arXiv: 1607.01341v9 [cs.CR] 26 Mayıs 2017
CEZAYIR
*
Jing Chen
Bilgisayar Bilimleri Bölümü
Stony Brook Üniversitesi
Stony Brook, NY 11794, ABD
jingchen@cs.stonybrook.edu
Silvio Micali
CSAIL
MIT
Cambridge, MA 02139, ABD
silvio@csail.mit.edu
Öz
Halka açık bir defter, herkes tarafından okunabilen ve artırılabilen kurcalamaya dayanıklı bir veri dizisidir.
Halka açık defterlerin sayısız ve zorlayıcı kullanımı vardır. Açıkça her türlü güvenliği sağlayabilirler işlemlerin - başlıklar, satışlar ve ödemeler gibi - tam olarak gerçekleştikleri sırayla.
Halka açık defterler yalnızca yolsuzluğu engellemekle kalmaz, aynı zamanda çok karmaşık uygulamaları da mümkün kılar.
kripto para birimleri ve akıllı sözleşmeler. Demokratik bir toplum biçiminde devrim yaratmaya hazırlar
çalışır. Ancak şu anda uygulandığı gibi, kötü ölçekleniyorlar ve potansiyellerine ulaşamıyorlar.
Algorand, halka açık bir defteri uygulamanın gerçekten demokratik ve verimli bir yoludur. Öncekinin aksine
iş kanıtına dayalı uygulamalar, ihmal edilebilir miktarda hesaplama gerektirir ve
ezici bir çoğunlukla yüksek olasılıkla "çatallaşmayan" bir işlem geçmişi oluşturur.
Algorand, (yeni ve süper hızlı) mesaj ileten Bizans anlaşmasına dayanmaktadır.
Somutluk için, Algorand'i yalnızca bir para platformu olarak tanımlayacağız.
1. Giriş
Para giderek sanal hale geliyor. Amerika Birleşik Devletleri'nin yaklaşık% 80'inin
dolar bugün yalnızca genel muhasebe girişleri olarak mevcuttur [5]. Diğer finansal araçlar da aynı şeyi yapıyor.
Evrensel olarak güvenilen merkezi bir varlığa güvenebileceğimiz ideal bir dünyada,
olası tüm siber saldırılara karşı, para ve diğer finansal işlemler yalnızca elektronik olabilir.
Ne yazık ki böyle bir dünyada yaşamıyoruz. Buna göre, merkezi olmayan kripto para birimleri,
Bitcoin [29] ve Ethereum gibi "akıllı sözleşme" sistemleri önerilmiştir [4]. Şurada:
bu sistemlerin kalbi, bir dizi işlemi güvenilir bir şekilde kaydeden paylaşılan bir defterdir,
* Bu ikinci yazar [ile arXiv kağıt daha resmi (asenkron) versiyonu 24] , bir kâğıt
kendisi Gorbunov ve Micali'ye [18] dayanmaktadır . Algorand'ın teknolojileri aşağıdakilerin amacıdır
patent başvuruları: US62 / 117,138 US62 / 120,916 US62 / 142,318 US62 / 218,817 US62 / 314,601
PCT / US2016 / 018300
US62 / 326,865 62 / 331,654 US62 / 333,340 US62 / 343,369 US62 / 344,667 US62 / 346,775 US62 / 351,011 US62 / 653,482
US62 / 352,195 US62 / 363,970 US62 / 369,447 US62 / 378,753 US62 / 383,299 US62 / 394,091 US62 / 400,361 US62 / 403,403
US62 / 410,721 US62 / 416,959 US62 / 422,883 US62 / 455,444 US62 / 458,746 US62 / 459,652 US62 / 460,928 US62 / 465,931

ödemeler ve sözleşmeler kadar çeşitli, kurcalamaya dayanıklı bir şekilde. Tercih edilen teknoloji

böyle bir kurcalamaya karşı korumanın blockchain olduğunu garanti eder. Blok zincirler aşağıdaki gibi uygulamaların arkasındadır:
kripto para birimleri [29] , finansal uygulamalar [4] ve Nesnelerin İnterneti [3]. Birkaç teknik blockchain tabanlı defterleri yönetmek için önerildi: işin kanıtı [29], hissenin kanıtı [2], pratik Bizans hata toleransı [8] veya bazı kombinasyonlar.
Ancak şu anda, defterleri yönetmek verimsiz olabilir. Örneğin, Bitcoin'in çalışma kanıtı yaklaşımı (orijinal [14] konseptine dayalı) büyük miktarda hesaplama gerektirir, savurgan ve kötü ölçekleniyor [1]. Buna ek olarak, fiilen gücü çok az elinde yoğunlaştırır.
Bu nedenle, halka açık bir defterin uygulanması için yeni bir yöntem öne sürmek istiyoruz. güvenilir ve dokunulmaz bir otorite tarafından yönetilen merkezi bir sistemin rahatlığı ve verimliliği, mevcut merkezi olmayan uygulamaların verimsizlikleri ve zayıflıkları. Yaklaşımımızı diyoruz Algorand, çünkü şimdiye kadar oluşturulan defteri esas alarak, seçmek için algoritmik rastgelelik kullandığımız için,
bir sonraki geçerli işlem bloğunu oluşturmaktan sorumlu bir dizi doğrulayıcı. Doğal olarak bu tür seçimlerin manipülasyonlara karşı kanıtlanabilir bir şekilde bağışık olmasını ve şu tarihe kadar öngörülemez olmasını sağlıyoruz
son dakika, ama aynı zamanda nihayetinde evrensel olarak net olduklarını da.
Algorand'ın yaklaşımı, ne prensipte ne de fiili olarak
farklı kullanıcı sınıfları yaratır (Bitcoin'de "madenciler" ve "sıradan kullanıcılar" olarak). Algorand'da "tümü
güç, tüm kullanıcılar kümesinde bulunur ”.
Algorand'ın dikkate değer bir özelliği, işlem geçmişinin yalnızca çok küçük
olasılık (örneğin, trilyonda bir, yani 10-18). Algorand ayrıca bazı yasal
ve politik endişeler.
Algorand yaklaşımı, blok zincirleri ve daha genel olarak herhangi bir oluşturma yöntemi için geçerlidir.
kurcalamaya dayanıklı bloklar dizisi. Aslında yeni bir yöntem ortaya koyuyoruz — alternatifi ve blok zincirlerinden daha verimli - bağımsız çıkarlar olabilir.
1.1 Bitcoin'in Varsayımı ve Teknik Sorunları
Bitcoin çok zekice bir sistemdir ve çok sayıda sonraki araştırmaya ilham kaynağı olmuştur. Yine de aynı zamanda sorunludur. Bunun altında yatan varsayımı ve teknik sorunları özetleyelim.
aslında Bitcoin gibi iş kanıtı üzerine kurulu tüm kripto para birimleri tarafından paylaşılıyor.
Bu özet için, Bitcoin'de bir kullanıcının birden fazla genel anahtara sahip olabileceğini hatırlamak yeterlidir.
paranın açık anahtarlarla ilişkilendirildiğini ve ödemenin bir dijital imza şeması olduğunu
Bir genel anahtardan diğerine bir miktar para aktaran dijital imza. Esasen,
Bitcoin, işlenmiş tüm ödemeleri , her biri birden çok bloktan oluşan B_1 , B_2 , ...
Herhangi bir sırayla alınan tüm B_1 ödemeleri , ardından B_2'nin ödemeleri herhangi bir sırayla gelecek şekilde ödemeler ,
vb. bir dizi geçerli ödeme oluşturur. Her blok ortalama olarak her 10 dakikada bir oluşturulur.
Bu blok dizisi bir zincirdir, çünkü herhangi bir değişikliğin, hatta herhangi bir değişikliğin yapılmasını sağlayacak şekilde yapılandırılmıştır.
tek bir blokta, sonraki tüm bloklara süzülerek, herhangi bir değişikliği tespit etmeyi kolaylaştırır.
ödeme geçmişi. (Göreceğimiz gibi, bu, her bloğa bir kriptografik
Bir öncekinin hash'i.) Böyle bir blok yapıya blok zinciri denir.
Varsayım: Hesaplama Gücünün Dürüst Çoğunluğu Bitcoin'in kötü niyetli olmadığını varsayar.
varlık (ne de koordine edilmiş kötü niyetli kuruluşlardan oluşan bir koalisyon), hesaplama işlemlerinin çoğunu kontrol eder.
blok üretimine ayrılmış güç. Aslında böyle bir varlık, blok zincirini değiştirebilir,

---

ve böylece ödeme geçmişini istediğiniz gibi yeniden yazın. Özellikle ödeme yapabilir $\wp$, ödenen yardımları elde edin ve ardından herhangi bir izini “silin”.
Teknik Sorun 1: Hesaplamalı Atık Bitcoin'in engellemeye yönelik çalışma kanıtı yaklaşımı

nesil olağanüstü miktarda hesaplama gerektirir. Şu anda sadece birkaç yüz
sistemdeki binlerce genel anahtar, en güçlü 500 süper bilgisayar yalnızca toplayabilir
Bitcoin oyuncularının ihtiyaç duyduğu toplam hesaplama gücünün yalnızca yüzde 12,8'i. Bu
önemli ölçüde daha fazla kullanıcı sisteme katılırsa, hesaplama miktarı büyük ölçüde artacaktır.
Teknik Problem 2: Günümüzde Güç Konsantrasyonu, aşırı miktardaki
sıradan bir masaüstü kullanarak yeni bir blok oluşturmaya çalışan bir kullanıcı için hesaplama gerekli (bir
cep telefonu), para kaybetmeyi bekliyor. Aslında, sıradan bir bilgisayarla yeni bir bloğu hesaplamak için,
Hesaplamaya güç sağlamak için gerekli elektriğin beklenen maliyeti beklenen ödülü aşıyor.
Yalnızca özel olarak oluşturulmuş bilgisayar havuzlarını kullanarak ("yeni bloklar kazmaktan" başka bir şey yapmayan)
yeni bloklar oluşturarak kar elde etmeyi bekleyebilir. Buna göre, bugün fiilen iki
ayrık kullanıcı sınıfları: yalnızca ödeme yapan sıradan kullanıcılar ve özel madencilik havuzları, sadece yeni bloklar arar.
Bu nedenle, son zamanlarda olduğu gibi, blok için toplam hesaplama gücünün olması şaşırtıcı olmamalıdır.
nesil sadece beş havuzda yatıyor. Bu tür durumlarda, çoğunluğun
hesaplama gücü dürüsttür, daha az güvenilir hale gelir.
Teknik Problem 3: Belirsizlik Bitcoin'de blockchain mutlaka benzersiz değildir. Aslında
en son kısmı genellikle çatallanır: blok zinciri —say— B 1 , ..., B k , B / olabilir.
k + 1 , B /
k + 2'ye göre
bir kullanıcı ve B 1 , ..., B k , B //
k + 1 , B //
k + 2 , B //
başka bir kullanıcıya göre k + 3 . Ancak birkaç bloktan sonra
zincire eklendiğinde, ilk k + 3 bloğun aynı olacağından emin olabilir misiniz?
bütün kullanıcılar için. Bu nedenle, son blokta yer alan ödemelere hemen güvenilemez.
zincir. Bloğun suyun içinde yeterince derin olup olmadığını beklemek ve görmek daha akıllıcadır.
blok zinciri ve dolayısıyla yeterince kararlı.
Ayrı olarak, yasa uygulama ve para politikası endişeleri de Bitcoin hakkında gündeme geldi.[1]

1.2 Özetle Algorand

Algorand ayarı çok zor bir ortamda çalışıyor. Kısaca
(a) İzin Verilmeyen ve İzin Verilen Ortamlar. Algorand bile verimli ve güvenli bir şekilde çalışıyor
tamamen izinsiz bir ortamda, keyfi olarak çok sayıda kullanıcının
herhangi bir soruşturma veya herhangi bir izin olmaksızın herhangi bir zamanda sistem. Elbette Algorand çalışıyor
izin verilen bir ortamda daha da iyi.

1 Bitcoin ödemeleri tarafından sunulan (sözde) anonimlik kara para aklama ve / veya finansman için kötüye kullanılabilir.
Suçlu şahısların veya terör örgütlerinin. Prensipte mükemmel olan geleneksel banknotlar veya altın külçeler
anonimlik, aynı zorluğu oluşturmalıdır, ancak bu para birimlerinin fizikselliği, parayı önemli ölçüde yavaşlatır
kolluk kuvvetleri tarafından bir dereceye kadar izlemeye izin vermek için transferler.
"Para basma" yeteneği, bir ulus devletin en temel güçlerinden biridir. Prensip olarak, bu nedenle, büyük
Bağımsız dalgalı bir para biriminin benimsenmesi bu gücü azaltabilir. Ancak şu anda Bitcoin olmaktan çok uzak
hükümetin para politikalarına yönelik bir tehdit ve ölçeklenebilirlik sorunları nedeniyle asla olmayabilir.

---

4. sayfa

(b) Çok Çekişmeli Ortamlar. Algorand, yapabilecek çok güçlü bir Düşmana karşı
(1) istediği herhangi bir kullanıcıyı istediği zaman anında bozmak şartıyla,
izinsiz ortamda, sistemdeki paranın 2 / 3'ü dürüst kullanıcıya aittir. (İçinde
izin verilen ortam, para ne olursa olsun, kullanıcıların 2 / 3'ünün dürüst olması yeterlidir.)
(2) tüm bozuk kullanıcıları tamamen kontrol eder ve mükemmel bir şekilde koordine eder; ve
(3) dürüst bir kullanıcı tarafından gönderilen her mesajın m olması koşuluyla, tüm mesajların teslimini planlayın
Dürüst kullanıcıların% 95'ine yalnızca m boyutuna bağlı olan λ m bir süre içinde ulaşır .
Ana Özellikler Güçlü düşmanımızın Algorand'daki varlığına rağmen
• Gerekli hesaplama miktarı minimumdur. Esasen, kaç kullanıcı olursa olsun
sistemde mevcutsa, bin beş yüz kullanıcının her biri, en fazla birkaç saniye
hesaplama.
• 10 dakikadan daha kısa sürede Yeni Bir Blok Oluşturulur ve fiilen blok zincirinden asla çıkmaz.
Örneğin, beklenti olarak, ilk uygulamada bir blok oluşturma süresi daha azdır.
Λ + 12.4λ'dan daha fazla, burada Λ, eşler arası dedikodularda bir bloğu yaymak için gerekli zamandır
moda, hangi blok boyutu seçilirse seçilsin ve λ, 1.500 200B-
uzun mesajlar. (Gerçekten merkezi olmayan bir sistemde, Λ esasen içsel bir gecikmedir.
Algorand ve blok oluşturmadaki sınırlayıcı faktör ağ hızıdır.) İkinci düzenleme şu özelliklere sahiptir:
aslında deneysel olarak (?) test edilmiştir, bu da bir bloğun 40'tan daha az bir sürede oluşturulduğunu gösterir.
saniye.
Ek olarak, Algorand'ın blok zinciri yalnızca ihmal edilebilir bir olasılıkla (yani birden az
bir trilyonda) ve böylece kullanıcılar yeni bir blokta yer alan ödemeleri,
blok belirir.
• Tüm güç kullanıcıların kendilerindedir. Algorand, gerçekten dağıtılmış bir sistemdir. Özellikle,
Hangi işlemleri kontrol edebilecek dış varlık (Bitcoin'deki "madenciler" olarak) yoktur.
tanınmış.
Algorand'ın Teknikleri.
1. Yeni ve Hızlı Bir Bizans Anlaşması Protokolü. Algorand, üzerinden yeni bir blok oluşturur
yeni bir kriptografik, mesaj ileten, ikili Bizans anlaşması (BA) protokolü, BA ⋆ . Protokol
BA ⋆ sadece bazı ek özellikleri (yakında tartışacağımız) karşılamakla kalmaz, aynı zamanda çok hızlıdır.
Kabaca söylersek, ikili giriş versiyonu 3 adımlı bir döngüden oluşur ve burada bir oyuncunun tek bir
diğer tüm oyunculara mesaj m i . Eksiksiz ve senkronize bir ağda yürütülür, daha fazlası ile
oyuncuların 2 / 3'ünden daha fazla dürüst, 1/3 olasılıkla, her döngüden sonra protokol sona eriyor
anlaşma. (BA ⋆ protokolünün Bizans anlaşmasının orijinal tanımını karşıladığını vurguluyoruz.
Pease, Shostak ve Lamport [31] , herhangi bir zayıflama olmadan.)
Algorand, farklı iletişimimizde bir anlaşmaya varmak için bu ikili BA protokolünü kullanıyor
model, her yeni blokta. Üzerinde anlaşmaya varılan blok, daha sonra, önceden belirlenen sayıda
uygun doğrulayıcıların dijital imzası ve ağ üzerinden yayılır.
2. Kriptografik Sıralama. Çok hızlı olmasına rağmen, BA ⋆ protokolü daha fazla
milyonlarca kullanıcı tarafından oynandığında hız. Buna göre, Algorand BA oyuncuları seçer ⋆ olmak

---

tüm kullanıcılar kümesinin çok daha küçük bir alt kümesi. Farklı bir güç yoğunlaşmasından kaçınmak için
Sorun, her yeni bir blok B r, BA, yeni bir çalıştırma ile inşa ve üzerinde anlaşılan olacak ⋆ ,
ayrı bir seçilmiş doğrulayıcı grubu tarafından, SV r . Prensip olarak, böyle bir set seçmek kadar zor olabilir
doğrudan B r'yi seçerek . Bu potansiyel sorunu kucaklayan, adını verdiğimiz bir yaklaşımla inceliyoruz.
Maurice Herlihy'nin anlayışlı önerisi, kriptografik tasnif. Sıralama pratiğidir
Yetkililerin geniş bir uygun bireyler kümesinden rastgele seçilmesi [6]. (Sıralama uygulandı

yüzyıllar boyunca: örneğin, Atina, Floransa ve Venedik cumhuriyetleri tarafından. Modern yargıda sistemler, rastgele seçim genellikle jüri seçiminde kullanılır. Rastgele örnekleme de son zamanlarda David Chaum [9] tarafından seçimleri savundu .) Merkezi olmayan bir sistemde, tabii ki,
Her doğrulayıcı seti SV r'nin üyelerini rastgele seçmek için gerekli olan rastgele paralar sorunludur. Bu nedenle, tüm kullanıcıların popülasyonundan her bir doğrulayıcı kümesini seçmek için kriptografiye başvuruyoruz.
otomatik (yani mesaj alışverişi gerektirmeyen) ve rasgele olması garanti edilen bir şekilde.
Temelde, önceki bloktan otomatik olarak belirlemek için bir kriptografik işlev kullanıyoruz.
B r-1 , bir kullanıcı, lider, yeni blok B r'yi önermekten sorumlu ve doğrulayıcı , SV r'yi
lider tarafından önerilen blok üzerinde anlaşmaya varmak için ücret alın. Kötü niyetli kullanıcılar etkileyebileceğinden
B r-1'in bileşimi (örneğin, bazı ödemelerini seçerek), özel olarak oluşturuyor ve kullanıyoruz
rth bloğunun liderinin ve doğrulayıcı kümesi SV r'nin gerçekten olduğunu kanıtlamak için ek girdiler rastgele seçilmiş.
3. Miktar (Tohum) Q r . Blok zincirindeki son B r-1 bloğunu kullanarak
otomatik olarak bir sonraki doğrulayıcı kümesini ve yeni bloğu inşa etmekten sorumlu lideri belirler B r . Bu yaklaşımla ilgili zorluk, sadece biraz farklı bir ödeme seçerek
önceki turda, güçlü Adversary'imiz bir sonraki lider üzerinde muazzam bir kontrol elde ediyor. O olsa bile
sistemdeki oyuncuların / paranın yalnızca 1 / 1000'ini kontrol etti, tüm liderlerin
kötü niyetli. (Önsezi Bölüm 4.1'e bakın .) Bu zorluk, tüm risk kanıtı yaklaşımlarının merkezinde yer alır,
ve bildiğimiz kadarıyla, şimdiye kadar tatmin edici bir şekilde çözülmedi.
Bu zorluğun üstesinden gelmek için, kasıtlı olarak ayrı ve dikkatli bir şekilde inşa ediyor ve sürekli güncelliyoruz.
bizim tarafımızdan kanıtlanabilir bir şekilde, sadece tahmin edilemez değil, aynı zamanda etkilenemez de olan Q r ,
güçlü Düşman. Olan Q ile ilgili olabilir , r o Q, olduğu gibi, r'inci tohum olarak r Algorand seçtiği, gizli kriptografik sıralama yoluyla, oluşturulmasında özel bir rol oynayacak tüm kullanıcılar
rth bloğu.
4. Gizli Kriptografik Sıralama ve Gizli Kimlik Bilgileri. Rastgele ve net
Doğrulayıcı setini ve sorumlu lideri seçmek için geçerli son blok B r-1'i kullanmak
yeni bloğu inşa etmek için B r yeterli değildir. B r-1'in B r'yi oluşturmadan önce bilinmesi gerektiğinden ,
B r-1'in içerdiği son etkili olmayan Q r-1 miktarı da bilinmelidir. Buna göre
doğrulayıcılar ve B r bloğunu hesaplamaktan sorumlu liderdir . Böylece güçlü düşmanımız
Bunlardan belki hemen yozlaşmış bütün bunlar B hakkında herhangi bir tartışma meşgul
önce r böylece almak için,
onayladıkları blok üzerinde tam kontrol.
Bu sorunu önlemek için liderler (ve aslında doğrulayıcılar da) rollerini gizlice öğrenirler, ancak gerçekten bu role sahip olan herkese kanıtlayabilecek uygun bir kimlik bilgisi hesaplayın. Ne zaman bir kullanıcı bir sonraki bloğun lideri olduğunu özel olarak anlar, önce gizlice kendi
önerilen yeni bloğa sahip olur ve ardından bunu (onaylanabilmesi için) kendi bloğuyla birlikte dağıtır.
Kimlik. Bu şekilde, Düşman hemen bir sonrakinin liderinin kim olduğunu anlayacaktır.
engelleniyor ve onu hemen yozlaştırabilmesine rağmen, Rakip için çok geç olacaktır.
yeni bir blok seçimini etkiler. Nitekim, liderin mesajını artık "geri arayamaz".
5

