler, protokol her zaman doğru bir cevapta geçerli bir imza oluşturur,
ve serbest yükleme oracle düğümlerini asla ödüllendirmez.
İddia 2. OCA protokolü asla bir serbest yükleme düğümünü ödüllendirmeyecektir.
Kanıt. (Taslak) O z'nin serbest yüklendiğini varsayalım . Daha sonra geçerli bir com-
azaltma yalnızca τ zamanından sonra , Adımda ilk dürüst O i'nin teslim aldığı zaman
Alg. 2 . O i en az n - 2f gelen n - f hazırlanmış mesajlar aldım
dürüst düğümlerden. Let Ç j anlamında olabildikleri en azından bunlardan biri n - dürüst düğümler
2f. O j
hazırlanmış bir mesaj gönderdikten sonra taahhütleri artık kabul etmeyeceğinden, C j
Bu tür dürüst düğümünün O j böylece artık süresi x sonra değişiklik ve C olacak j gözlenecektir
doğru bir kısmi İmza σ bir bağlılık doğruya z den O z . Bu nedenle, en fazla
Alg'de C'de n - (n - 2f) = 2f küme. 3O z'yi içerecektir . Böylece O z bir
ödül.
Ne yazık ki, OCA, serbest yüklenmeyen düğümlerin ödenmesini garanti edemez. Bir hile-
Düşman, bir taahhüdün kaldırılmasını aldıktan sonra, Adım 13'teki dürüst düğümleri acele edebilir.
Alg. 2 kendi kısmi imzalarını oluşturarak ve yalnızca bir dürüst düğüm dahil ederek

Σ üretiminde kısmi imza. Bu dürüst düğümün taahhüdü olmayabilir
herhangi bir düğüm tarafından toplanan n - f arasına dahil edilmiştir. Gelebilirdi
sonrasında.
İddia 3. OCA her zaman geçerli bir imza ile sonuçlanacaktır Σ = Sig sk [A] sonunda
CHAINLINK-SC'ye gönderildi.
Kanıt. (Taslak) n − f dürüst düğüm vardır ve f <n / 3, dolayısıyla> 2f ≥ f +1 vardır
dürüst düğümler ve dolayısıyla en azından t dürüst düğümler. Yani Alg'in 1. Adım. 2 tamamlanacak
başarıyla.
Benzer şekilde, n − f dürüst düğüm olduğu için, her kahin sonunda tamamlayacaktır
Alg. Adım 7 2 , hazırlanmış bir mesaj gönderme. Dürüst düğümler sonunda

---

Sayfa 37

en az n - f hazırlanmış mesajlar alın ve 13. Adım'a izin verecek şekilde geri çekilecektir.
bazı dürüst düğümler tarafından tamamlanacak.
İddia 4. CHAINLINK-SC tarafından oca'da alınan herhangi bir geçerli imza Sig sk [A],
geçerli değer A.
Kanıt. (Taslak) Geçerli bir imza Sig sk [A] 'nın doğru bir değer içerdiğini görmek kolaydır
A. Sig sk [A] ' yı ve en çok f <t düğümlerini hesaplamak için kısmi imzalar gerektiğinden
hatalı, dürüst bir düğüm tarafından A üzerinde en az bir kısmi imza sağlandı ve
bu nedenle doğru olması gerekir.
A.3 Tartışma
OCA, burada kısaca tartıştığımız birkaç tasarım zorluğunu tanıtır.
Dürüst düğümler için ödeme. Ne yazık ki, serbest yükleyicileri cezalandırırken
Ödeme yapılmadığında, OCA, dürüst düğümlere ödeme yapılacağını tam tersine garanti edemez.
Aslında, hiçbir düğümün hatalı olmadığı iyi huylu durumda bile, şanssız mesaj sıralaması
kısmi imzalara katkıda bulunan dürüst düğümlere neden olabilir Σ almama
ödeme.
Bu problem kısmen Alg yapılarak çözülebilir. 2 senkron. Specifi-
cally, "Bekle" adımları, düğümlerin bir süre beklemesini gerektirebilir.
Dürüst düğümlerden gelen mesajların oranı garantilidir. Bu durumda, kısmi olan tüm düğümler
Σ ile birleştirilen imzalar ödeme garantili olacaktır. Ancak yan etkiler
daha yavaş yürütme ve Δ 'yi doğru şekilde ayarlama zorluğu olacaktır.
Güçlü ödeme garantili bir asenkron protokol tasarlama sorunu-
tees, şu anda araştırmakta olduğumuz açık bir araştırma problemidir.
Gereksiz mesajlar. OCA, zincir içi iletişimi ve ide-
ally, yalnızca bir zincir üzeri mesaj içerir, yani imzalı bir yanıt response,
katılan kahinler. Ancak uygulamada, çünkü bir imza Σ anında gönderilmeyecektir
blok zincirine gün geçtikçe, birden fazla oracle bağımsız olarak imzalı yanıtlar gönderebilir
blok zincirine. Bu tür gereksiz mesajları sınırlamanın en iyi yolu oracle'ların
sadece blok zincirini değil, mempool'u, yani bekleyen mesajları izleyin.
Anahtar yönetimi. Elbette, çoğu zaman olduğu gibi, anahtar yönetimi büyük
bu tür protokollerde zorluk. Hisse senetlerinin dağıtımı yapılabilir
dağıtılmış bir şekilde [5] ve yeni düğümleri barındırmak için güncellemeler yapılabilir
ve çıkış yapan düğümleri kaldırmanın yanı sıra proaktif güvenlik, yani devam eden
düğümlerin anahtarlarının tehlikeye atılmasına karşı dayanıklılık. Ek olarak, düğümler organize
edilebilir