Sayfa 6

güçlü bir hükümetin, WikiLeaks tarafından viral olarak yayılan bir mesajı şişeye geri koyabileceğinden daha fazla.
Göreceğimiz gibi, liderin benzersizliğini garanti edemeyiz veya herkesin liderin kim olduğundan emin olamayacağından emin olamayız.
liderin kendisi dahil! Ancak, Algorand'da kesin ilerleme garanti edilecektir.
5. Oyuncu Değiştirilebilirliği. Yeni bir blok önerdikten sonra, lider de "ölebilir" (veya

Düşman tarafından bozulmuş), çünkü işi bitti. Ama, SV içinde verifiers için r , işler daha az olan basit. Nitekim, yeni B r bloğunun yeterince çok sayıda imzayla onaylanmasından sorumlu olarak , önce liderin önerdiği blok üzerinde Bizans anlaşmasını yürütmeleri gerekir. Problem şu, ne kadar verimli olursa olsun, BA ⋆ birden fazla adım ve oyuncularının 2 / 3'ünün dürüstlüğünü gerektirir.
Bu bir sorundur, çünkü verimlilik nedenlerinden ötürü, BA ⋆ oyuncu seti küçük SV r kümesinden oluşur.
tüm kullanıcılar arasından rastgele seçilir. Böylece, güçlü Düşmanımız yapamasa da tüm kullanıcıların 1 / 3'ünü bozarlar, kesinlikle SV r ! ' nin tüm üyelerini bozabilir .
Neyse ki , BA ⋆ protokolünün , mesajları bir uçtan uca yayarak yürütüldüğünü kanıtlayacağız. akran modası, oyuncu tarafından değiştirilebilir. Bu yeni gereksinim, protokolün doğru ve Her adımı tamamen yeni ve rastgele bir şekilde gerçekleştirilse bile verimli bir şekilde fikir birliğine varır.
ve bağımsız olarak seçilen oyuncular seti. Böylece, milyonlarca kullanıcıyla, her küçük oyuncu grubu BA adımıyla ilişkili ⋆ büyük olasılıkla sonraki küme ile boş kesişme noktasına sahiptir.
Ek olarak, BA ⋆'ın farklı adımlarındaki oyuncu setleri muhtemelen tamamen farklı olacaktır. kardinaliteler. Dahası, her setin üyeleri bir sonraki oyuncu setinin kim olacağını bilmiyor. olmak ve herhangi bir iç durumu gizlice geçmeyin.
Değiştirilebilir oyuncu özelliği, dinamik ve çok güçlü olanı yenmek için gerçekten çok önemlidir. Öngördüğümüz düşman. Değiştirilebilir oyuncu protokollerinin birçok alanda çok önemli olacağına inanıyoruz.
bağlamlar ve uygulamalar. Özellikle, güvenli bir şekilde küçük alt protokolleri yürütmek için çok önemli olacaklar
Daha büyük bir oyuncu evrenine gömülü, dinamik bir düşmana sahip, hatta onları bile yozlaştırabilen toplam oyuncuların küçük bir kısmı, küçük oyunculardaki tüm oyuncuları bozmakta zorluk çekmez. alt protokol.
Ek Bir Özellik / Teknik: Tembel Dürüstlük Dürüst bir kullanıcı, talimatına uyar çevrimiçi olmayı ve protokolü çalıştırmayı içeren talimatlar. Algorand sadece mütevazı hesaplama ve iletişim gereksinimi, çevrimiçi olma ve protokolü çalıştırma arka plan ”büyük bir fedakarlık değildir. Tabii ki, dürüst oyuncular arasında birkaç "eksiklik" ani bağlantı kaybı veya yeniden başlatma ihtiyacı nedeniyle, otomatik olarak tolere edilir (çünkü her zaman bu kadar az oyuncunun geçici olarak kötü niyetli olduğunu düşünebiliriz). Bununla birlikte, şunu belirtelim:
Algorand, dürüst kullanıcıların olduğu yeni bir modelde çalışacak şekilde basitçe uyarlanabilir. çoğu zaman çevrimdışı. Yeni modelimiz gayri resmi olarak aşağıdaki şekilde tanıtılabilir.
Tembel Dürüstlük. Kabaca konuşursak, bir kullanıcı i tembel ama dürüsttür, eğer (1) tüm reçetesini uygularsa
protokole katılması istendiğinde talimatlar ve (2) katılması istendiğinde
protokole nadiren ve uygun bir önceden haber vererek.
Böylesine rahat bir dürüstlük anlayışıyla, dürüst insanların olacağından daha da emin olabiliriz. ihtiyaç duyduğumuzda elinizin altındadır ve Algorand, bu durumda
Sistem, belirli bir zamanda bile güvenli bir şekilde çalışır.
katılan oyuncuların çoğu kötü niyetli.

---

7. Sayfa

1.3 Yakından İlgili Çalışma
İş kanıtı yaklaşımları (alıntı [29] ve [4] gibi) bizimkine oldukça ortogonaldir. Öyleyse Mesaj ileten Bizans anlaşmasına veya pratik Bizans hata toleransına dayanan yaklaşımlar (alıntı [8] gibi). Aslında, bu protokoller tüm kullanıcılar arasında çalıştırılamaz ve modelimizde, uygun şekilde küçük bir kullanıcı grubuyla sınırlandırılmalıdır. Aslında, güçlü düşmanımız benim
Bir BA protokolünü gerçekten çalıştırmakla yükümlü küçük bir sete dahil olan tüm kullanıcıları derhal bozar.

Yaklaşımımız , kullanıcıların "gücü" anlamında, hissenin ispatı [2] ile ilgili olarak düşünülebilir.
blok yapımında, sistemde sahip oldukları parayla orantılıdır (—söyleyin— aksine
“emanet” e koydukları para).
Bizimkine en yakın makale Pass ve Shi'nin Sleepy Consensus Modelidir [30]. Önlemek için
iş kanıtı yaklaşımında gerekli olan ağır hesaplama, belgeleri dayanır (ve nazikçe
krediler) Algorand'ın gizli kriptografik sıralaması. Ortak olan bu çok önemli yönle,
Kağıtlarımız arasında önemli farklılıklar var. Özellikle,
(1) Ayarlarına yalnızca izin verilir. Aksine, Algorand aynı zamanda izinsiz bir sistemdir.
(2) Nakamoto tarzı bir protokol kullanıyorlar ve bu nedenle blok zincirleri sık sık
çatallanıyor. olmasına rağmen
iş kanıtından vazgeçerek, protokollerinde gizlice seçilmiş bir liderden,
en uzun geçerli (daha zengin anlamda) blockchain. Bu nedenle çatallar kaçınılmazdır ve bunu beklemek gerekir
blok, zincirde yeterince "derin" dir. Nitekim bir düşmanla hedeflerine ulaşmak için
uyarlanabilir bozulmalar yapabilen, bir bloğun poli (N) derinliğinde olmasını gerektirirler; burada N,
sistemdeki toplam kullanıcı sayısı. Dikkat edin, bir bloğun üretilebileceğini varsayarsak bile
Bir dakika içinde, N = 1 milyon kullanıcı varsa, birinin yaklaşık 2 milyon yıl beklemesi gerekirdi.
bir bloğun N 2 -derin olması için ve yaklaşık 2 yıl boyunca bir bloğun N-derin olması için. Aksine,
Algorand'ın blok zinciri, Rüşvetin yolsuzluğuna rağmen yalnızca ihmal edilebilir bir olasılıkla çatallanıyor
anında ve uyarlamalı olarak kullanıcılar ve yeni bloklarına hemen güvenilebilir.
(3) Bireysel Bizans anlaşmalarını işlemezler. Bir anlamda, sadece garanti ediyorlar
"Artan değerler dizisi üzerinde nihai fikir birliği". Onlarınki bir durum çoğaltma protokolüdür.
BA'dan daha fazla ve bireysel bir faiz değeri üzerinde Bizans anlaşmasına varmak için kullanılamaz.
Aksine, Algorand milyonlarca kullanıcının hızlı bir şekilde
belirli bir ilgi değeri üzerinde Bizans anlaşmasına varmak.
(4) Zayıf senkronize saatler gerektirirler. Yani, tüm kullanıcıların saatleri küçük bir zamanla dengelenir
δ. Aksine, Algorand'da saatlerin yalnızca (esasen) aynı "hıza" sahip olması gerekir.
(5) Protokolleri tembel ama dürüst kullanıcılarla veya çevrimiçi kullanıcıların dürüst çoğunluğuyla çalışır.
Dürüst kullanıcıların toplu halde çevrimdışı olma sorununu gündeme getirdiği için Algorand'a nazikçe teşekkür ediyorlar.
cevap olarak tembel dürüstlük modelini ortaya koymak. Protokolleri sadece tembelde çalışmıyor
dürüstlük modeli, aynı zamanda düşman uykulu modelinde, bir düşmanın hangi kullanıcıları seçtiği
her zaman çevrimiçi kullanıcıların çoğunluğunun dürüst olması koşuluyla çevrimiçi ve çevrimdışıdır.2
2 Makalelerinin orijinal versiyonu, düşman uykulu modelinde aslında yalnızca güvenlik olarak değerlendiriliyordu. The
Algorand'ın kendilerininkinden önce gelen orijinal versiyonu da açıkça
çevrimiçi oyuncular her zaman dürüsttür, ancak tembel dürüstlük modeli lehine açıkça dikkate alınmaz.
(Örneğin, dürüst kullanıcıların yarısı herhangi bir noktada çevrimdışı olmayı seçerse, kullanıcıların çoğu
çevrimiçi çok kötü niyetli olabilir. Bu nedenle, bunun olmasını önlemek için, Düşman kendi
Bozuk oyuncuların da çevrimdışına geçmesi, ki bu açıkça kendi çıkarına aykırıdır.) Çoğunluğa sahip
bir protokol olduğuna dikkat edin.
tembel ama dürüst oyuncuların çoğu, çevrimiçi kullanıcıların çoğu her zaman kötü niyetli ise gayet iyi çalışıyor. Bu böyledir çünkü
Yeterli sayıda dürüst oyuncu, zamanın nadir bir noktasında çok önemli olacaklarını bilerek,
bu anlarda çevrimdışına çıkmamaya ya da Düşman tarafından çevrimdışına alınmaya zorlanamaz,
çünkü kim olduğunu bilmiyor.
çok önemli dürüst oyuncular olabilir.

---

8. Sayfa

(6) Basit ve dürüst bir çoğunluk gerektirirler. Aksine, Algorand'ın mevcut sürümü şunu gerektirir: 2/3 dürüst çoğunluk.
Yakınımızdaki bir diğer kağıt ise Ouroboros: Provably Secure Proof-of-Stake Blockchain Protokolü, Kiayias, Russell, David ve Oliynykov [20] tarafından . Onların sistemi de bizden sonra ortaya çıktı. Aynı zamanda
kanıtlanabilir bir şekilde iş kanıtından vazgeçmek için kriptografik sıralama kullanır. Ancak, onların sistem yine, çatalların hem kaçınılmaz hem de sık olduğu Nakamoto tarzı bir protokoldür.
(Bununla birlikte, modellerinde, blokların uykulu fikir birliği modeli kadar derin olması gerekmez.) Ayrıca,
sistemleri aşağıdaki varsayımlara dayanmaktadır: yazarların kendi sözleriyle, "(1)
ağ son derece eşzamanlıdır, (2) seçilen paydaşların çoğunluğu ihtiyaç duyulduğunda kullanılabilir her bir döneme katılmak için, (3) paydaşların uzun süre çevrimdışı kalmaması,
(4) yolsuzlukların uyarlanabilirliği, küçük bir gecikmeye tabidir.
güvenlik parametresi. " Aksine, Algorand ezici bir olasılıkla çatalsız ve
bu 4 varsayımdan hiçbirine dayanmamaktadır. Özellikle Algorand'da Düşman,
kontrol etmek istediği kullanıcıları anında bozuyor.

2 Hazırlıklar

2.1 Şifreleme İlkelleri

İdeal Hashing. Verimli bir şekilde hesaplanabilen bir kriptografik hash fonksiyonuna güveneceğiz, H, keyfi olarak uzun dizeleri sabit uzunluktaki ikili dizelerle eşler. Uzun bir geleneğin ardından modelliyoruz
H rastgele bir oracle olarak, esasen olası her dizge s'yi rastgele ve
bağımsız olarak seçilen (ve sonra sabitlenen) ikili dizi, seçilen uzunluktaki H (s).
Bu yazıda, H'nin 256-bit uzun çıktıları vardır. Gerçekten de, bu uzunluk yeterince kısadır.
sistem verimli ve sistemi güvenli hale getirmek için yeterince uzun. Örneğin, H'nin çarpışma olmasını istiyoruz.
esnek. Yani, H (x) = H (y) olacak şekilde iki farklı x ve y dizisi bulmak zor olmalı.
H, 256 bit uzun çıktılara sahip rastgele bir oracle olduğunda, böyle bir dizi çiftini bulmak gerçekten de zor. (Rastgele denemek ve doğum günü paradoksuna güvenmek, 2 256/2 = 2 128
denemeler.)
Dijital İmzalama. Dijital imzalar, kullanıcıların bilgileri birbirleriyle doğrulamasına olanak tanır herhangi bir gizli anahtarı paylaşmadan. Bir dijital imza şeması, üç hızlı
algoritmalar: bir olasılıksal anahtar oluşturucu G, bir imzalama algoritması S ve bir doğrulama algoritması V.
Yeterince yüksek bir tam sayı olan bir güvenlik parametresi k verildiğinde, bir kullanıcı i bir çift üretmek için G'yi kullanır.
k-bit anahtarları (yani dizeler): bir "genel" anahtar pk i ve eşleşen bir "gizli" imzalama anahtarı sk i . En önemlisi, bir
açık anahtar, karşılık gelen gizli anahtara "ihanet etmez". Yani, pk i bilgisi verildiğinde bile , hayır i'den başka biri sk i'yi astronomik zamandan daha kısa sürede hesaplayabilir .
Kullanıcı i kullanır sk i mesajları dijital olarak imzalamak için. Olası her mesaj için (ikili dizi) m, önce i
karma m ve ardından k-bit dizesini üretmek için H (m) ve sk i girişleri üzerinde S algoritmasını çalıştırır.
sig pk ben (m)
S (H (m), sk ben).3
3 H çarpışmaya dirençli olduğu için, m'yi imzalayarak birinin yanlışlıkla başka bir
mesaj m ′ .

Sayfa 9

İkili dizge sig pk i (m), i'nin m dijital imzası olarak adlandırılır (pk i'ye göre) ve olabilir
genel anahtar pk i bağlamdan açık olduğunda daha basit bir şekilde sig i (m) ile gösterilir .
Herkes pk bilerek ı i tarafından üretilen dijital imzaları doğrulamak için kullanabilirsiniz. Özellikle,

(a) bir i oyuncusunun ortak anahtarı pk i , (b) bir mesaj m ve (c) bir s dizesi, yani i'nin iddia edildiği gibi
m mesajının dijital imzası, doğrulama algoritması V, ya EVET ya da HAYIR çıktılar verir.
Dijital imza şemasından istediğimiz özellikler şunlardır:
1. Meşru imzalar her zaman doğrulanır: s = sig i (m) ise, V (pk i , m, s) = Y ES; ve
2. Dijital imzaların taklit edilmesi zordur: Sk i bilgisi olmadan bir dizi bulma zamanı
bu V (pk i , m, s) = Y ES, i ile asla imzalanmayan bir mesaj için, astronomik olarak uzun.
(Goldwasser, Micali ve Rivest'in [17] güçlü güvenlik gereksiniminin ardından , bu doğrudur
Herhangi bir başka mesajın imzası alınsa bile.)
Buna göre, bir başkasının onun adına mesaj imzalamasını önlemek için, bir oyuncunun
Anahtar sk imzalama i sırrı (dolayısıyla dönem "gizli anahtar"), ve mesajları doğrulamak için kimseyi etkinleştirmek için
o imzalıyor, pk i anahtarını duyurmakla ilgileniyorum (dolayısıyla "genel anahtar" terimi).
Genel olarak, bir mesaj m, imza sig den alınabilir değil ı (m). Neredeyse başa çıkmak için
kavramsal olarak uygun "geri alınabilir" özelliğini karşılayan dijital imzalarla (yani,
İmzalayanın ve mesajın bir imzadan kolayca hesaplanabileceğini garanti eder, biz tanımlarız
SIG pk ben (m) = (i, m, sig pk i (m)) ve
SIG i (m) = (i, m, sig i (m)), eğer pk i açıksa .
Benzersiz Dijital İmza. Ayrıca, dijital imza şemalarını (G, S, V) de dikkate alıyoruz.
ek mülkün ardından.
3. Benzersizlik. Pk / , m, s ve s / dizelerini bulmak zordur , öyle ki
s = s /
ve
V (pk / , m, s) = V (pk / , m, s /) = 1.
(Benzersizlik özelliğinin, yasal olarak oluşturulmamış pk / dizeleri için de geçerli olduğunu unutmayın.
genel anahtarlar. Bununla birlikte, özellikle, benzersizlik özelliği, birinin kullanılması durumunda eşleşen bir gizli anahtar sk ile birlikte bir genel anahtar pk'yi hesaplamak için belirtilen anahtar oluşturucu G,
ve böylece sk'yi biliyordu, onun için de iki farklı dijital reklam bulması esasen imkansız olurdu. pk'ye göre aynı mesajın imzaları.)
Uyarılar
• Benzersiz imzalardan doğrulanabilir rastgele işlevlere. Bir dijitale göre
benzersizlik özelliğine sahip imza şeması, eşleme m → H (sig i (m)) ile ilişkilendirilir
her olası dize m, benzersiz, rastgele seçilmiş, 256 bitlik bir dize ve bunun doğruluğu
eşleme sig imza verilir kanıtlanabilir i (m).
Yani, benzersizlik özelliğini esasen karşılayan ideal karma ve dijital imza şeması
ortaya konduğu şekliyle doğrulanabilir bir rastgele fonksiyonun temel bir uygulamasını sağlar ve Micali, Rabin ve Vadhan [27] . (Orijinal uygulamaları zorunlu olarak daha karmaşıktı, çünkü ideal hashlemeye güvenmediler.)

Sayfa 10

• Dijital imzalar için üç farklı ihtiyaç. Algorand'da, dijitale güvendiğim bir kullanıcı
için imzalar
(1) i'nin kendi ödemelerinin doğrulanması. Bu uygulamada, anahtarlar "uzun vadeli" olabilir (yani, uzun bir süre boyunca birçok mesajı imzalayın) ve sıradan bir imza şemasından gelir.
(2) Bir r turunun bazı adımlarında hareket etme hakkına sahip olduğumu kanıtlayan kimlik bilgilerinin oluşturulması. Buraya,
anahtarlar uzun vadeli olabilir, ancak benzersizlik özelliğini karşılayan bir şemadan gelmelidir.
(3) Harekete geçtiği her adımda gönderdiğim mesajı doğrulamak. Burada anahtarlar olmalı
geçici (yani, ilk kullanımlarından sonra yok edilir), ancak sıradan bir imza şemasından gelebilir.
• Küçük maliyetli bir basitleştirme. Basit olması açısından, her bir i kullanıcının tek bir long-dönem anahtarı. Buna göre, böyle bir anahtar, benzersizliği olan bir imza şemasından gelmelidir.
Emlak. Böyle bir basitliğin küçük bir hesaplama maliyeti vardır. Tipik olarak, aslında, benzersiz dijital

imzaların üretilmesi ve doğrulanması sıradan imzalardan biraz daha pahalıdır.
2.2 İdealleştirilmiş Halka Açık Defter
Algorand, idealleştirilmiş bir genel muhasebeye dayalı olarak aşağıdaki ödeme sistemini taklit etmeye çalışıyor.
1. Başlangıç Durumu. Para, bireysel genel anahtarlarla ilişkilendirilir (özel olarak oluşturulmuş ve Kullanıcılara aittir). Pk 1 , ..., pk j ilk genel anahtarlar ve a 1 , ..., a j ise kendi
ilk para birimi tutarları, ardından başlangıç durumu
S 0 = (pk 1 , bir 1), ..., (pk j , bir j),
sistemde ortak bilgi olduğu varsayılmaktadır.
2. Ödemeler. Pk şu anda ≥ 0 para birimine sahip bir açık anahtar olsun, pk / başka bir genel anahtar ve a / a'dan büyük olmayan negatif olmayan bir sayı. O halde, (geçerli) bir ödeme ℘ dijital bir Bir transfer belirten pk göre imza, / pk pk para birimleri / birlikte
bazı ek bilgilerle. Sembollerde,
℘ = SIG pk (pk, pk / , a / , I, H (I)),
yararlı olduğu düşünülen ancak hassas olmayan ek bilgileri temsil ettiğim durumlarda (ör. zaman bilgi ve ödeme tanımlayıcı) ve hassas kabul edilen ek bilgiler (ör.
ödeme nedeni, muhtemelen pk ve pk / sahiplerinin kimlikleri vb.).
Pk'ye (veya sahibine) ödeyen, her pk / (veya sahibine) alacaklı olarak ve a / as
ödeme miktarı ℘.
Ödeme Yoluyla Ücretsiz Katılım. Kullanıcıların sisteme istedikleri zaman katılabileceğini unutmayın. kendi açık / gizli anahtar çiftlerini oluşturmak. Buna göre, açık anahtar pk / içinde görünen Yukarıdaki ödeme ℘, hiçbir zaman paraya "sahip olmayan" yeni oluşturulmuş bir genel anahtar olabilir
önce.
3. Sihirli Defter. İdealleştirilmiş Sistemde, tüm ödemeler geçerlidir ve kurcalamaya karşı korumalı olarak görünür.
Herkesin görmesi için "gökyüzünde yayınlanan" ödeme setlerinin L listesi:
L = P AY 1 , P AY 2 , ...,

Her blok P AY r + 1 , bloğun ortaya çıkmasından bu yana yapılan tüm ödemelerin kümesinden oluşur. P AY r . İdeal sistemde, sabit (veya sınırlı) bir süre sonra yeni bir blok belirir.
Tartışma.
• Daha Genel Ödemeler ve Harcanmamış İşlem Çıktısı. Daha genel olarak, eğer bir açık anahtar pk, a miktarına sahipse, o zaman geçerli bir pk ödemesi, tutarları transfer edebilir a /
1 , a /
2 , ...,
sırasıyla pk / anahtarlarına
1 , pk /
2 , ..., j a / olduğu sürece
j ≤ a.
Bitcoin ve benzeri sistemlerde, bir açık anahtar pk'nin sahip olduğu para, ayrı olarak ayrılmıştır. miktarlar ve pk tarafından yapılan bir ödeme ℘ böyle ayrılmış bir miktarı a'nın tamamını transfer etmelidir.
Pk , başka bir anahtara a / < a'nın sadece bir kısmını transfer etmek istiyorsa , o zaman aynı zamanda bakiye, harcanmamış işlem çıktısı, başka bir anahtara, muhtemelen kendisini pk.
Algorand, ayrılmış miktarlara sahip anahtarlarla da çalışır. Ancak, odaklanmak için
Algorand'ın yeni yönleri, daha basit ödeme biçimlerimize bağlı kalmak kavramsal olarak daha kolaydır
ve kendileriyle ilişkilendirilmiş tek bir miktara sahip anahtarlar.
• Şu anki durum. İdealleştirilmiş Şema, doğrudan mevcut durum hakkında bilgi sağlamaz. sistemin durumu (yani, her bir açık anahtarın kaç para birimine sahip olduğu hakkında). Bu bilgi Magic Ledger'dan çıkarılabilir.
İdeal sistemde, aktif bir kullanıcı en son durum bilgilerini sürekli olarak depolar ve günceller,

ya da aksi takdirde sıfırdan ya da son kez yeniden inşa etmesi gerekecekti.
hesapladı. (Bu makalenin bir sonraki sürümünde, Algorand'ı,
kullanıcıların mevcut durumu verimli bir şekilde yeniden yapılandırması.)
• Güvenlik ve "Gizlilik". Dijital imzalar, hiç kimsenin sahte ödeme yapamayacağını garanti eder.
Başka kullanıcı. Bir ödemede $\wp$, genel anahtarlar ve tutar gizli değil, hassas
bilgi ben. Aslında, $\wp$'da yalnızca H (I) görünür ve H ideal bir hash işlevi olduğundan, H (I)
rastgele 256 bitlik bir değerdir ve bu nedenle neyin daha iyi olduğunu anlamanın bir yolu yoktur.
sadece tahmin etmek. Yine de, ne olduğumu kanıtlamak için (örneğin, ödemenin nedenini kanıtlamak için)
mükellef sadece I. Açıklanan I'in doğruluğu H (I) hesaplanarak doğrulanabilir.
ve elde edilen değeri $\wp$'nın son maddesiyle karşılaştırmak. Aslında, H çarpışmaya dirençli olduğu için,
H (I) = H (I /) olacak şekilde ikinci bir I / değeri bulmak zordur .
2.3 Temel Kavramlar ve Gösterimler
Anahtarlar, Kullanıcılar ve Sahipler Aksi belirtilmedikçe, her bir genel anahtar (kısaca "anahtar") uzundur.
benzersizlik özelliğine sahip bir dijital imza şemasına göre terim ve göreceli. Katıldığım bir genel anahtar
zaten sistemde bulunan başka bir genel anahtar j i'ye ödeme yaptığında sistem.
Renk için anahtarları kişiselleştiriyoruz. Bir anahtara "o" diyoruz, dürüst olduğumu söylüyorum, gönderdiğim
ve mesajları alır, vb. Kullanıcı, anahtarın eşanlamlısıdır. Bir anahtarı ayırt etmek istediğimizde
ait olduğu kişi, sırasıyla “dijital anahtar” ve “sahip” terimlerini kullanıyoruz.
İzinsiz ve İzinli Sistemler. Dijital anahtar ücretsiz ise sistem izinsizdir
herhangi bir zamanda katılmak ve bir sahip birden fazla dijital anahtara sahip olabilir; aksi takdirde izin verilir.

Sayfa 12

Benzersiz Gösterim Algorand'daki her nesnenin benzersiz bir temsili vardır. Özellikle,
her küme {(x, y, z,...): x ∈ X, y ∈ Y, z ∈ Z, ...} önceden belirlenmiş bir şekilde sıralanmıştır: ör., ilk sözlüksel olarak x, sonra y, vb.
Aynı Hızlı Saatler Küresel bir saat yoktur: daha ziyade, her kullanıcının kendi saati vardır. Kullanıcı saatleri
herhangi bir şekilde senkronize edilmesine gerek yoktur. Bununla birlikte, hepsinin aynı hıza sahip olduğunu varsayıyoruz.
Örneğin, i kullanıcısının saatine göre saat 12.00 olduğunda, saat 2:30 olabilir.
başka bir j kullanıcısının saati, ancak i'nin saatine göre 12:01 olduğunda, 2:31 olacaktır.
j'nin saatine. Yani, "her kullanıcı için bir dakika aynıdır (yeterince, esasen aynı)".
Yuvarlar Algorand, mermi adı verilen r = 0,1, ... mantıksal birimler halinde düzenlenir.
Raundları belirtmek için sürekli olarak üst simgeler kullanırız. Sayısal olmayan bir miktar Q olduğunu belirtmek için
(örneğin, bir dizge, bir genel anahtar, bir küme, bir dijital imza, vb.) bir r turunu ifade eder, biz sadece Q r yazarız .
Sadece Q gerçek bir sayı olduğunda (sayı olarak yorumlanabilen bir ikili dizgenin aksine),
Q (r) yazarız , böylece r sembolü Q'nun üssü olarak yorumlanamaz.
(A) tur r> 0'da, tüm genel anahtarların kümesi PK r'dir ve sistem durumu
S r = {(i, bir
(r)
i , ...): i ∈ PK r },
burada bir
(r)
ben
açık anahtarın kullanabileceği para miktarıdır i. PK r'nin ,
S r ve bu SR ayrıca her bir genel anahtar i için diğer bileşenleri de belirleyebilir.

Tur 0 için, PK 0 ilk genel anahtarlar kümesidir ve S 0 başlangıç durumudur. Hem PK 0 hem de S 0'ın sistemde ortak bilgi olduğu varsayılır. Basit olması için, r turunun başında, yani PK 1 , ..., PK r ve S 1 , ..., S r'dir .
Bir r turunda, sistem durumu S r'den S r + 1'e geçiş yapar : sembolik olarak,
Yuvarlak r: S r - → S r + 1 .
Ödemeler Algorand'da, kullanıcılar sürekli olarak ödeme yaparlar (ve bunları bir şekilde yayarlar) 2. alt bölümde açıklanmıştır . 7). Bir i ∈ PK r kullanıcısının ödemesi ℘ aynı biçime ve anlam bilgisine sahiptir
İdeal Sistemdeki gibi. Yani,
℘ = SIG i (i, i / , a, I, H (I)).
Ödeme ℘ münferit olarak r turunda geçerlidir (kısaca r tur ödemesidir) eğer (1) miktarı a, a'dan küçük veya eşittir
(r)
i ve (2) herhangi bir resmi payset P AY görünmez r ' r için / <r.
(Aşağıda açıklandığı gibi, ikinci koşul, ℘'nin henüz etkili olmadığı anlamına gelir.
Bir dizi r-r ödemesi, tutarlarının toplamı en fazla a
(r)
i .
Ödeme Kümeleri Bir rota-r ödeme seti P, her bir i kullanıcısı için ödemelerin P'deki i'nin (muhtemelen hiçbiri) toplu olarak geçerlidir. Tüm r-r maaş kümeleri ÖDEME (r) şeklindedir. Bir yuvarlak r
ödeme kümesi P, P'nin hiçbir üst kümesi r'ye eşit bir ödeme kümesi değilse maksimaldir.
Aslında bir ödemesinin aynı zamanda bir yuvarlak ρ, ℘ = SIG i (ρ, i, i / , a, I, H (I)) belirtmesini de öneriyoruz ,
ve bazı sabit negatif olmayan tamsayı k için [ρ, ρ + k] dışındaki herhangi bir turda geçerli olamaz.4
4 Bu, ℘'nin "etkili" olup olmadığını kontrol etmeyi kolaylaştırır (yani, bazı ödeme setlerinin P AY r , ℘ içerir. K = 0 olduğunda, ℘ = SIG i (r, i, i ′ , a, I, H (I)) ve ℘ / ∈ P AY r ise , ℘'yi yeniden sunmalıyım.

Sayfa 13

Resmi Maaş Kümeleri Her tur için, Algorand herkese açık olarak seçer (daha sonra açıklanacak şekilde)
tek (muhtemelen boş) bir maaş seti, P AY r , raundun resmi maaş seti. (Esasen, P AY r temsil eder "gerçekte" gerçekleşmiş olan yuvarlak ödemeler.)
İdeal Sistem'de (ve Bitcoin'de) olduğu gibi, (1) yeni bir j kullanıcısının sisteme girmesinin tek yolu resmi payset P AY ait bir ödeme, alıcı olmak r , belirli bir yuvarlak r; ve
(2) P AY r bir sonraki raundun durumunu, S r + 1 , mevcut raundun durumunu , S r belirler .
Sembolik,
P AY r : S r - → S r + 1 .
Özellikle,
1. yuvarlak r + 1, PK kamu Anahtar dizisi r + 1 , PK birliği oluşur r ve tüm kümesi P AY r ödemelerinde ilk kez görünen alacaklı anahtarları ; ve
2. para miktarı a
(r + 1)
ben
r + 1 turunda sahip olduğum bir kullanıcı, bir i (r) toplamıdır — yani,
önceki turda sahip olduğum para miktarı (0 ise i ∈ PK r) - ve tutarların toplamı P AY r .
Özetle, İdeal Sistemde olduğu gibi, her S r + 1 durumu , önceki ödeme geçmişinden çıkarılabilir:
P AY 0 , ..., ÖDEME r .
2.4 Bloklar ve Kanıtlanmış Bloklar
Algorand 0'da , bir r turuna karşılık gelen B r bloğu şunları belirtir: r'nin kendisi; ödeme seti yuvarlak r, P AY r ; açıklanacak bir Q r miktarı ve önceki bloğun karması, H (B r-1).

Bu nedenle, bazı sabit B 0 bloğundan başlayarak, geleneksel bir blok zincirimiz var:
B 1 = (1, P AY 1 , Q 0 , H (B 0)), B 2 = (2, P AY 2 , Q 1 , H (B 1)), B 3 = (3, P AY 3 , Q 2 , H (B 2)), ...
Algorand'da, bir bloğun gerçekliği aslında ayrı bir bilgi parçasıyla doğrulanır,
B r'yi kanıtlanmış bir bloğa dönüştüren bir "blok sertifika" CERT r , B r . Magic Ledger, bu nedenle, kanıtlanmış bloklar dizisi tarafından uygulanır,
B 1 , B 2 , ...
Tartışma Göreceğimiz gibi, CERT r , H (B r) için bir dizi dijital imzadan oluşur;
SV r üyelerinin çoğunluğu, bu üyelerin her birinin gerçekten ait olduğuna dair bir kanıtla birlikte SV r . Elbette, CERT r sertifikalarını blokların içine dahil edebiliriz, ancak bulabiliriz ayrı tutmak için kavramsal olarak daha temiz.)
Bitcoin'de her blok özel bir özelliği karşılamalıdır, yani “bir çözümün bir çözümünü içermelidir. blok oluşturmayı hesaplama açısından yoğun hale getiren ve her ikisini de kaçınılmaz hale getiren kripto bulmacası ”
ve nadir değil. Aksine, Algorand'ın blok zincirinin iki ana avantajı vardır:
minimum hesaplama ve ezici bir çoğunlukla yüksek olasılıkla çatallaşmayacaktır. Her bir blok B i olan
blok zincirine girer girmez güvenli bir şekilde sonlandırılır.

Sayfa 14

2.5 Kabul Edilebilir Arıza Olasılığı
Algor'un güvenliğini analiz etmek için ve istekli olduğumuz olasılık F'yi belirtiriz.
bir şeylerin ters gittiğini kabul edin (örneğin, bir doğrulayıcı kümesi SV r'nin dürüst bir çoğunluğa sahip olmadığını).
Kriptografik hızlı arama fonksiyonu H'nin çıktı uzunluğu durumunda olduğu gibi, F de bir parametredir.
Ancak, bu durumda olduğu gibi, daha sezgisel bir değer elde etmek için F'yi somut bir değere ayarlamayı yararlı buluyoruz.
Algorand'da aynı anda yeterli güvenlikten yararlanmanın gerçekten mümkün olduğunu kavrayın ve yeterli verimlilik. İlk olarak F'nin istenildiği gibi ayarlanabilen bir parametre olduğunu vurgulamak için
ve sırasıyla belirlediğimiz ikinci düzenlemeler
F = 10-12
ve F = 10-18 .
Tartışma 10-12'nin aslında trilyonda birden az olduğuna dikkat edin ve biz böyle bir
Uygulamamızda F seçimi yeterlidir. 10 -12'nin olasılık olmadığını vurgulayalım
Düşmanın dürüst bir kullanıcının ödemelerini taklit edebileceği. Tüm ödemeler dijitaldir imzalanır ve bu nedenle, uygun dijital imzalar kullanılırsa, bir ödemenin taklit edilme olasılığı 10-12'den çok daha düşüktür ve aslında 0'dır. Tolere etmeye istekli olduğumuz kötü olay
F olasılıkla Algorand'ın blockchain çatalı olmasıdır. Dikkat edin, bizim F ayarımız ve
bir dakikalık uzun turlarda, Algorand'ın blok zincirinde,
(kabaca) 1,9 milyon yılda bir. Buna karşılık, Bitcoin'de çatallar oldukça sık görülür.
Daha talepkar bir kişi F'yi daha düşük bir değere ayarlayabilir. Bu amaçla, ikinci düzenlememizde F'yi 10-18 olarak ayarlamayı düşünüyoruz . Her saniye bir bloğun oluşturulduğunu varsayarsak, 10 18 Evren tarafından şu ana kadar geçen tahmini saniye sayısı: Büyük Patlamadan günümüze zaman. Bu nedenle, F = 10-18 ile , saniyede bir blok üretilirse, kişi yaşını beklemelidir.
Evren bir çatal görmek için.
2.6 Tartışmalı Model
Algorand, çok düşmanca bir modelde güvenli olacak şekilde tasarlanmıştır. Açıklayalım.
Dürüst ve Kötü Amaçlı Kullanıcılar Bir kullanıcı, tüm protokol talimatlarını izlerse dürüsttür ve mükemmel bir şekilde mesaj gönderip alabilir. Bir kullanıcı kötü niyetli (yani, Bizans, dağıtılmış bilgi işlem sözlüğü) kendi öngördüğü talimatlardan keyfi bir şekilde sapabilirse.
The Adversary The Adversary, verimli (teknik olarak polinom zamanlı) bir algoritmadır.
renk aradı, istediği herhangi bir kullanıcıyı istediği zaman anında kötü niyetli hale getirebilir (konu

sadece bozabileceği kullanıcı sayısının üst sınırına).
Adversary, tüm kötü niyetli kullanıcıları tamamen kontrol eder ve mükemmel bir şekilde koordine eder. Tüm eylemleri gerçekleştirir
tüm mesajlarını almak ve göndermek de dahil olmak üzere kendi adına ve sapmalarına izin verebilir keyfi yollarla verdikleri talimatlar. Veya bozuk bir kullanıcının gönderimini izole edebilir.
ve mesaj alma. Bir kullanıcının kötü niyetli olduğunu başka hiç kimsenin otomatik olarak öğrenmediğini açıklığa kavuşturalım,
Her ne kadar benim kötü niyetim, Düşmanın ona yaptırdığı eylemlerle ortaya çıkabilir.
Ancak bu güçlü düşman,
• Sınırsız hesaplama gücüne sahip değildir ve dijital
ihmal edilebilir bir olasılık dışında, dürüst bir kullanıcının imzası; ve

---

Sayfa 15

• Dürüst kullanıcılar arasındaki mesaj alışverişine hiçbir şekilde müdahale edemez.
Dahası, dürüst kullanıcılara saldırma yeteneği aşağıdaki varsayımlardan biri ile sınırlandırılmıştır.
Dürüstlük Paranın Çoğunluğu Paranın Dürüst Çoğunluğunun (HMM) sürekliliğini düşünüyoruz
varsayımlar: yani, her negatif olmayan tamsayı k ve gerçek h> 1/2 için,
HHM k > h: her turdaki dürüst kullanıcılar, tüm paranın h'den daha büyük bir kısmına sahiptiler.
sistem r - k turunda.
Tartışma. Tüm kötü niyetli kullanıcıların eylemlerini mükemmel bir şekilde koordine ettiğini varsayarak (sanki kontrol ediliyormuş gibi)
tek bir varlık tarafından, Düşman) oldukça kötümser bir hipotezdir. Aralarında mükemmel koordinasyon
birçok kişiye ulaşmak zordur. Belki de koordinasyon yalnızca ayrı gruplar içinde gerçekleşir
kötü niyetli oyuncular. Ancak, kötü niyetli kullanıcıların koordinasyon seviyesinden emin olamadığından
zevk alabilir, güvende olsak iyi olur.
Düşmanın gizlice, dinamik olarak ve anında kullanıcıları yozlaştırabileceğini varsayarsak, karamsar. Sonuçta, gerçekçi olarak, bir kullanıcının işlemlerinin tam kontrolünü ele geçirmek biraz zaman almalıdır.
HMM k > h varsayımı , örneğin, bir tur (ortalama olarak) uygulanırsa,
bir dakika içinde, belirli bir turdaki paranın çoğunluğu dürüst ellerde kalacaktır.
k = 120 ise en az iki saat ve k = 10.000 ise en az bir hafta.
HMM varsayımlarının ve önceki Dürüst Bilgi İşlem Gücünün Çoğunluğunun
varsayımlar, bilgi işlem gücü parayla satın alınabildiğinden,
Eğer kötü niyetli kullanıcılar paranın çoğuna sahipse, o zaman bilgi işlem gücünün çoğunu elde edebilirler.

2.7 İletişim Modeli

Mesajın yayılmasını öngörüyoruz - yani, "eşler arası dedikodu"[5] - tek yol olmak
iletişim.
Geçici Varsayım: Tüm Ağda Mesajların Zamanında Teslim Edilmesi. İçin
Bu makalenin çoğu bölümünde, yayılan her mesajın neredeyse tüm dürüst kullanıcılara ulaştığını varsayıyoruz.
zamanında. Bu varsayımı, ağ ile ilgilendiğimiz Bölüm [10'da] kaldıracağız.
ya doğal olarak meydana gelen ya da ters olarak indüklenen bölümler. (Göreceğimiz gibi, sadece varsayıyoruz
Ağın bağlı her bileşeni içinde mesajların zamanında teslimi.)
Yayılan mesajların (ağın tamamında) zamanında teslim edilmesini yakalamanın somut bir yolu, devamındaki:
Tüm erişilebilirlik ρ>% 95 ve mesaj boyutu $\mu \in Z +$ için , $\lambda \rho, \mu$ vardır öyle ki,
Dürüst bir kullanıcı, t zamanında μ baytlık mesaj m'yi yayarsa,
daha sonra m, zamanla $t + \lambda \rho, \mu$, dürüst kullanıcıların en az bir fraksiyonuna ρ ulaşır .

5 Esasen, Bitcoin'de olduğu gibi, bir kullanıcı bir m mesajını yaydığında, ilk defa m aldığım her aktif kullanıcı,
rastgele ve bağımsız olarak, uygun şekilde az sayıda aktif kullanıcıyı, yani "komşularını", m'yi yönlendirdiği,
muhtemelen onlardan bir onay alana kadar. M'nin yayılımı, hiçbir kullanıcı almadığında sona erer ilk defa m.
15

Sayfa 16

Bununla birlikte, yukarıdaki özellik, açıkça ve ayrı ayrı Algorand protokolümüzü destekleyemez.
en son blok zincirini başka bir kullanıcı / depozitör / vb. tarafından elde etmek için bir mekanizma tasarlama.
Aslında, yeni bir blok B oluşturmak için r doğrulayıcı uygun bir dizi zamanında yuvarlak r almalıdır sadece
mesajlar, aynı zamanda önceki turların mesajları, B r-1 ve diğer tüm önceki turları bilmek için
B r'deki ödemelerin geçerli olup olmadığını belirlemek için gerekli olan bloklar . Devamındaki bunun yerine varsayım yeterlidir.
Mesaj Yayılımı (MP) Varsayımı: Tüm ρ>% 95 ve μ ∈ Z + için , λ ρ, μ vardır
öyle ki, her zaman t ve tüm μ-bayt mesajlar için m dürüst bir kullanıcı tarafından t - λ ρ, μ'dan önce yayılır ,
m, t zamanında, dürüst kullanıcıların en azından bir bölümü tarafından alınır.
Protokol Algorand / aslında az sayıda kullanıcının her birine talimat verir (yani, bir
Algorand /'da bir turun adımı verilen (küçük) bir öngörülen boyutta ayrı bir mesaj yaymak için,
ve bu talimatları yerine getirmek için gereken süreyi sınırlamamız gerekiyor. Bunu MP'yi zenginleştirerek yapıyoruz
varsayım aşağıdaki gibidir.
Tüm n, ρ>% 95 ve μ ∈ Z + için, her zaman t ve tüm μ bayt için olacak şekilde λ n, ρ, μ vardır
mesajlar m 1 , ..., m n , her biri dürüst bir kullanıcı tarafından t − λ n, ρ, μ , m 1 , ..., m n alınmadan önce yayılır ,
t zamanına göre, dürüst kullanıcıların en az bir kesri ρ kadar.
Not
• Yukarıdaki varsayım kasıtlı olarak basittir, ancak aynı zamanda makalemizde gerekenden daha güçlüdür.6
• Basit olması için, ρ = 1 olduğunu varsayıyoruz ve böylece ρ'dan bahsetmeyi bırakıyoruz.
• Karamsar bir şekilde, MP varsayımını ihlal etmemesi koşuluyla, Düşman
tüm mesajların teslimini tamamen kontrol eder. Özellikle dürüst tarafından fark edilmeden
Kullanıcılar, Düşman, hangi dürüst oyuncunun hangi mesajı ne zaman alacağına keyfi olarak karar verebilir,
ve istediği herhangi bir mesajın teslimini keyfi olarak hızlandırır.7
3 BA Protokolü BA
⋆
geleneksel bir ortamda
Daha önce de vurgulandığı gibi, Bizans anlaşması Algorand'ın önemli bir bileşenidir. Nitekim, aracılığıyla
Algorand'ın çatallardan etkilenmediği bir BA protokolünün kullanılması. Ancak, bize karşı güvende olmak için
Güçlü Düşman, Algorand, yeni oyuncu değiştirilebilirliğini karşılayan bir BA protokolüne güvenmelidir
kısıtlama. Ek olarak, Algorand'ın verimli olması için böyle bir BA protokolünün çok verimli olması gerekir.
BA protokolleri ilk olarak idealleştirilmiş bir iletişim modeli için tanımlandı, senkronize tam
ağlar (SC ağları). Böyle bir model, BA protokollerinin daha basit bir tasarımına ve analizine izin verir.
6 Dürüst yüzde h ve kabul edilebilir başarısızlık olasılığı F göz önüne alındığında, Algorand bir üst sınırı, N,

bir adımda maksimum doğrulayıcı üye sayısına. Bu nedenle, MP varsayımının yalnızca n ≤ N için geçerli olması gerekir.
Ek olarak, belirtildiği gibi, MP varsayımı, yan yana ne kadar başka mesajın yayılacağına bakılmaksızın geçerlidir.
m j 'ler. Bununla birlikte, göreceğimiz gibi, Algorand'da mesajlar, esasen çakışmayan bir zamanda yayılır.
tek bir bloğun yayıldığı veya en fazla N doğrulayıcının küçük bir (örneğin, 200B) yaydığı aralıklar
İleti. Böylece, MP varsayımını daha zayıf ama aynı zamanda daha karmaşık bir şekilde yeniden ifade edebiliriz.
7 Örneğin dürüst oyuncuların gönderdiği mesajları anında öğrenebilir. Böylece, kötü niyetli bir kullanıcı i ' , kim
her zaman kendi m mesajını seçebilir, dürüst kullanıcı i ile eş zamanlı bir mesaj yaymak istedi ' dayalı mesaj m aslında i tarafından yayılır. Bu yetenek, dağıtılmış hesaplama tabiriyle acele etme ile ilgilidir. Edebiyat.
16