Sayfa 38

n boyutunu sınırlandırmak için farklı klikler halinde. ChainLink'in bunları kullanmasını öneriyoruz
esnek, duyarlı ve güvenli bir dağıtılmış oracle sağlamak için teknikler.
B SGX Güven Varsayımları
Intel'in daha güçlü doğruluk güvencesi sağlama rolünde, SGX geliştirir ancak

ChainLink'teki diğer bütünlük korumalarının yerini almaz. Başka bir deyişle, SGX kullanımı
sistemi kesinlikle daha güçlü hale getirir.
Gizlilik için SGX'e güvenmek Intel'e güvenmeyi gerektirir, ancak bu güven doludur
cumscribed. Intel'in CPU'larında bir arka kapıya sahip olmadığını varsayarsak,
kapalı alan verilerinin sızması, enklav durumunu denetlemek için bir araca sahip değildir. (Böyle
bir arka kapı mümkündür, ancak bu, üzerinde fiziksel kanıtların bulunmasını gerektirir.
her kullanıcının makinesi ve ciddi bir itibar riski oluşturur.)
Intel veya Intel'in üretim süreçlerini bozan bir rakip, baskıda
birden fazla sahtecilik onaylama anahtarları (platform EPID anahtarları). Böyle bir düşman yaratabilir
SGX özellikli sunuculara gömülü olmayan ancak bunun yerine denetimlere izin veren EPID
anahtarları
SGX olmayan bir platformda oluşturulacak tasyonlar. Gerçekte, bu düşman yaratabilir
Geçerli görünen SGX doğrulamaları oluşturan, ancak hiçbir koruma sağlamayan sahte sunucular
gizli kod için tions. Elbette bu tür düğümlerden daha fazlası varsa,
düşman oracle yanıtlarını bozabilir. Daha sorunlu bir şekilde, bu tür düğümler
oracle düğümleri tarafından işlenen hassas verileri düşmana ifşa edin. Fal-
sify EPID anahtarları, mevcut, geçerli SGX'i bozma yeteneği anlamına gelmez.
örnekler.
Elbette bugün, hoşumuza gitse de gitmese de, bunu kabul etmek de önemlidir.
Intel'e güven kaçınılmazdır. Bunu okuduğunuz makinedeki CPU
kağıt bu gerçeğe tanıklık ediyor - ya da değilse, kullandığınız sunucudaki CPU
bu kağıdı indirdi.
Elbette, güvenilir donanımın birden çok cihazdan kullanılması tercih edilir.
satıcılar ve diğerlerinin eşdeğer yetenekler yaratacağı umulmaktadır. Yeni,
güvenilir donanım için açık mimariler ve güven varsayımlarını zayıflatma yolları
bu tür donanımlar için gerekli olan aktif araştırma alanlarıdır, örneğin [19] , [34]. Yeteneği
Satıcılar veya mimariler arasında çeşitlendirmek, veri gizliliğini sağlamaz,
ancak.
Ayrıca, gizlilik güvencesi için teknikler üzerine araştırma yapmakla da ilgileniyoruz.
kapak trafiğinin kullanımı yoluyla dağıtılmış ağlar.