Sayfa 17

Buna göre, bu bölümde, yeni bir BA protokolünü BA tanıtmak ⋆ SC ağlar ve ihmal etmek için tamamen oyuncu değiştirilebilirliği sorunu. BA ⋆ protokolü ayrı bir değerin bir katkısıdır.
Aslında, SC ağları için şimdiye kadar bilinen en verimli kriptografik BA protokolüdür.
Biz BA değiştirmek bizim Algorand protokolü dahilinde kullanmak için ⋆ yani bizim farklı hesaba gelince, biraz
iletişim modeli ve bağlam, ancak X bölümünde BA ⋆'nın nasıl kullanıldığını vurguladığınızdan emin olun
Algorand / gerçek protokolümüz dahilinde .
BA BA'ın faaliyet gösterdiği modeli ve Bizans anlaşması fikrini hatırlayarak başlayalım .
3.1 Eşzamanlı Tam Ağlar ve Eşleşen Rakipler
Bir SC ağında, her integral zamanında r = 1,2, ... işaretleyen ortak bir saat vardır.
Her çift seferde r'ye tıklayın, her bir oyuncu i anında ve aynı anda tek bir
mesaj m r
kendisi dahil her j oyuncuya i, j (muhtemelen boş mesaj). Her m r
i, j alındı
zamanında gönderenin kimliği ile birlikte j oyuncusu tarafından r + 1'e tıklayın i.
Yine, bir iletişim protokolünde, bir oyuncu, öngörülen tüm kurallara uyuyorsa dürüsttür.
talimatlar, aksi takdirde kötü niyetli. Tüm kötü niyetli oyuncular tamamen kontrol edilir ve mükemmeldir
Özellikle adrese gönderilen tüm mesajları anında alan Düşman tarafından koordine edilir.
kötü niyetli oyuncular ve gönderdikleri mesajları seçer.
Düşman, istediği herhangi bir dürüst kullanıcıyı herhangi bir tuhaf tıklama ile anında kötüye kullanabilir.
Sadece kötü niyetli oyuncuların sayısının olası bir üst sınırına tabi olmak istiyor. Yani,
Rakip, "dürüst bir kullanıcı tarafından zaten gönderilmiş mesajlara müdahale edemez",
her zamanki gibi teslim edilir.
Düşman ayrıca, her bir çift turda anında görebilme özelliğine de sahiptir.
Şu anda dürüst oyuncuların gönderdiği ve bu bilgileri anında kullanarak seçim yapmak için kullandığı mesajlar
kötü niyetli oyuncuların aynı anda gönderdiği mesajlar işaretlenir.
Uyarılar
• Düşman Güç. Yukarıdaki ortam çok düşmanca. Nitekim Bizans anlaşmasında
edebiyat, birçok ortam daha az hasımdır. Bununla birlikte, bazı daha çekişmeli ortamlarda
Ayrıca, Dürüst bir oyuncunun gönderdiği mesajları gördükten sonra Düşmanın, i
belirli bir zamanda tıklama r, tüm bu mesajları ağdan anında silme yeteneğine sahiptir
bozuk i, şimdi kötü niyetli olduğum mesajı seçin, r'ye tıklayın ve onlara sahip olun.
her zamanki gibi teslim edilir. Bizim ortamımızda sahip olduğu Rakip maçlarının öngörülen gücü.

• Fiziksel Soyutlama. Öngörülen iletişim modeli daha fiziksel bir modeli özetler,
her oyuncu çiftinin (i, j) ayrı ve özel bir iletişim hattı l i, j ile bağlandığı .
Yani, gönderilen mesajlar hakkında başka hiç kimse enjekte edemez, müdahale edemez veya bilgi alamaz.
l ben, j . Düşmanın l i, j'ye erişmesinin tek yolu , i veya j'yi bozmaktır.
• Gizlilik ve Kimlik Doğrulama. SC ağlarında mesaj gizliliği ve kimlik doğrulama garantilidir varsayımla. Aksine, mesajların yayıldığı iletişim ağımızda
eşler arası, kimlik doğrulama dijital imzalarla garanti edilir ve gizlilik yoktur.
Bu nedenle, BA ⋆ protokolünü bizim ayarımıza uyarlamak için, değiştirilen her mesaj dijital olarak imzalanmalıdır.
(gönderildiği durumu daha ayrıntılı olarak tanımlar). Neyse ki, kullandığımız BA protokolleri Algor'da kullanmayı düşünün ve mesaj gizliliği gerektirmez.

Sayfa 18

3.2 Bizans Anlaşması Kavramı
Bizans anlaşması kavramı, Pease Shostak ve Lamport [31] tarafından
ikili durum, yani her başlangıç değeri bir bitten oluştuğunda. Ancak, hızla uzatıldı
keyfi başlangıç değerlerine. (Fischer [16] ve Chor ve Dwork [10] anketlerine bakın .) BA tarafından protokol, keyfi değerli olanı kastediyoruz.
Tanım 3.1. Senkron bir ağda, P, oyuncu seti ortak olan bir n-oyuncu protokolü olsun.
oyuncular arasında bilgi, $n \geq 2t + 1$ olacak şekilde pozitif bir tam sayıdır. P'nin bir
keyfi değer (sırasıyla ikili) (n, t) - Sağlamlık $\sigma \in (0,1)$ ile Bizans anlaşma protokolü
özel sembolü $\perp$ içermeyen her V değer kümesi için (sırasıyla, $V = \{0,1\}$ için),
Oyuncuların en çok tinin kötü niyetli olduğu ve her oyuncunun bir
başlangıç değeri v i ∈ V, her dürüst oyuncu j 1 olasılıkla durur ve dışarıdan bir değer çıkarır i ∈ V ∪ {⊥}
En az σ olasılıkla aşağıdaki iki koşulu karşılayacak şekilde:
1. Anlaşma: Tüm dürüst oyuncular için i = çıkacak şekilde ∈ V ∪ {⊥} vardır i.
2. Tutarlılık: eğer, tüm dürüst oyuncular için bir v ∈ V değeri için, v i = v ise, o zaman dışarı = v.
Dışarıya P'nin çıktısı ve her bir dışarı i'ye oyuncu i'nin çıktısı olarak atıfta bulunuruz.
3.3 BA Notasyonu #
BA protokollerimizde, bir oyuncunun kendisine verilen bir mesajı kaç oyuncunun gönderdiğini sayması gerekir.
belirli bir adım. Buna göre, gönderilebilecek her olası değer v için,
s
i (v)
(veya s temiz olduğunda sadece # i (v)) adım s'de v'yi aldığım oyuncuların j sayısıdır.
Bir oyuncunun her j oyuncusundan tam olarak bir mesaj aldığını hatırlayarak
oyuncular n'dir, öyleyse, tüm i ve s'ler için, $\sum$ v # s
i (v) = n.
3.4 İkili BA Protokolü BBA ⋆
Bu bölümde , daha fazla dürüstlüğe dayanan yeni bir ikili BA protokolü olan BBA ⋆ sunuyoruz .
oyuncuların üçte ikisinden fazlası ve çok hızlı: kötü niyetli oyuncular ne yaparsa yapsın,
ana döngüsünün her yürütülmesi, oyuncuları 1/3 olasılıkla kabul eder.
Her oyuncunun, benzersiz imzayı karşılayan bir dijital imza şemasının kendi genel anahtarı vardır.
Emlak. Bu protokolün eşzamanlı tam ağda çalıştırılması amaçlandığından,
bir oyuncunun mesajlarının her birini imzalamasına ihtiyacım var.
3. Adımda yeterince yaygın bir rastgele bit oluşturmak için dijital imzalar kullanılır. (Algorand,
Diğer tüm mesajların kimliğini doğrulamak için dijital imzalar kullanılır.)
Protokol asgari bir kurulum gerektirir: oyuncularınkinden bağımsız olarak ortak bir rastgele dizge r anahtarlar. (Algorand'da, r aslında Q r miktarı ile değiştirilir .)
BBA Protokolü ⋆ , oyuncuların Boolean değerlerini defalarca değiş tokuş ettiği ve
farklı oyuncular bu döngüden farklı zamanlarda çıkabilir. Bu döngüden çıkan bir oyuncu,

bir adımda, ya özel bir değer 0 ∗ ya da özel bir değer 1 ∗, böylece tüm oyunculara
İlerideki tüm adımlarda i'den sırasıyla 0 ve 1 aldıklarını "varsayalım". (Alternatif olarak şöyle dedi: varsayalım

---

Sayfa 19

başka bir i oyuncusundan bir j oyuncusu tarafından alınan son mesajın biraz olduğunu b. Sonra, herhangi bir adımda
i, j'den herhangi bir mesaj almadığında, ona b bitini göndermişim gibi davranır.)
Protokol, 3 adımlı döngüsünün kaç kez yürütüldüğünü temsil eden bir sayaç γ kullanır.
BBA'nın başlangıcında ⋆ , γ = 0. (Kişi γ'yi küresel bir sayaç olarak düşünebilir, ancak aslında artmıştır döngü her yürütüldüğünde her bir oyuncu tarafından.)
N ≥ 3t + 1 vardır, burada t, olası maksimum kötü niyetli oyuncu sayısıdır. Bir ikili
string x, ikili gösterimi (0s olası başlangıçlarla) x olan tamsayı ile tanımlanır;
ve lsb (x), x'in en az anlamlı bitini gösterir.
BBA Protokolü ⋆
(İletişim) Adım 1. [Coin-Fixed-To-0 Adım] Her i oyuncuya b i gönderir .
1.1 # 1 ise
i (0) ≥ 2t + 1, sonra i, b i = 0'ı belirler, 0 send gönderir, i = 0 çıkarır ve HALTS.
1.2 Eğer # 1 ise
i (1) ≥ 2t + 1, sonra, i, b ben = 1 olur.
1.3 Aksi takdirde, i = 0 olur.
(İletişim) Adım 2. [Coin-Fixed-To-1 Step] Her i oyuncuya b i gönderir .
2.1 Eğer # 2 ise
i (1) ≥ 2t + 1, sonra i, b i = 1'i belirler, 1 ∗ gönderir, i = 1 çıkarır ve HALTS.
2.2 Eğer # 2
i (0) ≥ 2t + 1, sonra b ben = 0 olarak ayarlıyorum .
2.3 Aksi takdirde, i = 1 olur.
(İletişim) Adım 3. [Madeni Para Olarak Çevrilmiş Adım] Her oyuncu i b i ve SIG i (r, γ) gönderir .
3.1 Eğer # 3
i (0) ≥ 2t + 1, sonra i, b ben = 0 olur.
3.2 Eğer # 3 ise
i (1) ≥ 2t + 1, sonra i, b ben = 1 olur.
3.3 Aksi takdirde, bu adımda uygun bir mesaj gönderen S i = {j ∈ N 3} olsun,
i ayarlar i = c
lsb (min j∈S ben H (SIG i (r, γ))); γ i'yi 1 artırır ; ve 1. Adıma döner.
Teorem 3.1. Her n, ≥ 3t + 1, BBA ⋆ ikili sağlamlığı 1 (n, t) -BA protokolüdür.
Teorem 3. 1'in bir kanıtı [26] ' da verilmiştir. Ortamımıza uyarlanması ve oyuncu tarafından değiştirilebilirliği
mülkiyet yenidir.
Tarihsel Düşünce Olasılıklı ikili BA protokolleri ilk olarak Ben-Or tarafından önerilmiştir.
eşzamansız ayarlar [7]. BBA Protokolü ⋆ , açık anahtar ortamımıza yeni bir uyarlamadır.
Feldman ve Micali'nin ikili BA protokolü [15]. Protokolleri, beklenen bir şekilde çalışan ilk kişiydi.
sabit adım sayısı. Oyuncuların kendilerine ortak bir jeton uygulamalarını sağlayarak işe yaradı,
Bunu harici bir güvenilir parti aracılığıyla uygulayan Rabin tarafından önerilen bir fikir [32] .

Sayfa 20

3.5 Dereceli Konsensus ve Protokol GC
Keyfi değerler için, Bizans anlaşmasından çok daha zayıf bir konsensüs fikrini hatırlayalım.
Tanım 3.2. P, tüm oyuncuların ortak bilgi olduğu bir protokol olsun ve her biri
oyuncu i özel olarak keyfi bir başlangıç değeri biliyor v /
i .
N oyuncuyla yapılan her yürütmede P'nin (n, t) dereceli bir uzlaşma protokolü olduğunu söylüyoruz.

Çoğu t kötü niyetli, her dürüst oyuncu, değer dereceli bir çift çıktısını durdurur (v i , g i),
g i ∈ {0,1,2}, aşağıdaki üç koşulu sağlamak için:
1. Tüm dürüst oyuncular için i ve j, | g i - g j | ≤ 1.
2. Tüm dürüst oyuncular için i ve j, g i , g j > 0 ⇒ v i = v j .
3. Eğer v /
1 = ⋯ = v /
Bir değer v için n = v, sonra tüm dürüst oyuncular i için v i = v ve g i = 2.
Tarihsel Not Dereceli bir fikir birliği kavramı, basitçe derecelendirilmiş bir fikir birliği kavramından türetilmiştir.
Feldman ve Micali tarafından [15] 'te ortaya atılan , haçlı nosyonunu güçlendiren yayın
Dolev [12] tarafından ortaya atılan ve Turpin ve Coan [33] tarafından rafine edilen anlaşma . 8
Olarak [15], Yazarlar ayrıca için, protokol, gradecast yayın -graded (n, t) 3-adım Resim
n ≥ 3t + 1. N> 2t + 1 için daha karmaşık (n, t) dereceli yayın protokolü daha sonra bulundu
Katz ve Koo [19] tarafından .
Aşağıdaki iki adımlı protokol GC, gradecast'in son iki adımından oluşur.
gösterim. Bu gerçeği vurgulamak ve Algorand / protokolünün adımlarını eşleştirmek için bölüm 4.1 , biz
sırasıyla 2 ve 3 GC adımlarını adlandırın.
GC protokolü
Adım 2. Gönderdiğim her oyuncu v /
ben tüm oyunculara.
Adım 3. Her oyuncu i tüm oyunculara x dizisini ancak ve ancak # 2 ise gönderir.
ben (x) ≥ 2t + 1.
Çıktı Belirleme. Her oyuncu i , aşağıdaki gibi hesaplanan çifti (v i , g i) çıkarır:
• Bazı x'ler için # 3
i (x) ≥ 2t + 1, sonra v i = x ve g ben = 2.
• Bazı x'ler için # 3
i (x) ≥ t + 1, sonra v i = x ve g ben = 1.
• Aksi takdirde, v i = ⊥ ve g i = 0.
Teorem 3.2. N ≥ 3t + 1 ise, GC bir (n, t) dereceli yayın protokolüdür.
Kanıt, [15] ' teki protokol derecelendirmesinin hemen ardından gelir ve bu nedenle ihmal edilir. 9
8 Temelde, derecelendirilmiş bir yayın protokolünde, (a) her oyuncunun girişi, seçkin bir oyuncunun kimliğidir.
oyuncu, ek bir özel girdi olarak keyfi bir v değerine sahip gönderen ve (b) çıktılar,
derecelendirilmiş konsensüsün aynı özellikleri 1 ve 2, artı aşağıdaki özellik 3 ′ : gönderen dürüstse, v i = v ve
g i = 2 tüm dürüst oyuncu i için.
9 Aslında, protokolünde, 1. adımda, gönderen kendi özel değerini v tüm oyunculara gönderir ve her oyuncuya
v ′
i aslında aşama 1 'de göndericiden alınan değerin, oluşmaktadır.

Sayfa 21

3.6 Protokol BA ⋆
Biz şimdi keyfi değeri BA protokol BA tarif ⋆ ikili BA protokol BBA yoluyla ⋆ ve
kademeli konsensüs protokolü GC. Aşağıda, her bir oyuncunun başlangıç değeri v /
i .
Protokol BA ⋆
Adım 1 ve 2. i her oyuncu, v / girişinde GC'yi yürütür.
i , bir çift hesaplamak için (v i , g i).
Adım 3, ... Her oyuncu i BBA'yı - ilk giriş 0 ile, eğer g i = 2 ve 1 ise - bu şekilde
biti hesaplamak için i .
Çıktı Belirleme. Her oyuncu i , eğer dışarı i = 0 ise v i , aksi takdirde ⊥ çıktısı verir .

Teorem 3.3. Her n, ≥ 3t + 1, BA ⋆ sağlamlığı 1 ile bir (n, t) -BA protokolüdür.
Kanıt. Önce Tutarlılığı, ardından Anlaşmayı kanıtlarız.
Tutarlılık Kanıtı. Varsayalım ki, bir değer için v value V, v /
i = v. Ardından, 3 özelliğine göre
GC'nin uygulanmasından sonra derecelendirilmiş fikir birliği, tüm dürüst oyuncular çıktı (v, 2). Buna göre, 0
BBA'nın icrasının sonunda tüm dürüst oyuncuların ilk kısmı ⋆ . Böylelikle Anlaşma ile
ikili Bizans anlaşmasının mülkiyeti, BA'nın icrasının sonunda ⋆ , tüm dürüstler için i = 0
oyuncular. Bu, BA ⋆'daki her dürüst oyuncunun çıktısının v i = v olduğu anlamına gelir .
✷
Sözleşme Kanıtı. BBA ⋆ bir ikili BA protokolü olduğundan,
(A) tüm dürüst i oyuncusu için i = 1, veya
(B) tüm dürüst oyuncu i için i = 0.
A maddesinde bütün dürüst oyuncuların çıkış ⊥ ba ⋆ ve böylece Anlaşma tutar. Şimdi B durumunu ele alalım.
bu durumda, BBA ⋆ uygulamasında , en az bir dürüst oyuncu i'nin başlangıç biti 0'dır (Gerçekten, eğer
Dürüst oyuncuların ilk biti 1 idi, daha sonra BBA'nın Tutarlılık özelliğine göre ⋆ ,
out j = 1 tüm dürüstler için j.) Buna göre, GC'nin yürütülmesinden sonra, i (v, 2) çiftini bazıları
için çıkarır .
değer v. Dolayısıyla, derecelendirilmiş konsensüsün 1. özelliğine göre, tüm dürüst oyuncular için g j >
0 j. Buna göre, tarafından
derecelendirilmiş fikir birliğinin 2. özelliği, v j = v tüm dürüst oyuncular için j. Bu, sonunda
BA ⋆ , her dürüst oyuncu j çıktı verir v. Dolayısıyla, Anlaşma B durumunda da geçerlidir.
✷
Hem Tutarlılık hem de Sözleşme geçerli olduğundan, BA ⋆ keyfi değerde bir BA protokolüdür.
Tarihsel Not Turpin ve Coan, n ≥ 3t + 1 için herhangi bir ikili (n, t) -BA
protokol keyfi değerli (n, t) -BA protokolüne dönüştürülebilir. İndirgeme keyfi değer
Kademeli konsensüs yoluyla ikili Bizans anlaşmasına Bizans anlaşması daha modüler ve
daha temiz ve Algorand protokolümüz Algorand / ' in analizini basitleştirir .
Genelleştirilmesi BA ⋆ tüm iletişim yoluyla bile Algorand Algorand kullanılmak üzere çalışır
dedikodu. Bununla birlikte, geleneksel ve tanıdık bir iletişim ağında sunulmasına rağmen,
önceki teknikle daha iyi bir karşılaştırma ve daha kolay bir anlayış sağlamak için,
BA ⋆ protokolü çalışır
dedikodu ağlarında da. Aslında, Algorand'ın ayrıntılı düzenlemelerinde, onu sunacağız.
doğrudan dedikodu ağları için. Ayrıca, oyuncu değiştirilebilirliğini de karşıladığını belirtmeliyiz.
Algorand için öngörülen çok hasım modelde güvende olmak için çok önemli olan mülk.

Sayfa 22

Dedikodu yapan bir iletişim ağında çalışan herhangi bir BA oynatıcı tarafından değiştirilebilir
protokol,
yaratıcı Algorand sistemi içinde güvenli bir şekilde kullanılır. Özellikle Micali ve Vaikunthanatan
Dürüst oyuncuların basit bir çoğunluğuyla da çok verimli bir şekilde çalışmak
için BA ⋆'ı genişletti . Bu
protokol de Algorand'de kullanılabilir.
4 Algorand'ın İki Düzeni
Tartışıldığı gibi, çok yüksek bir seviyede, bir Algorand turu ideal olarak aşağıdaki gibi ilerler. İlk olarak, rastgele
seçilen kullanıcı, lider, yeni bir blok önerir ve dolaştırır. (Bu süreç başlangıçta içerir
birkaç potansiyel lider seçmek ve daha sonra, en azından zamanın önemli bir kısmında,
tek bir ortak lider ortaya çıkar.) İkincisi, rastgele seçilen bir kullanıcı komitesi seçilir ve
liderin önerdiği blok üzerinde Bizans anlaşmasına varır. (Bu süreç şunları içerir:
BA protokolünün her adımı, ayrı olarak seçilen bir komite tarafından yürütülür.) Üzerinde mutabık kalınan blok

daha sonra , komite üyelerinin belirli bir eşiği (T H) ile dijital olarak imzalanır . Bu dijital imzalar Herkesin yeni bloğun olduğundan emin olması için sirküle edilir. (Bu,
İmzalayanların kimlik bilgileri ve yalnızca yeni bloğun karmasını doğrulayarak herkesin
Hash'i netleştirildikten sonra bloğu öğrenmesi garantilidir.)
Sonraki iki bölümde, Algorand'ın iki düzenlemesini sunuyoruz, Algorand /
1 ve Algorand /
2 ,
dürüst kullanıcıların çoğunluğu varsayımı altında çalışır. Bölüm 8'de bunların nasıl benimseneceğini gösteriyoruz
dürüst bir paranın çoğunluğu varsayımı altında çalışmak için somutlaşmış örnekler.
Algorand /
1 sadece komite üyelerinin 2 / 3'ünün dürüst olduğunu öngörür. Ayrıca
Algorand /
1 , Bizans anlaşmasına ulaşmak için gereken adımların sayısı uygun şekilde yüksek bir sınırla sınırlandırılmıştır.
bir sayı, böylece bir anlaşma içinde ezici bir olasılıkla anlaşmaya varılması garanti edilir.
sabit adım sayısı (ancak potansiyel olarak Algorand /
2). İçinde
son adımda henüz anlaşmaya varılamayan uzak bir durumda komite,
her zaman geçerli olan boş blok.
Algorand /
2 , bir komitedeki dürüst üye sayısının her zaman
veya sabit bir eşik t H'ye eşittir (ki bu, çok büyük bir olasılıkla, en azından
Komite üyelerinin 2 / 3'ü dürüsttür). Ek olarak, Algorand /
2 Bizans anlaşmasına izin verir
rastgele bir adım sayısında (ancak potansiyel olarak Algorand /
1).
Bu temel uygulamaların birçok varyantını türetmek kolaydır. Özellikle, verildiğinde kolaydır
Algorand /
2 , Algorand /
1 keyfi bir şekilde Bizans anlaşmasına varmayı sağlamak için
adım sayısı.
Her iki uygulama da aşağıdaki ortak çekirdeği, notasyonları, kavramları ve parametreleri paylaşır.
4.1 Ortak Bir Çekirdek
Hedefler İdeal olarak, her r turu için Algorand aşağıdaki özellikleri sağlar:
1. Mükemmel Doğruluk. Tüm dürüst kullanıcılar aynı blok B r üzerinde hemfikirdir .
2. Tamlık 1. Olasılık 1 ile, B r , P AY r , maksimumdur.10
10 Maaş kümeleri geçerli ödemeleri içerecek şekilde tanımlandığından ve dürüst kullanıcılar yalnızca geçerli ödemeler yapmak üzere
P AY r , tüm dürüst kullanıcıların “şu anda ödenmemiş” ödemelerini içerir.

Sayfa 23

Elbette, tek başına mükemmel doğruluğu garanti etmek önemsizdir: herkes her zaman resmi görevliyi seçer
payset P AY r boş bırakılmalıdır. Ancak bu durumda, sistem bütünlüğe 0 sahip olacaktır. Ne yazık ki,
Hem mükemmel doğruluğu hem de eksiksizliği garanti etmek 1 kötü niyetli kişilerin varlığında kolay değildir
kullanıcılar. Algorand böylece daha gerçekçi bir hedef benimsiyor. Gayri resmi olarak, h'nin yüzdeyi göstermesine izin vermek
Dürüst olan kullanıcıların içinde, h> 2/3, Algorand'ın hedefi
H'ye yakın mükemmel doğruluk ve eksiksizlik, ezici bir olasılıkla garanti eder.
Doğruluğun eksiksizlik yerine ayrıcalıklı olması makul bir seçim gibi görünüyor:
bir tur bir sonrakinde işlenebilir, ancak mümkünse çatallardan kaçınılmalıdır.
Led Bizans Anlaşması Mükemmel Doğruluk aşağıdaki gibi garanti edilebilir. Başlangıçta

r turunda, her kullanıcı i kendi aday bloğunu oluşturur B r
i ve sonra tüm kullanıcılar Bizans'a ulaşır
bir aday blokta anlaşma. Girişimize göre, kullanılan BA protokolü şunları gerektirir:
2/3 dürüst çoğunluk ve oyuncu tarafından değiştirilebilir. Her bir adımı küçük ve
herhangi bir iç değişkeni paylaşmayan rastgele seçilmiş doğrulayıcılar kümesi.
Ne yazık ki, bu yaklaşımın eksiksizlik garantisi yoktur. Bu böyledir çünkü aday
Dürüst kullanıcıların blokları büyük olasılıkla birbirinden tamamen farklıdır. Böylece, nihayetinde
üzerinde mutabık kalınan blok, her zaman maksimal olmayan ödeme setine sahip bir blok
olabilir. Aslında, her zaman
boş blok, B ε , yani payset'i boş olan blok. iyi varsayılan, boş olan.
Algorand / bu tamlık sorununu aşağıdaki gibi önler. İlk olarak, r, ℓ r turu için bir lider seçilir.
Ardından, ℓ r kendi aday bloğu olan B r'yi yayar
ℓ r . Son olarak, kullanıcılar blok üzerinde bir anlaşmaya varırlar
aslında ℓ aldıkları r . ℓ zaman Çünkü r Mükemmel doğruluğu ve bütünlüğü, dürüst
1 her ikisi de tutar, Algorand / ℓ r'nin h'ye yakın olasılıkla dürüst olmasını sağlar . (Lider olduğunda
kötü niyetli ise, üzerinde anlaşmaya varılan bloğun boş bir ödeme setine sahip olması umurumuzda
değil. Sonuçta, bir
kötü niyetli lider ℓ r her zaman kötü niyetle B r'yi seçebilir
ℓ r dürüst sonra boş blok olması, ve
bunu yaymak, böylece dürüst kullanıcıları boş blok üzerinde anlaşmaya zorlar.)
Lider Seçimi Algorand'larda, rth blok B r = (r, P AY r , Q r , H (B r-1) biçimindedir .
Girişte daha önce bahsedildiği gibi, Q r-1 miktarı ,
esasen bizim çok güçlü Düşmanımız tarafından manipüle edilemez. (Bu bölümün ilerleyen
kısımlarında
bunun neden böyle olduğuna dair bir fikir verin.) Bir r turunun başlangıcında, tüm kullanıcılar şunu
bilir:
Şimdiye kadar blok zinciri, B 0 , ..., B r-1 , bundan önceki her turun kullanıcı setini çıkarıyorlar:
PK 1 , ..., PK r-1'dir . R turunun potansiyel bir lideri, kullanıcı i öyle ki
.H (SIG i (r, 1, Q r-1)) ≤ s.
Açıklayalım. Q r-1 miktarı B r-1 bloğunun bir parçası olduğundan ve temelde yatan
imza şeması benzersizlik özelliğini karşılar, SIG i (r, 1, Q r-1) benzersiz bir ikili dizedir
i ve r ile ilişkili. Böylece, H rastgele bir oracle olduğundan, H (SIG i (r, 1, Q r-1)) rastgele bir 256-
bittir
i ve r ile benzersiz şekilde ilişkilendirilmiş uzun dize. Sembol "." H'nin önünde (SIG i (r, 1, Q r-1))
ondalık (bizim durumumuzda ikili) nokta, böylece r i
.H (SIG i (r, 1, Q r-1)) a'nın ikili açılımıdır.
0 ile 1 arasında, i ve r ile benzersiz şekilde ilişkilendirilmiş rasgele 256 bitlik sayı. Böylece olasılık
R ı , esas s veya daha düşük p eşit olmasıdır. (Potansiyel lider seçim mekanizmamız,
Micali ve Rivest'in [28] mikro ödeme planından esinlenilmiştir .)
Olasılık p, ezici (yani 1 - F) olasılıkla en az bir olasılıkla seçilmiştir.
potansiyel doğrulayıcı dürüsttür. (Gerçekteyse, p böyle en küçük olasılık olarak seçilir.)

Sayfa 24

Unutmayın, kendi imzalarını hesaplayabilen tek kişi ben olduğum için, yalnızca kendisi
1. raundun potansiyel bir doğrulayıcısı olup olmadığını belirlemek. Bununla birlikte, kendi ehliyetini
ifşa ederek,
σ r
ben
SIG i (r, 1, Q r-1), r turunun potansiyel bir doğrulayıcısı olduğunu herkese kanıtlayabilirim.
Lider ℓ r , hash edilmiş kimlik bilgisi, daha küçük olan potansiyel lider olarak tanımlanır.
diğer tüm potansiyel liderin karma kimlik bilgisi j: yani, H (σ
r, s
ℓ r) ≤ H (σ
r, s

j).
Not kötü niyetli bir ℓ beri, o r onun kimlik, yuvarlak r doğru liderini açıklayamazsınız may
asla bilinemez ve olası olmayan bağlar dışında, inde r gerçekten de r turunun tek lideridir.
Sonunda son ama önemli bir detayı gündeme getirelim: potansiyel bir lider olabileceğim bir kullanıcı (ve dolayısıyla
Lider) bir r turunun yalnızca sisteme en az k tur için ait olması durumunda. Bu garantiler
Q r'nin ve gelecekteki tüm Q miktarlarının manipüle edilemezliği . Aslında, potansiyel liderlerden biri aslında Q r'yi belirleyecektir .
Doğrulayıcı Seçimi Her bir adım s> 1, küçük bir doğrulayıcı grubu, SV r, s tarafından yürütülür .
Yine, her doğrulayıcı i ∈ SV r, s sistemdeki kullanıcılar arasından rastgele seçilir k turları
r'den önce ve tekrar özel miktar Q r-1 üzerinden . Spesifik olarak, i ∈ PK r-k , SV r, s'de bir doğrulayıcıdır , eğer
.H (SIG i (r, s, Q r-1)) ≤ p / .
Bir kez daha, onun SV r, s'ye ait olup olmadığını sadece ben bilirim , ama eğer durum buysa, bunu şu şekilde kanıtlayabilirdi:
ehliyetini sergileyen σ
r, s
ben
H (SIG i (r, s, Q r-1)). Bir doğrulayıcı i ∈ SV r, s bir mesaj gönderir, m
r, s
ben , içinde
r turunun s adımı ve bu mesaj onun kimlik bilgilerini içerir σ
r, s
i , doğrulayıcıları mümkün kılmak için
o m'yi tanımak için yuva adımı
r, s
ben
meşru bir adım mesajıdır.
P / olasılığı , SV r, s'de # good olmasını sağlayacak şekilde seçilir.
Dürüst kullanıcılar ve kötü kötü niyetli kullanıcıların sayısı, çok büyük bir olasılıkla aşağıdakiler iki koşul geçerlidir.
Uygulama Algorand için /
1 :
(1) # iyi> 2 · # kötü ve
(2) # good + 4 · #bad <2n, burada n, SV r, s'nin beklenen kardinalitesidir .
Uygulama Algorand için /
2 :
(1) # iyi> t H ve
(2) # iyi + 2 # kötü <2t H , burada t H belirtilen bir eşiktir .
Bu koşullar, yeterince yüksek olasılıkla, (a) BA'nın son adımında
protokol, en azından dijital olarak yeni blok B imzalamak için dürüst oyuncu sayısını orada verilecektir r ,
(b) tur başına yalnızca bir blok gerekli sayıda imzaya sahip olabilir ve (c) kullanılan BA
protokol (her adımda) gerekli 2/3 dürüst çoğunluğa sahiptir.
Blok Oluşturmanın Netleştirilmesi Raund lideri ℓ r dürüstse, karşılık gelen blok
formda
B r = (r, P AY r , SIG ℓ r (Q r-1), H (B r-1)),
burada payset P AY r maksimumdur. (tüm maaş setlerinin tanım gereği toplu olarak geçerli olduğunu hatırlayın.)
Başka (yani ℓ eğer r zararlı olan), B r, iki olası formların birine sahiptir:
B r = (r, P AY r , SIG i (Q r-1), H (B r-1))
ve
B r = B r
ε
(r, ∅, Q r-1 , H (B r-1)).

Sayfa 25

İlk formda, P AY r bir (zorunlu olmayan maksimum) ödeme kümesidir ve P AY r = ∅ olabilir; ve ben r turunun potansiyel bir lideri. (Ancak, ben lider ℓ olmayabilir r . Bu gerçekten eğer durumunda olabilecekler
ℓ r kimliğini gizli tutar ve kendini ifşa etmez.)
İkinci biçim, BA protokolünün r turda yürütülmesinde tüm dürüst oyuncular
boş blok B r olan varsayılan değeri verir
ε uygulamamızda. (Tanım gereği, olası
Bir BA protokolünün çıktıları, genel olarak ⊥ ile gösterilen bir varsayılan değer
içerir. Bölüm 3.2'ye bakınız .)
Her iki durumda da maaş kümeleri boş olsa da, B r = (r, ∅, SIG i (Q r-1), H (B r-1))
ve B r
ε sözdizimsel olarak farklı bloklardır ve iki farklı durumda ortaya çıkar: sırasıyla, "tümü
BA protokolünün uygulanmasında yeterince sorunsuz gitti ”ve"
BA protokolü ve varsayılan değer çıktı ”.
Şimdi Algorand / 'nin r turunda B r bloğunun oluşumunun nasıl ilerlediğini sezgisel olarak tanımlayalım .
İlk adımda, uygun olan her oyuncu, yani her oyuncu i ∈ PK r-k , potansiyel olup olmadığını kontrol eder.
Önder. Durum buysa, şu ana kadar gördüğü tüm ödemeleri kullanarak bana sorulur ve
mevcut blok zinciri, B 0 , ..., B r-1 , gizlice bir maksimal ödeme seti hazırlamak için, P AY r
ben ve gizlice
aday bloğunu bir araya getirir, B r = (r, P AY r
ben , SIG ben (Q r-1), H (B r-1)). Yani, sadece o değil
B r'ye dahil et
i , ikinci bileşeni olarak yeni hazırlanmış maaş seti, aynı zamanda üçüncü bileşeni olarak,
Q, kendi imza r-1 , son blok üçüncü bileşeni, B r-1 . Sonunda, kendi
yuvarlak-r-adım-1 mesajı, m
r, 1
i , (a) aday bloğu B r'yi içerir
i , (b) uygun imzası
aday bloğunun (yani, B r hash'inin imzası)
i , ve (c) kendi kimlik bilgisi σ
r, 1
ben , kanıtlıyorum
o gerçekten de r turunun potansiyel bir doğrulayıcısıdır.
(Dürüst biri mesajını verene kadar not edin,
r, 1
i , Düşmanın benim bir
potansiyel doğrulayıcı. Dürüst potansiyel liderleri yozlaştırmak isterse, Düşman da
yozlaşmış rastgele dürüst oyuncular. Ancak, bir kez beni görürse
r, 1
i , i'nin kimlik bilgilerini içerdiğinden,
Düşman, beni bilir ve bozabilir, ancak beni önleyemez
r, 1
Viral olarak çoğaltılan i ,
sistemdeki tüm kullanıcılara ulaşmak.)
İkinci adımda, seçilen her doğrulayıcı j ∈ SV r, 2 turun liderini belirlemeye çalışır.
Özellikle, j adım 1 kimlik bilgilerini alır, σ
r, 1
ben 1
, ..., σ

r, 1
ben n
uygun adım-1 mesajında yer alan m
r, 1
ben
o aldı; hepsinin karması, yani H (σ
r, 1
i 1), ..., H (σ r, 1
i n); kimlik bilgisini bulur,
σ
r, 1
ℓ j
, hash'i sözlükbilimsel olarak minimum olan; ve r'yi düşünür
j r turunun lideri olmak.
Değerlendirilen her kimlik bilgisinin Q r-1'in dijital imzası olduğunu, SIG i (r, 1, Q r-1)
olduğunu hatırlayın.
i ve Q r-1 tarafından benzersiz bir şekilde belirlenir , bu H rastgele oracle'dır ve dolayısıyla her bir H (SIG i (r, 1, Q r-1)
r turunun her bir potansiyel lideri için benzersiz olan rastgele 256 bit uzunluğunda bir dizedir.
Bundan şu sonuca varabiliriz: 256 bitlik Q r-1 dizgisinin kendisi rastgele ve bağımsız bir
şekilde olsaydı
seçiliyse, r. raundun tüm potansiyel liderlerinin karma kimlik bilgileri seçilecekti. Aslında hepsi potansiyel liderler ve onların kimlik bilgileri de iyi tanımlanmıştır (ister gerçekten hesaplanmış ister değil). Ayrıca, r turunun potansiyel liderleri kümesi, raund kullanıcılarının rastgele bir alt kümesidir.
r - k ve dürüst bir potansiyel lider, mesajını her zaman doğru bir şekilde oluşturur ve yayar m r
ben ,
i'nin kimlik bilgilerini içeren. Bu nedenle, dürüst kullanıcıların yüzdesi h olduğu için, ne olursa olsun kötü niyetli potansiyel liderler bunu yapabilir (örneğin, kendi kimlik bilgilerini ifşa edebilir veya gizleyebilir), en azından
karma potansiyel lider kimlik bilgisi, mutlaka herkes tarafından tanınan dürüst bir kullanıcıya aittir raundun lideri ℓ r olmak r. Buna göre, eğer 256 bitlik Q r-1 dizgisinin kendisi rastgele olsaydı ve bağımsız olarak seçilmiş, olasılıkla h (a) lider ℓ r dürüsttür ve (b) ℓ j = ℓ r herkes için
dürüst adım-2 doğrulayıcıları j.
Gerçekte, karma kimlik vardır evet rastgele seçilebilir, ancak S bağlıdır r-1 , olduğu

---

Sayfa 26

rastgele ve bağımsız olarak seçilmez. Biz Q Ancak, analizimizde kanıtlamak zorundadır r-1 olduğu
Bir raund liderinin olasılıkla dürüst olmasını garanti edecek kadar manipüle edilemez
h / h'ye yeterince yakın: yani, h / > h 2 (1 + h - h 2). Örneğin, h =% 80 ise h / > .7424.
(Lider ℓ bunların doğru bir şekilde yapmak turun lideri tespit ettikten r dürüst),
2. adım doğrulayıcılarının görevi, inandıkları şeyleri başlangıç değerleri olarak kullanarak BA'yı yürütmeye başlamaktır.
liderin bloğu olmak. Aslında, gerekli iletişim miktarını en aza indirmek için,
bir doğrulayıcı j ∈ SV r 2 , onun giriş değeri v olarak, kullanmaz /
j Bizans protokol, blok B j o
aslında ℓ j'den almıştır (j kullanıcısı lider olduğuna inanmaktadır), ancak lider, ancak
bu bloğun karması, yani v /
j = H (B ben). Böylece, BA protokolünün sona ermesi üzerine, doğrulayıcılar
Son adımın% 'si istenen round-r bloğu B r'yi hesaplamaz, ancak hesapla (kimlik doğrulama ve çoğaltmak) H (B r). Buna göre, H (B r), yeteri kadar çok sayıda doğrulayıcı tarafından dijital olarak imzalandığından
BA protokolünün son adımında, sistemdeki kullanıcılar H (B r) 'nin yeni

blok. Bununla birlikte, aynı zamanda, çalıştırma oldukça eşzamansız olduğu için almaları (veya beklemeleri) gerekir.
Rakip ne olursa olsun, protokolün gerçekten kullanılabilir olmasını
sağladığı B r'nin kendisini bloke edin.
yapabilir.
Eşzamansızlık ve Zamanlama Algoritması ve /
1 ve Algorand /
2 önemli derecede eşzamansızlığa sahiptir.
Bunun nedeni, Düşmanın, mesajların teslimini planlamada büyük bir serbestliğe sahip olmasıdır.
yayılır. Ek olarak, bir turdaki toplam adım sayısının sınırlı olup olmadığına bakılmaksızın,
varyans, gerçekte atılan adımların sayısına katkıda bulunur.
B 0 , ..., B r-1 sertifikalarını öğrenir öğrenmez , bir kullanıcı i Q r-1'i hesaplar ve çalışmaya başlar.
r turunda, potansiyel bir lider mi yoksa r turunun bazı s adımlarında doğrulayıcı mı olduğunu kontrol eder.
Tartışılan eşzamansızlığın ışığında, adım s'de hareket etmem gerektiğini varsayarak, çeşitli yöntemlere güveniyorum.
harekete geçmeden önce yeterli bilgiye sahip olmasını sağlayacak stratejiler.
Örneğin, aşağıdakilerin doğrulayıcılarından en az belirli sayıda mesaj almayı bekleyebilir.
bir önceki adıma geçebilir veya mesajları yeterince almasını sağlamak için yeterli bir süre bekleyin.
önceki adımın birçok doğrulayıcısı.
Tohum Q r ve Look-Geri Parametre k Hatırlama ideal miktarları Q, o r gerektiği
rastgele ve bağımsız olmasına rağmen, yeterince manipüle edilemez olmaları için yeterli olacaktır.
Düşman.
İlk bakışta, Q r-1'i H (P AY r-1) ile çakışması için seçebilir ve böylece
B r-1'de Q r-1'i açıkça belirtin . Ancak temel bir analiz, kötü niyetli kullanıcıların
bu seçim mekanizmasından yararlanın.[11] Bazı ek çabalar gösteriyor ki, sayısız başkalarının
11 r - 1. raundun başındayız . Dolayısıyla, Q r − 2 = P AY r − 2 herkes tarafından biliniyor ve Düşman özel olarak
kontrol ettiği potansiyel liderlerin kim olduğunu bilir. Düşmanın kullanıcıların% 10'unu kontrol ettiğini varsayın ve
çok yüksek bir olasılıkla, kötü niyetli bir kullanıcı w'nin r - 1 turunun potansiyel lideri olduğunu varsayalım.
H (SIG w (r - 2, 1, Q r − 2)) o kadar küçük ki, son derece olasılık dışı, dürüst bir potansiyel lider aslında
r - 1 turunun lideri (Gizli bir kriptografik sıralama mekanizması aracılığıyla potansiyel liderleri seçtiğimiz için,
Düşman, dürüst potansiyel liderlerin kim olduğunu bilmez.) Bu nedenle, Düşman kıskanılacak durumdadır.
P AY ′ istediği maaş setini seçme pozisyonu ve r - 1 turunun resmi maaşı haline gelmesini sağlayın. Ancak,
daha fazlasını yapabilir. Ayrıca, yüksek olasılıkla (*) kötü niyetli kullanıcılarından birinin lider olmasını sağlayabilir.
aynı zamanda r yuvarlak, böylece P AY r'nin ne olacağını özgürce seçebilir . (Ve bunun gibi. En azından uzun bir süre için, yani
Bu yüksek olasılıklı olaylar gerçekten meydana geldiği sürece.) Garanti etmek için (*), Düşman aşağıdaki gibi davranır. Let P AY '
Düşmanın r - 1 turu için tercih ettiği maaş seti olun. Ardından, H (P AY ′) hesaplar ve bazılarının
Zaten kötü niyetli oyuncu z, SIG z (r, 1, H (P AY ′)) özellikle küçük, yani yeterince küçük, çok yüksek
olasılık z, r turunun lideri olacaktır. Eğer durum buysa, w'ye aday bloğunu seçmesi talimatını verir.

Sayfa 27

geleneksel blok miktarlarına dayanan alternatifler, Adversary tarafından kolayca kullanılabilir.

kötü niyetli liderlerin çok sık olduğu. Bunun yerine markamızı spesifik ve endüktif olarak tanımlıyoruz
Düşman tarafından manipüle edilemez olduğunu kanıtlayabilmek için yeni miktar Q r . Yani,
Q r
H (SIG ℓ r (Q r-1), r), eğer B r boş blok değilse ve Q r
H (Q r-1 , r) aksi halde.
Bu Q r yapısının neden işe yaradığının sezgisi aşağıdaki gibidir. Bir an için varsayalım ki
Q r-1 gerçekten rastgele ve bağımsız olarak seçilir. Öyleyse, Q r olacak mı? ℓ r dürüst olduğunda cevap (kabaca konuşursak) evet. Bu böyledir çünkü
H (SIG ℓ r (·), r): {0,1} 256 - → {0,1} 256
rastgele bir işlevdir. Ancak ℓ r kötü niyetli olduğunda, Q r artık tek sesli olarak Q r-1'den tanımlanmamaktadır.
ve ℓ r . Q r için en az iki ayrı değer vardır . Biri Q r olmaya devam ediyor
H (SIG ℓ r (Q r-1), r),
ve diğeri H (Q r-1 , r). İlk olarak, ikinci seçenek biraz keyfi olsa da, şunu tartışalım:
ikinci bir seçim kesinlikle zorunludur. Bunun nedeni kötü niyetli bir ℓ olmasıdır r hep neden olabilir ikinci adımın dürüst doğrulayıcıları tarafından alınacak tamamen farklı aday bloklar.12 kez
durum böyledir, bloğun nihai olarak BA protokolü aracılığıyla kararlaştırıldığından emin olmak kolaydır.
r yuvarlak varsayılan olacaktır ve bu nedenle hiç kimsenin Q r-1 dijital imzasını içermeyecektir . Fakat sistem devam etmelidir ve bunun için r turu için bir lidere ihtiyacı vardır. Bu lider otomatik olarak ve açıkça seçildiğinde, Düşman onu önemsiz bir şekilde bozacaktır. Bir önceki tarafından seçilmişse Q, r-1 , aynı işlemi ile, ℓ daha r daha yuvarlak r + 1 lider olacaktır. Özellikle öneriyoruz
aynı gizli kriptografik sıralama mekanizmasını kullanın, ancak yeni bir Q-miktarına uygulayın: yani,
H (Q r-1 , r). Bu miktarın H'nin çıktısı olması, çıktının rastgele olmasını garanti eder,
ve H'nin diğer tüm kullanımları bir veya 3+ girişe sahipken, H'nin ikinci girişi olarak r'yi dahil ederek, Böyle bir Q r'nin bağımsız olarak seçilmesini "garanti eder" . Yine, bizim spesifik alternatif
Q r seçimimiz
önemli değil, mesele şu ki ℓ r'nin Q r için iki seçeneği var ve bu yüzden şansını ikiye katlayabilir bir sonraki lider olarak başka bir kötü niyetli kullanıcıya sahip olmak.
Q için seçenekler r bile kötü niyetli bir ℓ kontrol eden düşman için daha çok sayıda olabilir r . Örneğin, x, y ve z, r turunun üç kötü niyetli potansiyel lideri olsun, öyle ki
H (σ r, 1
x) <H (σ r, 1
y) <H (σ r, 1
z)
ve H (σ
r, 1
z) özellikle küçüktür. Yani, o kadar küçük ki, H (σ r, 1
z)
her dürüst potansiyel liderin hash edilmiş kimlik bilgilerinden daha küçük. Sonra, x'den kendi kimlik bilgisine sahipseniz, Düşmanın y'nin r - 1 turunun lideri olma şansı yüksektir. Bu
Q r için başka bir seçeneği olduğunu ima eder : SIG y (Q r-1). Benzer şekilde, Düşman olabilir z'nin r - 1 turunun lideri olması için hem x hem de y'ye kimlik bilgilerini vermemelerini isteyin
ve Q r için başka bir seçenek elde etmek : yani, SIG z (Q r-1).
Tabii ki, ancak, bunların ve diğer seçeneklerin her birinin başarısız olma şansı sıfır değildir, çünkü Düşman, dürüst potansiyel kullanıcıların dijital imzalarının karmasını tahmin edemez.
B r − 1
ben
= (r - 1, P AY ′ , H (B r − 2). Aksi takdirde, yeni bir ödeme oluşturmaya devam etmesi için x ve y başka iki kötü niyetli kullanıcısı var
℘ ′ , birinden diğerine, ta ki bazı kötü niyetli kullanıcılar için z (hatta bazı sabit kullanıcılar için z) H (SIG z (P AY ′ ∪ {℘}))
özellikle de küçük. Bu deney oldukça hızlı bir şekilde sona erecek. Düşman, w'den teklif etmesini istediğinde

aday blok B r − 1
ben
= (r - 1, P AY ′ ∪ {℘}, H (B r − 2).
12 “İkinci adımın zamanı dolmak üzere olduğunda” Örneğin, ℓ, basit (ama aşırı) tutmak için r olabilir her kullanıcıya doğrudan farklı bir aday bloğu B i e-posta ile gönderin. Bu şekilde, 2. adım doğrulayıcıları kim olursa olsun,
tamamen farklı bloklar almış olacak.
27

Sayfa 28

Dikkatli, Markov zinciri benzeri bir analiz, Düşmanın hangi seçenekleri seçtiği önemli değil r − 1 turunda yapmak için, sisteme yeni kullanıcılar enjekte edemediği sürece,
dürüst bir kullanıcının r + 40 turunun lideri olma olasılığı h'nin çok altında. Nedeni bu r - k turunun potansiyel liderlerinin halihazırda r - k turunda bulunan kullanıcılar olmasını talep ediyoruz.
Bu, r −k turunda, Düşmanın şu olasılıkla çok fazla değiştirememesini sağlamanın bir yoludur. dürüst bir kullanıcı r turunun lideri olur. Aslında, ekleyeceği kullanıcılar ne olursa olsun sistem, r − k ila r turlarında, potansiyel liderler olmaya uygun değiller (ve daha sonra lider) r. Böylece geriye bakma parametresi k nihayetinde bir güvenlik parametresidir. (Olmasına rağmen,
Bölüm 7'de göreceğimiz gibi, aynı zamanda bir tür "uygunluk parametresi" de olabilir.)
Geçici Anahtarlar Protokolümüzün yürütülmesi bir çatal oluşturamasa da,
ihmal edilebilir bir olasılık varsa, Düşman, yasal bloktan sonra, r. blokta bir çatal oluşturabilir. blok r oluşturuldu.
Kabaca, B r oluşturulduktan sonra, Düşman her adımın kimlerin doğrulayacağını öğrenmiştir. r yuvarlak. Böylece, hepsini bozabilir ve onları yeni bir bloğu onaylamaya zorlayabilir.
˜
B r . Bu sahte blok ancak meşru olandan sonra yayılabileceğinden,
dikkat etmek aldanmaz.13 Yine de, ˜
B r sözdizimsel olarak doğru olurdu ve biz
imal edilmesini önlemek istiyorum.
Bunu yeni bir kural aracılığıyla yapıyoruz. Esas olarak, doğrulayıcı grubu SV üyeleri r, s adım s yuvarlak r kullanımı geçici genel anahtarlar pk
r, s
ben
mesajlarını dijital olarak imzalamak için. Bu anahtarlar tek
yalnızca kullanım ve bunlara karşılık gelen gizli anahtarlar sk
r, s
ben
kullanıldıktan sonra yok edilir. Bu şekilde, eğer bir doğrulayıcı ise
Daha sonra bozulursa, Düşman onu başlangıçta imzalamadığı başka herhangi bir şeyi imzalamaya zorlayamaz.
Doğal olarak, Düşmanın yeni bir anahtar hesaplamasının imkansız olmasını sağlamalıyız ˜
p
r, s
ben
ve dürüst bir kullanıcıyı, adım s'de kullanılacak doğrulayıcı i ∈ SV r'nin doğru geçici anahtarı olduğuna ikna edin .

4.2 Gösterimler, Kavramlar ve Parametrelerin Ortak Özeti

Notasyonlar

- r ≥ 0: mevcut tur numarası.
- s ≥ 1: r turundaki geçerli adım numarası.
- B r : r turunda oluşturulan blok.
- PK r : r - 1 turunun sonunda ve r turunun başlangıcında açık anahtarlar kümesi.

• S r : r - 1 turunun sonunda ve r turunun başlangıcındaki sistem durumu.14
• P AY R : B içerdiği payset r .
• ℓ r : r tur lideri. ℓ r , r turunun P AY r ödeme setini seçer (ve bir sonraki Q r'yi belirler).
• Q r : r turunun tohumu, r turunun sonunda üretilen bir miktar (yani ikili dizi)
ve r + 1 turu için doğrulayıcıları seçmek için kullanılır. Q r bloklardaki maaş setlerinden bağımsızdır ve ℓ r tarafından değiştirilemez .
13 Büyük bir TV ağının haber sunucusunu bozmayı ve bugün bir haber filmi üretip yayınlamayı düşünün.
Bakan Clinton'ın son başkanlık seçimlerini kazandığını gösteriyor. Çoğumuz bunu bir aldatmaca olarak kabul ederdik. Fakat
komadan çıkan biri kandırılabilir.
14 Senkronize olmayan bir sistemde, "r - 1 turunun sonu" ve "r turunun başlangıcı" kavramı dikkatlice tanımlanması gerekir. Matematiksel olarak, PK r ve S r , S 0 başlangıç durumundan ve bloklardan hesaplanır.
B 1 , ..., B r − 1 .

Sayfa 29

• SV r, s : r turunun s adımları için seçilen doğrulayıcılar kümesi.
• SV r : r turu için seçilen doğrulayıcılar kümesi, SV r = ∪ s≥1 SV r, s .
• MSV r, s ve HSV r, s : sırasıyla kötü niyetli doğrulayıcılar ve dürüst doğrulayıcılar kümesi
SV r, s . MSV r, s ∪ HSV r, s = SV r, s ve MSV r, s ∩ HSV r, s = ∅.
• n 1 ∈ Z + ve n ∈ Z + : sırasıyla, her SV'de beklenen potansiyel lider sayısı r, 1 ,
ve s> 1 için her SV r, s'de beklenen doğrulayıcı sayısı .
N 1 << n olduğuna dikkat edin , çünkü SV r'de en az bir dürüst üyeye ihtiyacımız var , 1 , ama en azından
her SV r'deki dürüst üyelerin çoğunluğu, s> 1.
• h ∈ (0,1): 2 / 3'ten büyük bir sabit. h sistemdeki dürüstlük oranıdır. Yani
Her PK içinde varsayımına bağlı olarak dürüst kullanıcılara veya dürüst para, kesir, kullanılan r olan en azından h.
• H: rastgele oracle olarak modellenen bir kriptografik karma işlevi.
• ⊥: H'nin çıktısı ile aynı uzunlukta özel bir dizi.
• F ∈ (0,1): izin verilen hata olasılığını belirten parametre. Bir olasılık ≤ F
"ihmal edilebilir" ve ≥ 1 - F olasılığı "ezici" olarak kabul edilir.
• p h ∈ (0,1): r, ℓ r turunun liderinin dürüst olma olasılığı . İdeal olarak p h = h. İle
Düşmanın varlığı , analizde p h'nin değeri belirlenecektir.
• k ∈ Z + : geriye dönük parametre. Yani, r - k turu, r turu için doğrulayıcıların olduğu yerdir.
- yani SV r ⊆ PK r-k arasından seçilir .15
• p 1 ∈ (0,1): r turunun ilk adımı için, r - k turundaki bir kullanıcı SV r, 1 ile
olasılık p 1
n 1
| PK r − k |
.
• p ∈ (0,1): r turunun her s> 1 adımı için, r - k turundaki bir kullanıcı SV r, s ile seçilir
olasılık p
n
| PK r − k |
.
• CERT r : B r sertifikası . H (B r) ' nin bir dizi t H imzasıdır.
yuvarlak r.
• B r
(B r , CERT r) kanıtlanmış bir bloktur.
Bir kullanıcı , kanıtlanmış bloğun her iki parçasına da sahip olup olmadığını (ve başarıyla doğrularsa) B r'yi bilir .

Farklı kullanıcılar tarafından görülen CERT r farklı olabilir.
• τ r
i : B r'yi tanıdığım bir kullanıcının (yerel) saati . Algorand protokolünde her kullanıcının kendi kendi saati. Farklı kullanıcıların saatlerinin senkronize edilmesine gerek yoktur, ancak aynı hıza sahip olmalıdır.
Sadece analizin amacı için, bir referans saati düşünüyoruz ve oyuncuların
onunla ilgili zamanlar.
• α
r, s
ben
ve β
r, s
i : sırasıyla bir kullanıcı i'nin Adım s'yi çalıştırmaya başladığı ve sona erdirdiği (yerel) zaman. yuvarlak r.
• Λ ve λ: esasen, sırasıyla Adım 1'i yürütmek için gereken zamanın üst sınırları ve
Algorand protokolünün diğer herhangi bir adımı için gereken süre.
Parametre Λ, tek bir 1MB bloğu yaymak için gereken zamanı sınırlar. (Gösterimimizde,
Λ = λ ρ, 1MB . Gösterimimizi hatırlayarak, basitlik için ρ = 1 olarak belirlediğimizi ve blokların en fazla 1MB uzunluğunda seçildiyse , Λ = λ 1,1,1MB'ye sahibiz .)
15 Kesin konuşmak gerekirse, “r - k” “max {0, r - k}” olmalıdır.

Sayfa 30

Λ parametresi, Adım s> 1'de doğrulayıcı başına bir küçük mesajı yaymak için gereken süreyi üst sınırlar.
(Bitcoin'de olduğu gibi, 32B anahtarlı eliptik eğri imzaları kullanıldığında, bir doğrulayıcı mesajı 200B uzunluğundadır.
Böylece, gösterimimizde, λ = λ n, ρ, 200B .)
Λ = O (λ) olduğunu varsayıyoruz.
Kavramlar
• Doğrulayıcı seçimi.
Her bir r ve adım s> 1 için, SV r, s
{i ∈ PK r-k : .H (SIG i (r, s, Q r-1)) ≤ p}. Her biri
kullanıcı i ∈ PK r-k imzasını uzun vadeli anahtarını kullanarak özel olarak hesaplar ve
i ∈ SV r, s veya değil. Eğer i ∈ SV r, s ise , o zaman SIG i (r, s, Q r-1) i'nin (r, s) -credential'tır, kısaca ifade edilir
σ tarafından
r, s
i .
Turun ilk adımı için r, SV r, 1 ve σ
r, 1
ben
benzer şekilde tanımlanır, p, p 1 ile değiştirilir . The
SV r, 1'deki doğrulayıcılar potansiyel liderlerdir.
• Lider seçimi.
Kullanıcı i ∈ SV r 1 ℓ ile gösterilen yuvarlak r lideri r , eğer, H (σ
r, 1
i) ≤ H (σ
r, 1
j) tüm potansiyel için
liderler j ∈ SV r, 1 . İki oyuncunun kimlik bilgilerinin karmaları karşılaştırıldığında
bağlar olayı, protokol bağları her zaman sözlükbilimsel olarak (uzun vadeli kamuya açık anahtarlar) potansiyel liderler.
Tanım olarak, oyuncu ℓ hash değeri r kimlik s’de tüm kullanıcılar arasında en küçüğüdür

PK r-k . Potansiyel bir liderin, lider olup olmadığına özel olarak karar veremeyeceğini unutmayın, diğer potansiyel liderlerin kimlik bilgilerini görmeden.
Karma değerleri rastgele tekdüze olduğundan, SV r, 1 boş olmadığında, ℓ r her zaman vardır ve en azından olasılıkla dürüst h. N 1 parametresi , her birinin
SV r, 1 çok büyük bir olasılıkla boş değildir.
• Blok yapısı.
Boş olmayan bir blok B r = (r, P AY r , SIG ℓ r (Q r-1), H (B r-1)) biçimindedir ve boş bir blok B r biçimindedir
ϱ = (r, ∅, Q r-1 , H (B r-1)).
Boş olmayan bir bloğun, içinde ödeme yapılmazsa, boş bir ödeme seti P AY r içerebileceğini unutmayın .
bu tur ya da lider kötü niyetli ise. Bununla birlikte, boş olmayan bir blok, kimliğinin
ℓ r , kimlik bilgisi σ
r, 1
ℓ r
ve SIG ℓ r (Q r-1) hepsi zamanında ifşa edildi. Protokol garanti eder
eğer lider dürüstse, o zaman blok çok büyük bir olasılıkla boş olmayacaktır.
• Tohum Q r .
B r boş değilse , Q r
H (SIG ℓ r (Q r-1), r), aksi takdirde Q r
H (Q r-1 , r).
Parametreler
• Çeşitli parametreler arasındaki ilişkiler.
- R raundunun doğrulayıcıları ve potansiyel liderleri, PK r-k'deki kullanıcılardan seçilir , burada k, R - k - 1 turunda Rakip, Q r-1'i geri tahmin edemeyecek şekilde seçilir.
F'den daha iyi bir olasılıkla: aksi takdirde, kötü niyetli kullanıcılar
r - k turu için, bunların tümü r turunda potansiyel liderler / doğrulayıcılar olacak ve

---

Sayfa 31

zararlı bir lider ya da SV bir kötü amaçlı çoğunluğa sahip , r s istenen bazı adımlar için s onu.
- Her turun 1. Adımında r, n 1 seçilmiştir, öyle ki ezici bir olasılıkla, SV r, 1 = ∅.
• Önemli parametrelerin örnek seçimleri.
- H'nin çıktıları 256 bit uzunluğundadır.
- h =% 80, n 1 = 35.
- Λ = 1 dakika ve λ = 10 saniye.
• Protokolün başlatılması.
Protokol 0 zamanında r = 0 ile başlar. “B -1 ” veya “CERT -1 ” olmadığı için,
sözdizimsel olarak B -1 , üçüncü bileşeni Q -1'i belirten genel bir parametredir ve tüm kullanıcılar 0 anında B -1'i bilir .
5 Algorand ′
1
Bu bölümde, aşağıdaki varsayım altında Algorand / çalışma versiyonunun bir versiyonunu oluşturuyoruz .
Kullanıcıların Dürüst Çoğunluğu Varsayımı: Her PK r'deki kullanıcıların 2 / 3'ünden fazlası dürüsttür.
Bölüm 8'de , yukarıdaki varsayımı istenen Dürüst Çoğunluk ile nasıl değiştireceğimizi gösteriyoruz.
Para varsayımı.
5.1 Ek Gösterimler ve Parametreler
Notasyonlar
• m ∈ Z + : ikili BA protokolündeki maksimum adım sayısı, 3'ün katı.
• L r ≤ m / 3: 1'i görmek için gereken Bernoulli denemelerinin sayısını temsil eden rastgele bir değişken,
her deneme 1 olasılıkla olduğunda

p h
2
ve en fazla m / 3 deneme vardır. Tüm denemeler başarısız olursa o zaman
L r
m / 3. L r , blok B r'yi oluşturmak için gereken süreyi üst sınırlamak için kullanılacaktır .
• t H = 2n
3
+ 1: protokolün bitiş koşullarında ihtiyaç duyulan imza sayısı.
• CERT r : B r sertifikası . H (B r) ' nin bir dizi t H imzasıdır.
yuvarlak r.
Parametreler
• Çeşitli parametreler arasındaki ilişkiler.
- R turunun her s> 1 adımı için n seçilir, öyle ki, çok büyük bir olasılıkla,
| HSV r, s | > 2 | MSV r, s |
ve
| HSV r, s | + 4 | MSV r, s | <2n.
H'nin değeri 1'e ne kadar yakınsa, n'nin o kadar küçük olması gerekir. Özellikle, kullanıyoruz
(varyantlar
of) İstenilen koşulların ezici bir olasılıkla geçerli olmasını sağlamak için Chernoff sınırları.
- m, çok büyük olasılıkla L r <m / 3 olacak şekilde seçilir .
• Önemli parametrelerin örnek seçimleri.
- F = 10-12 .
- n ≈ 1500, k = 40 ve m = 180.

Sayfa 32

5.2 Algorand'de Geçici Anahtarları Uygulama
′
1
Daha önce de belirtildiği gibi, bir doğrulayıcı i ∈ SV r, s'nin mesajını dijital olarak imzalamasını
diliyoruz.
r, s
ben
adım
r turunda, geçici bir genel anahtara göre pk
r, s
i , geçici bir gizli anahtar sk kullanarak
r, s
ben
o
o kullandıktan sonra derhal yok eder. Bu nedenle, her kullanıcının yapabilmesini sağlamak için
verimli bir yönteme ihtiyacımız var.
pk'yi doğrula
r, s
ben
gerçekten de i'nin m imzasını doğrulamak için kullanılacak anahtardır
r, s
i . Bunu a (en iyisine) yapıyoruz
bilgimiz) kimlik tabanlı imza şemalarının yeni kullanımı.
Yüksek düzeyde, böyle bir şemada, merkezi bir otorite A, bir genel ana anahtar, PMK,
ve karşılık gelen bir gizli ana anahtar, SMK. Bir U, A oyuncusunun kimliği, U hesaplanırsa,
SMK aracılığıyla , genel anahtar U ile ilgili bir gizli imza anahtarı sk U ve özel olarak sk U verir .
U. (Aslında, kimlik tabanlı bir dijital imza şemasında, bir U kullanıcısının genel anahtarı U'nun
kendisidir!)

Bu şekilde, eğer A, etkinleştirmek istediği kullanıcıların gizli anahtarlarını hesapladıktan sonra SMK'yı yok ederse,
dijital imzalar üretir ve herhangi bir hesaplanmış gizli anahtarı tutmazsa, U
açık anahtar U ile ilgili mesajları dijital olarak imzalayabilir. Böylece, "U'nun adını" bilen herkes, otomatik olarak U'nun genel anahtarını tanır ve böylece U'nun imzalarını doğrulayabilir (muhtemelen genel ana anahtar PMK).
Uygulamamızda, A yetkisi i kullanıcısıdır ve tüm olası kullanıcılar kümesi U ile çakışmaktadır.
—say— S = {i} × {r / , ..., r / + 10 6 } × {1, ..., m + 3} ' deki yuvarlak adım çifti (r, s) , burada
r / verilen
yuvarlak ve m + 3 bir turda meydana gelebilecek adım sayısının üst sınırı. Bu
yol, pk
r, s
ben
(i, r, s), böylece herkes i'nin imzasını gören SIG
r, s
pk r, s
ben
(m
r, s
i) ezici bir şekilde
olasılık, derhal r / 'yi takip eden ilk bir milyon raund için doğrulayın .
Diğer bir deyişle, önce PMK ve SMK üretiyorum. Ardından, PMK'nın benim ustası olduğunu duyurdu herhangi bir raund r ∈ [r / , r / + 10 6] için genel anahtar ve sırrı özel olarak üretmek ve saklamak için SMK kullanır
anahtar sk
r, s
ben
her üçlü için (i, r, s) ∈ S. Bu yapılırsa, SMK'yı yok eder. Olmadığını belirlerse
SV r'nin bir bölümü , s , sonra sk'yi terk edebilirim
r, s
ben
tek başına (protokol onun kimlik doğrulamasını yapmasını gerektirmediğinden
r) adımının s adımlarındaki herhangi bir mesaj. Aksi takdirde, önce sk kullanırım
r, s
ben
mesajını dijital olarak imzalamak m
r, s
ben ve
sonra skleri yok eder
r, s
i .
Sisteme ilk girdiğinde ilk genel ana anahtarını yayınlayabileceğimi unutmayın. Yani,
(yuvarlak r sisteme i getiren aynı ödeme ℘ / veya r yuvarlak bitiminde /) da olabilir
i her tur r ∈ [r için ortak bir ana anahtar olduğunu, i'nin talebi belirtmek / r / + 10 6] ile PMK -örn olduğu
şeklinde olan bir çift de dahil olmak üzere (PMK, [r / r / + 10 6]).
Ayrıca, m + 3 bir turdaki maksimum adım sayısı olduğundan, bir tur olduğunu varsayarak
bir dakika sürer, bu şekilde üretilen geçici anahtarların zulası neredeyse iki yıl dayanacaktır. Aynı zaman, bu geçici gizli anahtarları üretmem çok uzun sürmeyecek. Eliptik eğri tabanlı bir kullanım 32B anahtarlı sistemde, her gizli anahtar birkaç mikrosaniye içinde hesaplanır. Böylece, m + 3 = 180 ise,
daha sonra tüm 180M gizli anahtarlar bir saatten daha kısa sürede hesaplanabilir.
Geçerli yuvarlak r yaklaşıyoruz zaman / + 10 6 sonraki milyon mermi ele, i
yeni bir (PMK / , SMK /) çifti oluşturur ve bir sonraki geçici anahtar zulasının ne olduğunu bildirir
—Örneğin— SIG i (PMK / , [r / + 10 6 + 1, r / + 2 · 10 6 + 1]) yeni bir blok girerek, ya bir

ayrı "işlem" veya bir ödemenin parçası olan bazı ek bilgiler olarak. Bunu yaparak,
Herkese PMK / bir sonraki adımda i'nin geçici imzalarını doğrulamak için kullanması gerektiğini bildiririm
milyon mermi. Ve bunun gibi.
(Bu temel yaklaşımı izleyerek, geçici anahtarları kullanmadan uygulamanın diğer yollarını unutmayın. kimlik tabanlı imzalar kullanmak kesinlikle mümkündür. Örneğin, Merkle ağaçları aracılığıyla.16)
16 Bu yöntemde, bir genel-gizli anahtar çifti oluşturur (pk r, s
ben
, sk r, s
ben
) her bir yuvarlak adım çifti (r, s) için —say— içinde

Sayfa 33

Geçici anahtarları uygulamanın başka yolları da kesinlikle mümkündür - örneğin, Merkle ağaçları yoluyla.
5.3 Algorand Adımlarını Eşleştirme ′
1
BA olanlarla ⋆
Söylediğimiz gibi, Algorand'da bir tur /
1 en fazla m + 3 adıma sahiptir.
Adım 1. Bu adımda, her bir potansiyel lider i kendi aday bloğu B r'yi hesaplar ve yayar.
ben ,
kendi kimlik bilgileriyle birlikte, σ
r, 1
i .
Bu kimlik bilgilerinin açıkça i'yi tanımladığını hatırlayın. Bu böyledir çünkü σ
r, 1
ben
SIG i (r, 1, Q r-1).
Potansiyel doğrulayıcı i, mesajının bir parçası olarak, H (B r
i).
Bir ödeme veya kimlik belgesi ile uğraşmayan, bu i imzası, geçici kamuoyuna göredir.
anahtar pk
r, 1
i : yani sig yayar
pk r, 1
ben
(H (B r
i)).
B r'yi yaymak yerine, geleneklerimiz göz önüne alındığında
ben ve sig
pk r, 1
ben
(H (B r
i)) olabilirdi
yayılmış SIG
pk r, 1
ben
(H (B r
i)). Ancak, analizimizde açık erişime sahip olmamız gerekir.
sig
pk r, 1
ben

(H (B r
i)).
Adımlar 2. Bu adımda, her doğrulayıcı, ℓ r
i hash edilmiş kimlik bilgisine sahip potansiyel lider olmak
en küçüğü ve B r
i ℓ önerdiği blok için r
i . Verimlilik adına, biz
Doğrudan B r yerine H (B r) üzerinde anlaşmak isterim , alacağı mesajı yayar
BA ⋆'nin ilk adımında başlangıç değeri v / ile yayılır
i = H (B r
i). Yani, v /
ben ,
elbette geçici olarak imzaladıktan sonra. (Yani, doğru geçici ile ilgili olarak imzaladıktan sonra
açık anahtar, bu durumda pk
r, 2
i .) Tabii ki ben de kendi kimlik belgesini iletiyorum.
BA ⋆'nın ilk adımı, derecelendirilmiş konsensüs protokolü GC'nin ilk adımını içerdiğinden, Adım
Algorand 2 / GC birinci aşamasına denk düşer.
Adımlar 3. Bu adımda, her doğrulayıcı i ∈ SV r, 2 BA ⋆'nın ikinci adımını yürütür . Yani o gönderir
GC'nin ikinci adımında da göndereceği mesajın aynısı. Yine, i'nin mesajı geçicidir
imzalı ve i'nin kimlik bilgileriyle birlikte. (Şu andan itibaren, bir doğrulayıcı olduğunu söylemeyi
ihmal edeceğiz
Mesajını geçici olarak imzalar ve aynı zamanda kimlik bilgilerini yayar.)
Adım 4. Bu adımda, her doğrulayıcı i ∈ SV r, 4 , GC çıktısını (v i , g i) ve geçici olarak hesaplar
belirti ve o BA üçüncü adımda göndermiş aynı mesajı gönderir ⋆ içinde olduğu,
BBA'nın ilk adımı ⋆ , başlangıç biti g i = 2 ise 0 , aksi takdirde 1.
Adım s = 5, ..., m + 2. Böyle bir adım, eğer ulaşılırsa, BA ⋆'nın s − 1 adımına karşılık gelir ve
dolayısıyla
Adım s - BBA'nın 3'ü ⋆ .
Yayılma modelimiz yeterince eşzamansız olduğundan, olasılığı hesaba katmalıyız
Böyle bir adımın ortasında, bir doğrulayıcıya , onu kanıtlayan bilgilerle ulaşılır.
bu B r bloğu zaten seçilmiştir. Bu durumda, kendi r turunu kendi yürütmesini durdururum.
Algorand / ve round- (r + 1) talimatlarını uygulamaya başlar.
{r ′ , ..., r ′ + 10 6 } × {1, ..., m + 3}. Sonra bu genel anahtarları kanonik bir şekilde sipariş eder, jth
public anahtarlarını saklar.
Bir Merkle ağacının j. yaprağındaki anahtar ve halka duyurduğu R i kök değerini hesaplar . İmzalamak
istediğinde
pk r, s anahtarına göre bir mesaj
ben
, i sadece gerçek imzayı sağlamakla kalmaz, aynı zamanda pk r, s için kimlik doğrulama yolunu da
sağlar.
ben
R i'ye göre . Bu kimlik doğrulama yolunun aynı zamanda pk r, s
ben
j. yaprakta saklanır. Gerisi
detaylar kolaylıkla doldurulabilir.

Sayfa 34

Buna göre, ilgili talimatlara ek olarak bir doğrulayıcı i ∈ SV r, s'nin talimatları
3 BBA - Aşama s ⋆ , BBA yürütülmesi kontrol içerir ⋆ önceki olarak durdurdu
Adım s / . BBA ⋆ sadece 0'a Sabit Madeni Parayla veya 1'e Sabit Madeni Para ile
durdurulabildiğinden,
talimatlar ayırt eder

A (Bitiş Koşulu 0): s / - 2 ≡ 0 mod 3 veya
B (Bitiş Koşulu 1): s / - 2 ≡ 1 mod 3.
Aslında, A durumunda, B r bloğu boş değildir ve bu nedenle ek talimatlar gereklidir.
B r'yi uygun sertifikası CERT r ile birlikte uygun şekilde yeniden yapılandırdığımdan emin olun . B durumunda,
B r bloğu boştur ve bu nedenle i'ye B r = B r
ε = (r, ∅, H (Q r-1 , r), H (B r-1)),
ve CERT r hesaplamak için .
S adımının uygulanması sırasında, B r bloğunun halihazırda sahip olduğuna dair herhangi bir kanıt görmezsem ,
oluşturulduktan sonra BBA ⋆ s - 3 adımında göndereceği mesajın aynısını gönderir .
Adım m + 3. m + 3. Adım sırasında i ∈ SV r, m + 3 , B r bloğunun şu anda oluşturulmuş olduğunu görürse
bir önceki adım s / , daha sonra yukarıda açıklandığı gibi ilerler.
Else, daha doğrusu o BBA adım m göndermiş aynı mesajı göndererek ⋆ , i
sahip olduğu bilgilere dayanarak, B r'yi ve karşılık gelen
sertifika CERT r .
Aslında, bir raundun toplam adım sayısını m + 3 ile yukarı sınırladığımızı hatırlayın.
5.4 Gerçek Protokol
Bir r turunun her adımında, bir doğrulayıcı i ∈ SV r'nin uzun vadeli genel-gizli anahtar çiftini kullandığını hatırlayın.
kimlik bilgilerini üretmek için, σ
r, s
ben
SIG i (r, s, Q r-1) ve s = 1 durumunda SIG i (Q r-1). Doğrulayıcı i
geçici gizli anahtar skini kullanır
r, s
ben
(r, s) -message m imzalamak
r, s
i . Basitlik için, r ve s olduğunda
açık, sig pk r, s yerine esig i (x) yazıyoruz
ben
(x) i'nin bir değerin uygun geçici imzasını belirtmek için
r turunun s adımlarında x ve SIG pk r, s yerine ESIG i (x) yazın
ben
(x), (i, x, esig i (x)) anlamına gelir.
1. Adım: Teklifi Engelleyin
Her kullanıcı için talimatlar i ∈ PK r-k : Kullanıcı i, kendi r turunun kendi 1. Adımını başlatır.
B r-1'i bilir .
• Kullanıcı i , B r-1'in üçüncü bileşeninden Q r-1'i hesaplar ve i ∈ SV r, 1 veya
değil.
• Eğer i / ∈ SV r, 1 ise , o zaman i Adım 1'in kendi yürütmesini hemen durdurur.
• i ∈ SV r, 1 ise, yani i potansiyel bir lider ise,
şimdiye kadar kendisine iletildi ve maksimum maaş seti P AY r
ben onlardan. Sonra o
"aday bloğu" B r'yi hesaplar
i = (r, P AY r
ben , SIG ben (Q r-1), H (B r-1)). Sonunda hesaplar
mesaj m
r, 1
ben
= (B r
ben , esig ben (H (B r
i)), σ

r, 1
i) geçici gizli anahtarını yok eder
r, 1
ben ve sonra
m'yi yayar
r, 1
i .
34

Sayfa 35

Açıklama. Uygulamada, Adım 1'in küresel uygulamasını kısaltmak için, (r, 1) -
mesajlar seçilerek yayılır. Yani, sistemdeki her i kullanıcısı için ilk (r, 1) -
aldığı ve başarıyla doğruladığı mesaj,17 oyuncu, her zamanki gibi yayıyor. Hepsi için
diğer (r, 1) -mesajları aldığım ve başarılı bir şekilde doğruladığım oyuncunun, yalnızca hash
içerdiği kimlik bilgilerinin değeri, içerdiği kimlik bilgilerinin karma değerleri arasında en küçük
olanıdır
Şimdiye kadar aldığı ve başarıyla doğruladığı tüm (r, 1) -mesajlarında. Ayrıca, önerildiği gibi
Georgios Vlachos'a göre, her bir potansiyel lider i'nin kendi kimlik bilgilerini de yayması yararlıdır σ
r, 1
ben
ayrı olarak: bu küçük mesajlar bloklardan daha hızlı hareket eder, m'nin zamanında yayılmasını sağlar.
r, 1
j 's
içerilen kimlik bilgilerinin küçük karma değerlere sahip olduğu, büyük karma değerlere sahip olanların
olduğu yerlerde
hızla kaybolur.
Adım 2: Dereceli Konsensüs Protokolü GC'nin İlk Adımı
Her kullanıcı için talimatlar i ∈ PK r-k : Kullanıcı i, kendi r turunun 2. Adımını başlatır.
B r-1'i bilir .
• Kullanıcı i , B r-1'in üçüncü bileşeninden Q r-1'i hesaplar ve i ∈ SV r, 2 veya
değil.
• Eğer i / ∈ SV r, 2 ise, i Adım 2'nin kendi yürütmesini hemen durdurur.
• Eğer i ∈ SV r, 2 ise , bir süre bekledikten sonra t 2
λ + Λ, i aşağıdaki gibi davranır.
1. Kullanıcıyı ℓ bulur, öyle ki H (σ
r, 1
ℓ
) ≤ H (σ
r, 1
j) tüm kimlik bilgileri için σ
r, 1
j
bu parçası
Şimdiye kadar aldığı başarıyla doğrulanmış (r, 1) -mesajlar.a
2. Eğer from'den geçerli bir mesaj almışsa m
r, 1
ℓ
= (B r
ℓ , esig ℓ (H (B r
ℓ)), σ
r, 1
ℓ
),b sonra ayarlar
v /

ben
H (B r
ℓ); aksi halde v /
ben
⊥.
3. i mesajını hesaplar m
r, 2
ben
(ESIG i (v /
i), σ
r, 2
i),c geçici gizli anahtarını yok eder
sk
r, 2
i ve sonra m'yi yayar
r, 2
i .
a Esasen, i kullanıcısı, r turunun liderinin kullanıcı ℓ olduğuna özel olarak karar verir.
b Yine, oyuncunun imzaları ve karmalarının tümü başarıyla doğrulandı ve P AY r
ℓ
B r'de
ℓ geçerli bir maaş setidir
round r - i, P AY r olup olmadığını kontrol etmese de
ℓ
ℓ için maksimaldir veya değildir.
c Mesaj m r, 2
ben
düşündüğüm oyuncunun v ′
i sonraki bloğun karması olmak veya sonraki bloğu ele almak
bloğun boş olması.
17 Yani, tüm imzalar doğrudur ve hem blok hem de hash geçerlidir - kontrol etmesem de
dahil edilen maaş setinin teklif veren için maksimum olup olmadığı.

Sayfa 36

Adım 3: GC'nin İkinci Adımı
Her kullanıcı için talimatlar i ∈ PK r-k : i Kullanıcısı, kendi r turunun 3. adımını başlatır.
B r-1'i bilir .
• Kullanıcı i , B r-1'in üçüncü bileşeninden Q r-1'i hesaplar ve i ∈ SV r, 3 veya
değil.
• Eğer i / ∈ SV r, 3 ise , i Adım 3'ün kendi yürütmesini hemen durdurur.
• Eğer i ∈ SV r, 3 ise , bir süre bekledikten sonra t 3
t 2 + 2λ = 3λ + Λ, i aşağıdaki gibi davranır.
1. Eğer bir v / = value değeri varsa , öyle ki, tüm geçerli mesajlar arasında m
r, 2
j
o aldı
bunların 2 / 3'ünden fazlası formdadır (ESIG j (v /), σ
r, 2
j) herhangi bir çelişki olmaksızın,a
sonra mesajı hesaplar m
r, 3
ben
(ESIG i (h /), σ

r, 3
i). Aksi takdirde m hesaplar
r, 3
ben
(ESIG i (⊥), σ
r, 3
i).
2. Geçici gizli anahtarını yok ederim
r, 3
i ve sonra m'yi yayar
r, 3
i .
a Yani, sırasıyla ESIG j (v ′) ve farklı bir ESIG j (v ′ ′) içeren iki geçerli mesaj almamışsa ,
bir oyuncudan j. Dürüst bir oyuncu
belirli bir biçimde mesajlar istiyorsa, birbiriyle çelişen mesajlar asla sayılmaz veya geçerli sayılmaz.
36

Sayfa 37

Adım 4: GC'nin Çıktısı ve BBA'nın İlk Adımı ⋆
Her kullanıcı için talimatlar i ∈ PK r-k : Kullanıcı i, kendi r turunun 4. Adımını başlatır.
B r-1'i bilir .
• Kullanıcı i , B r-1'in üçüncü bileşeninden Q r-1'i hesaplar ve i ∈ SV r, 4 veya
değil.
• Eğer i / ∈ SV r, 4 ise , o zaman 4. Adımın kendi uygulamasını hemen durdurur.
• Eğer i ∈ SV r, 4 ise , bir süre bekledikten sonra t 4
t 3 + 2λ = 5λ + Λ, i aşağıdaki gibi davranır.
1. GC'nin çıktısı olan v i ve g i'yi aşağıdaki gibi hesaplar .
(a) Tüm geçerli mesajlar arasında m olacak şekilde bir v / = ⊥ değeri varsa
r, 3
j
o sahip
alınan, 2 / 3'ünden fazlası formdadır (ESIG j (v /), σ
r, 3
j), sonra ayarlar
v ben
v / ve g i
2.
(b) Aksi takdirde, tüm geçerli mesajlar arasında bir v / = value değeri varsa
m
r, 3
j
almışsa, bunların 1 / 3'ünden fazlası formdadır (ESIG j (v /), σ
r, 3
j), sonra
o v setleri i
v / ve g i
1.a
(c) Aksi takdirde, v i'yi belirler
H (B r
ǫ) ve g i
0.
2. BBA ⋆'nın girdisi olan b i'yi şu şekilde hesaplar :
b ben
0 eğer g ben = 2 ve b ben

Aksi takdirde 1.
3. m mesajını hesaplar
r, 4
ben
(ESIG i (b i), ESIG i (v i), σ
r, 4
i), geçici olanını yok eder
gizli anahtar sk
r, 4
i ve sonra m'yi yayar
r, 4
i .
a (b) durumunda v ′ ' nin, varsa, benzersiz olması gerektiği kanıtlanabilir .
37

Sayfa 38

Adım s, 5 ≤ s ≤ m + 2, s - 2 ≡ 0 mod 3: BBA'nın 0'a Sabit Bir Madeni Para Adımı ⋆
Her kullanıcı için talimatlar i ∈ PK r-k : i Kullanıcısı, kendi r aşamasını başlatır.
B r-1'i bilir .
• Kullanıcı i , B r-1'in üçüncü bileşeninden Q r-1'i hesaplar ve i ∈ SV r, s olup olmadığını kontrol eder .
• Eğer i / ∈ SV r, s , o zaman i Adım s'nin kendi yürütmesini hemen durdurur.
• Eğer i ∈ SV r ise, o zaman aşağıdaki gibi davranır.
- O, t zamanında bir miktarda kadar bekler s
t s-1 + 2λ = (2s - 3) λ + Λ geçti.
- Bitiş Koşulu 0: Böyle bir bekleme sırasında ve herhangi bir zamanda, bir
string v = ⊥ ve bir adım s / öyle ki
(a) 5 ≤ s / ≤ s, s / - 2 ≡ 0 mod 3 — yani, Adım s / 0'a Sabit Madeni Para adımıdır,
(b) en az t H = 2n aldım
3
+ 1 geçerli mesaj m
r, s ′ -1
j
= (ESIG j (0),
ESIG j (v), σ
r, s ′ -1
j
),a ve
(c) geçerli bir mesaj aldım m
r, 1
j
= (B r
j , esig j (H (B r
j)), σ
r, 1
j) v = H (B r
j),
daha sonra, s adımının (ve aslında r turunun) kendi yürütmesini,
herhangi bir şeyi yaymak; ayarlar B r = B r
j ; ve kendi CERT setleri r mesajlarının kümesi olmak
m
r, s ′ -1
j
(b) alt adımı.b

- Bitiş Koşulu 1: Böyle bir bekleme sırasında ve herhangi bir zamanda, bir
adım s / öyle ki
(a ') 6 ≤ s / ≤ s, s / - 2 ≡ 1 mod 3 — yani, Adım s / 1'e Sabit Madeni Para adımıdır ve
(b ') en az t H geçerli mesaj aldım m
r, s ′ -1
j
= (ESIG j (1), ESIG j (v j),
σ
r, s ′ -1
j
),c
daha sonra, s adımının (ve aslında r turunun) kendi yürütmesini,
herhangi bir şeyi yaymak; ayarlar B r = B r
ϱ ; ve kendi CERT setleri r mesajlarının kümesi olmak
m
r, s ′ -1
j
alt adımın (b ').
- Aksi takdirde, bekleme sonunda kullanıcı i aşağıdakileri yapar.
Tüm geçerli bileşenlerin ikinci bileşenlerinde v i'yi v j'lerin çoğunluk oyu olarak belirler.
m
r, s-1
j
aldı.
B i'yi şu şekilde hesaplar .
Tüm geçerli m'nin 2 / 3'ünden fazlası
r, s-1
j
aldığı formdadır
(ESIG j (0), ESIG j (v j), σ
r, s-1
j
), sonra b i ayarlar
0.
Aksi takdirde, tüm geçerli m'nin 2 / 3'ünden fazlası
r, s-1
j
aldığı formdadır
(ESIG j (1), ESIG j (v j), σ
r, s-1
j
), sonra b i ayarlar
1.
Aksi takdirde, b i ayarlar
0.
M mesajını hesaplar
r, s
ben
(ESIG i (b i), ESIG i (v i), σ
r, s
i), geçici olanını yok eder
gizli anahtar sk
r, s
i ve sonra m'yi yayar
r, s

i .
a Oyuncu i'den 1'e imza atan j'den bir mesaj almış olsa bile, j oyuncusunun böyle bir mesajı sayılır.
Bitiş Koşulu 1 için de benzer şeyler. Analizde gösterildiği gibi, bu, tüm dürüst kullanıcıların bilmesini sağlamak için yapılır.
B r zaman içinde birbirlerinden λ.
b Kullanıcı i artık B r'yi ve kendi raundunun bitirdiğini biliyor . Hala genel bir kullanıcı olarak mesajların yayılmasına yardımcı oluyor, ancak
herhangi bir yayılımı a (r, s) -verifier olarak başlatmaz. Özellikle, tüm mesajların yayılmasına yardımcı oldu.
Protokolümüz için yeterli olan CERT r . Ayrıca b i ayarlaması gerektiğini unutmayın
İkili BA protokolü için 0, ancak b i
zaten bu durumda gerekli değildir. Gelecekteki tüm talimatlar için benzer şeyler.
c Bu durumda, v j'lerin ne olduğu önemli değildir .

Sayfa 39

Aşama s, $6 \leq s \leq m + 2$, $s - 2 \equiv 1 \bmod 3$: BBA A Düğme Sabit için-1 Aşama ⋆
Her kullanıcı için talimatlar i ∈ PK r-k : i Kullanıcısı, kendi r aşamasını başlatır.
B r-1'i bilir .
• Kullanıcı i , B r-1'in üçüncü bileşeninden Q r-1'i hesaplar ve i ∈ SV r, s veya
değil.
• Eğer i / ∈ SV r, s , o zaman i Adım s'nin kendi yürütmesini hemen durdurur.
• Eğer i ∈ SV r ise, o zaman aşağıdakileri yapar.
- O, t zamanında bir miktarda kadar bekler s
(2s - 3) λ + Λ geçti.
- Bitiş Koşulu 0: Coin-Fixed-To-0 adımlarıyla aynı talimatlar.
- Bitiş Koşulu 1: Coin-Fixed-To-0 adımlarıyla aynı talimatlar.
- Aksi takdirde, bekleme sonunda kullanıcı i aşağıdakileri yapar.
Tüm geçerli bileşenlerin ikinci bileşenlerinde v i'yi v j'lerin çoğunluk oyu olarak belirler.
m
r, s-1
j
aldı.
B i'yi şu şekilde hesaplar .
Tüm geçerli m'nin 2 / 3'ünden fazlası
r, s-1
j
aldığı formdadır
(ESIG j (0), ESIG j (v j), σ
r, s-1
j
), sonra b i ayarlar
0.
Aksi takdirde, tüm geçerli m'nin 2 / 3'ünden fazlası
r, s-1
j
aldığı formdadır
(ESIG j (1), ESIG j (v j), σ
r, s-1
j
), sonra b i ayarlar
1.
Aksi takdirde, b i ayarlar
1.
M mesajını hesaplar

r, s
ben
(ESIG i (b i), ESIG i (v i), σ
r, s
i), geçici olanını yok eder
gizli anahtar sk
r, s
i ve sonra m'yi yayar
r, s
i .
39

Sayfa 40

Adım s, 7 ≤ s ≤ m + 2, s - 2 ≡ 2 mod 3: BBA'nın Gerçekten Ters Çevrilmiş Bir Madeni Para Adımı ⋆
Her kullanıcı için talimatlar i ∈ PK r-k : i Kullanıcısı, kendi r aşamasını başlatır.
B r-1'i bilir .
• Kullanıcı i , B r-1'in üçüncü bileşeninden Q r-1'i hesaplar ve i ∈ SV r, s veya
değil.
• Eğer i / ∈ SV r, s , o zaman i Adım s'nin kendi yürütmesini hemen durdurur.
• Eğer i ∈ SV r ise, o zaman aşağıdakileri yapar.
- O, t zamanında bir miktarda kadar bekler s
(2s - 3) λ + Λ geçti.
- Bitiş Koşulu 0: Coin-Fixed-To-0 adımlarıyla aynı talimatlar.
- Bitiş Koşulu 1: Coin-Fixed-To-0 adımlarıyla aynı talimatlar.
- Aksi takdirde, bekleme sonunda kullanıcı i aşağıdakileri yapar.
Tüm geçerli bileşenlerin ikinci bileşenlerinde v i'yi v j'lerin çoğunluk oyu olarak belirler.
m
r, s-1
j
aldı.
B i'yi şu şekilde hesaplar .
Tüm geçerli m'nin 2 / 3'ünden fazlası
r, s-1
j
aldığı formdadır
(ESIG j (0), ESIG j (v j), σ
r, s-1
j
), sonra b i ayarlar
0.
Aksi takdirde, tüm geçerli m'nin 2 / 3'ünden fazlası
r, s-1
j
aldığı formdadır
(ESIG j (1), ESIG j (v j), σ
r, s-1
j
), sonra b i ayarlar
1.
Aksi takdirde, SV
r, s-1
ben
geçerli bir aldığı (r, s - 1) -verifiers kümesi
mesaj m
r, s-1

j
. O b setleri i
lsb (min
j∈ SV r, s − 1
ben
H (σ
r, s-1
j
)).
M mesajını hesaplar
r, s
ben
(ESIG i (b i), ESIG i (v i), σ
r, s
i), geçici olanını yok eder
gizli anahtar sk
r, s
i ve sonra m'yi yayar
r, s
i .
40

Sayfa 41

Adım m + 3: BBA'nın Son Adımı ⋆ a
Her kullanıcı için talimatlar i ∈ PK r-k : Kullanıcı i, kendi m + 3 turunun kendi Adım m + 3'ünü başlatır.
B r-1'i bilir .
• Kullanıcı i , B r-1'in üçüncü bileşeninden Q r-1'i hesaplar ve i ∈ SV r, m + 3 veya değil.
• Eğer i / ∈ SV r, m + 3 ise , o zaman i , m + 3 Adımının kendi yürütmesini hemen durdurur.
• Eğer i ∈ SV r, m + 3 ise o zaman aşağıdakileri yapar.
- Bir miktar t m + 3 olana kadar bekler.
t m + 2 + 2λ = (2m + 3) λ + Λ geçti.
- Bitiş Koşulu 0: Coin-Fixed-To-0 adımlarıyla aynı talimatlar.
- Bitiş Koşulu 1: Coin-Fixed-To-0 adımlarıyla aynı talimatlar.
- Aksi takdirde, bekleme sonunda kullanıcı i aşağıdakileri yapar.
Ben yola çıkıyor
1 ve B r
B r
ϱ .
M mesajını hesaplar
r, m + 3
ben
= (ESIG i (dışarı i), ESIG i (H (B r)), σ
r, m + 3
ben
), yok eder
geçici gizli anahtar sk
r, m + 3
ben
ve sonra m'yi yayar
r, m + 3
ben
B r onaylamak için .b
Bir olasılık BBA ezici ile ⋆ bu adımdan önce sona erdi ve biz tamlığı için bu adımı belirtin.

b Adım m + 3'teki bir sertifikanın ESIG i (out i) içermesi gerekmez . Bunu yalnızca tekdüzelik için dahil ediyoruz:
sertifikalar artık hangi adımda oluşturulurlarsa oluşturulsun tek tip bir biçime sahipler.

Sayfa 42

Round-r Bloğunun Doğrulayıcı Olmayanlar Tarafından Yeniden İnşası
Sistemdeki her i kullanıcısı için talimatlar: Kullanıcı i öğrenir öğrenmez kendi turunu başlatır
B r-1 , aşağıdaki gibi blok bilgilerini bekler.
- Böyle bir bekleme sırasında ve herhangi bir zamanda, bir v dizisi ve bir adım s / böyle varsa
o
(a) s / - 2 ≡ 0 mod 3 ile 5 ≤ s / ≤ m + 3,
(b) en az t H geçerli mesaj aldım m
r, s ′ -1
j
= (ESIG j (0), ESIG j (v), σ
r, s ′ -1
j
), ve
(c) geçerli bir mesaj aldım m
r, 1
j
= (B r
j , esig j (H (B r
j)), σ
r, 1
j) v = H (B r
j),
sonra, r turunu kendi yürütmesini hemen durdururum; ayarlar B r = B r
j ; ve kendi CERT setleri r
mesaj seti olmak m
r, s ′ -1
j
(b) alt adımı.
- Eğer, bu bekleme sırasında ve zaman herhangi bir noktada, bir basamak s vardır / şekildedir
(a ') s / - 2 ≡ 1 mod 3 ile 6 ≤ s / ≤ m + 3 ve
(b ') en az t H geçerli mesaj aldım m
r, s ′ -1
j
= (ESIG j (1), ESIG j (v j), σ
r, s ′ -1
j
),
sonra, r turunu kendi yürütmesini hemen durdururum; ayarlar B r = B r
ϱ ; ve kendi CERT setleri r
mesaj seti olmak m
r, s ′ -1
j
alt adımın (b ').
- Böyle bir bekleme sırasında ve herhangi bir zamanda, en az t H geçerli mesaj aldıysam
m
r, m + 3
j
= (ESIG j (1), ESIG j (H (B r
ϱ)), σ

r, m + 3
j
), sonra kendi r turunu yürütmesini durdururum
hemen, B r = B r
, ∈ ve kendi CERT setleri r mesajları m kümesi olmak
r, m + 3
j
1 için
ve H (B r
ϱ).
5.5 Algorand Analizi ′
1
Analizde kullanılan her r ≥ 0 turu için aşağıdaki gösterimleri sunuyoruz.
• İlk dürüst kullanıcının B r-1'i bildiği zaman T r olsun .
• I r + 1 [T r + 1 , T r + 1 + λ] aralığı olsun .
Protokolün başlatılmasıyla T 0 = 0 olduğuna dikkat edin. Her s ≥ 1 ve i ∈ SV r, s için şunu hatırlayın
α
r, s
ben
ve β
r, s
ben
sırasıyla oyuncu i adımlarının başlangıç ve bitiş zamanlarıdır. Dahası,
her 2 ≤ s ≤ m + 3 için t s = (2s - 3) λ + Λ olduğunu hatırlayın . Ayrıca I 0
{0} ve t 1
0.
Son olarak, L r ≤ m / 3'ün Bernoulli denemelerinin sayısını temsil eden rastgele bir değişken olduğunu hatırlayın.
her deneme 1 olasılıkla 1 olduğunda 1'i görmek gerekiyor
p h
2
ve en fazla m / 3 deneme vardır. Düştüm
denemeler başarısız olur sonra L r
m / 3.
Analizde, ihtiyaç duyulan zamana göre aslında ihmal edilebilir olduğu için hesaplama süresini göz ardı ediyoruz.
mesajları yaymak için. Her durumda, biraz daha büyük λ ve Λ kullanarak, hesaplama süresi
doğrudan analize dahil edilebilir. Aşağıdaki ifadelerin çoğu "ezici bir şekilde"
olasılık ”ve bu gerçeği analizde tekrar tekrar vurgulamayabiliriz.

Sayfa 43

5.6 Ana Teorem
Teorem 5.1. Aşağıdaki özellikler, her raund r round 0 için çok büyük bir olasılıkla geçerlidir:
1. Tüm dürüst kullanıcılar aynı blok B r üzerinde hemfikirdir .
2. Lider ℓ r dürüst olduğunda, B r bloğu ℓ r tarafından üretilir , B r maksimum kazanç seti içerir
ℓ r tarafından zamana göre alındı α
r, 1
ℓ r , T r + 1 ≤ T r + 8λ + Λ ve tüm dürüst kullanıcılar zaman içinde B r'yi bilir
aralık I r + 1 .
3. Lider ℓ r kötü niyetli olduğunda, T r + 1 ≤ T r + (6L r + 10) λ + Λ ve tüm dürüst kullanıcılar
B r bilir
zaman aralığında r + 1 .
4. L r için p h = h 2 (1 + h - h 2) ve lider ℓ r en azından p h olasılıkla dürüsttür .
Ana teoremimizi ispatlamadan önce iki noktaya değinelim.

Uyarılar.
• Blok Oluşturma ve Gerçek Gecikme. B r bloğunu oluşturma zamanı T r + 1 −T r olarak tanımlanmıştır .
Yani, dürüst bir kullanıcının B r'yi ilk kez öğrenmesi ile
Dürüst bir kullanıcı ilk kez B r-1'i öğrenir . Raund lideri dürüst olduğunda, Mülk 2 bizim
ana teorem , ne olursa olsun, B r'yi üretmek için kesin zamanın 8λ + Λ zamanı olduğunu garanti eder.
h> 2/3 kesin değeri olabilir. Lider kötü niyetli olduğunda, Özellik 3,
B r'yi oluşturmak için beklenen süre (12
p h
+ 10) λ + Λ, yine kesinlik ne olursa olsun
h değeri.18 Bununla birlikte, B r'yi üretmek için beklenen süre, h'nin kesin değerine bağlıdır.
Nitekim, Özellik 4'e göre, p h = h 2 (1 + h - h 2) ve lider en azından olasılıkla dürüsttür
p h , bu nedenle
E [T
r + 1 - T r] ≤ h 2 (1 + h - h 2) · (8λ + Λ) + (1 - h 2 (1 + h - h 2)) ((
12
h 2 (1 + h - h 2)
+ 10) λ + Λ).
Örneğin, h =% 80 ise, E [T r + 1 - T r] ≤ 12,7λ + Λ.
• λ ve Λ. Algorand / adımında doğrulayıcılar tarafından gönderilen mesajların
boyutunun baskın olduğuna dikkat edin
dijital imza anahtarlarının uzunluğuna göre, sayısı sabit kalsa bile
kullanıcılar çok büyük. Ayrıca, herhangi bir s> 1 adımında, aynı beklenen doğrulayıcı sayısı n olduğuna dikkat edin.
kullanıcı sayısı ister 100K, ister 100M veya 100M olsun kullanılabilir. Bu böyledir çünkü n yalnızca h ve F'ye bağlıdır. Özetle, bu nedenle, ani bir gizli anahtar uzunluğunu artırma ihtiyacını ortadan kaldırarak,
λ'nın değeri, içindeki kullanıcı sayısı ne kadar büyük olursa olsun aynı kalmalıdır.
Öngörülebilir gelecek.
Buna karşılık, herhangi bir işlem oranı için işlem sayısı,
kullanıcılar. Bu nedenle, tüm yeni işlemleri zamanında işlemek için bir bloğun boyutu
kullanıcı sayısı arttıkça büyür ve Λ'nin de büyümesine neden olur. Bu nedenle, uzun vadede sahip olmalıyız
λ << Λ. Buna göre, daha büyük bir λ katsayısına ve aslında bir katsayıya sahip olmak uygundur.
1 için Λ.
Teoremin Kanıtı 5.1. Özellikler 1–3'ü tümevarımla ispatlıyoruz: r - 1 turu için geçerli olduklarını varsayarak
(genelliği kaybetmeden, r = 0 olduğunda otomatik olarak "-1 turu" için geçerli olurlar),
yuvarlak r.
18 Gerçekten, E [T r + 1 - T r] ≤ (6E [L r] + 10) λ + Λ = (6 · 2
p h + 10) λ + Λ = (12
p h + 10) λ + Λ.

Sayfa 44

B r-1 endüktif hipotez tarafından benzersiz bir şekilde tanımlandığından, SV r, s kümesi benzersiz bir şekilde tanımlanır
r turunun her adımı için. N 1 , SV r, 1 = ∅ seçimine göre çok büyük olasılıkla. Biz şimdi
Bölüm 5. 7 ve 5.8'de ispatlanan aşağıdaki iki önermeyi belirtiniz . İndüksiyon boyunca ve
iki lemmanın ispatları, 0 turunun analizi neredeyse tümevarım adımıyla aynıdır,
ve ortaya çıktıklarında farklılıkları vurgulayacağız.
Lemma 5.2. [Tamlık Lemması] 1–3 Özelliklerinin, liderin
ℓ r ezici bir olasılıkla dürüst,
• Tüm dürüst kullanıcılar , ℓ r tarafından üretilen ve bir maksimal değeri içeren aynı blok B r üzerinde hemfikirdir.

ℓ r tarafından zamana göre alınan ödeme seti α
r, 1
ℓ r
∈ I r ; ve
• T r + 1 ≤ T r + 8λ + Λ ve tüm dürüst kullanıcılar B r'yi I r + 1 zaman aralığında bilirler .
Lemma 5.3. [Sağlamlık Lemması] Liderin
ℓ r kötü niyetli, büyük olasılıkla, tüm dürüst kullanıcılar aynı blok üzerinde hemfikir B r , T r + 1 ≤ T r + (6L r + 10) λ + Λ ve tüm dürüst kullanıcılar B r'yi I r + 1 zaman aralığında bilir .
Özellikler 1-3, Lemmas 5. 2 ve 5.3'ü r = 0'a ve endüktif adıma uygulayarak geçerli olur . En sonunda, Özellik 4'ü Bölüm 5.9'da kanıtlanan aşağıdaki lemma olarak yeniden ifade ediyoruz .
Lemma 5.4. Verilen Özellikler r'den önceki her tur için 1–3, L r için p h = h 2 (1 + h - h 2) ve lider ℓ r olasılıkla dürüsttür, en azından p h .
Yukarıdaki üç lemmayı bir araya getiren Teorem 5 .1 geçerli olur.
■
Aşağıdaki lemma, endüktif
hipotez ve yukarıdaki üç lemmanın ispatlarında kullanılacaktır.
Lemma 5.5. Özellikler 1-3 r - 1 için geçerli olduğunu varsayın. R turunun her s ≥ 1 adımı için ve her dürüst doğrulayıcı i ∈ HSV r, s , bizde
(a) α
r, s
ben
∈ I r ;
(b) eğer oyuncu i bir süre t s beklediyse , o zaman β
r, s
ben
∈ [T r + t s , T r + λ + t s] r> 0 için ve
β
r, s
ben
= r = 0 için t s ; ve
(c) eğer oyuncu i bir süre t s beklediyse , o zaman zamana göre β
r, s
i , o tüm mesajları aldı
tüm dürüst doğrulayıcılar tarafından gönderilir j ∈ HSV r, s ′ tüm s / <s adımları için .
Dahası, s ≥ 3'ün her adımı için, buna sahibiz
(d) iki farklı oyuncu i, i / ∈ SV r, s ve aynı olan iki farklı v, v / değeri yoktur
Her iki oyuncunun süre t bir miktarda bekledi şekilde uzunluğu, s fazla bütün 2/3 daha geçerli mesajlar m
r, s-1
j
Aldığım oyuncu v için imza attı ve geçerli tüm oyuncuların 2 / 3'ünden fazlası
mesajlar m
r, s-1
j
oyuncu i / aldığı v / için imzaladı .
Kanıt. I B bildiği oyuncu olarak özelliği (a) endüktif hipotezi tarafından doğrudan aşağıdaki r-1 içinde zaman aralığı I r ve kendi adımını hemen başlatır. Özellik (b), doğrudan (a) 'dan gelir: çünkü
i oyuncusu süre t bir miktarda bekledi s hareket etmeden önce, β
r, s
ben
= α
r, s
ben
+ t s . A olduğunu unutmayın
r, s

ben
= 0 için
r = 0.
Şimdi Özelliği (c) kanıtlıyoruz. Eğer s = 2 ise, o zaman Özellik (b) ile, tüm doğrulayıcılar için j ∈ HSV r, 1'e sahibiz
β
r, s
ben
= α
r, s
ben
+ t s ≥ T r + t s = T r + λ + Λ ≥ β
r, 1
j
+ Λ.

Sayfa 45

Her doğrulayıcı j ∈ HSV r, 1 mesajını zamanında gönderir β
r, 1
j
ve mesaj tüm dürüstçe ulaşır
kullanıcı sayısı en fazla zaman, zamana göre β
r, s
ben
oyuncu, içindeki tüm doğrulayıcılar tarafından gönderilen mesajları aldım
HSV r, 1 istenildiği gibi.
Eğer s> 2 ise t s = t s-1 + 2λ. Özelliğe göre (b), tüm s / <s adımları ve tüm doğrulayıcılar için j ∈ HSV r, s ′ ,
β
r, s
ben
= α
r, s
ben
+ t s ≥ T r + t s = T r + t s-1 + 2λ ≥ T r + t s ′ + 2λ = T r + λ + t s ′ + λ ≥ β
r, s ′
j
+ λ.
Her doğrulayıcı j ∈ HSV r, s ′ mesajını zamanında gönderir β
r, s ′
j
ve mesaj tüm dürüstçe ulaşır
kullanıcı sayısı en fazla λ zaman diliminde users
r, s
ben
oyuncu, tüm dürüst doğrulayıcılar tarafından gönderilen tüm mesajları aldım
HSV'de r, tüm s / <s için s ′ . Böylece Mülkiyet (c) geçerlidir.
Son olarak, Mülkiyeti (d) kanıtlıyoruz. Doğrulayıcılar j ∈ SV r, s-1'in en fazla iki şeyi imzaladığına dikkat edin.
Adım s - 1'in geçici gizli anahtarlarını kullanarak: çıktısıyla aynı uzunlukta bir v j değeri
hash fonksiyonu ve ayrıca a bit b j ∈ {0,1} eğer s - 1 ≥ 4. Bu yüzden lemmanın ifadesinde
v ve v / değerlerinin aynı uzunluğa sahip olmasını istiyoruz: birçok doğrulayıcı hem bir hash değerini imzalamış olabilir
v ve biraz b, böylece her ikisi de 2/3 eşiğini geçer.

Çelişki adına, istenen doğrulayıcılar i, i / ve değerleri v, v / var olduğunu varsayın .
MSV r, s-1'deki bazı kötü niyetli doğrulayıcıların hem v hem de v / imzalamış olabileceğini , ancak her birinin dürüst olduğunu unutmayın.
HSV r'deki doğrulayıcı , s-1 bunlardan en fazla birini imzaladı. Mülkiyet (c) ile, hem i hem de i / aldım
HSV r, s-1'deki tüm dürüst doğrulayıcılar tarafından gönderilen tüm mesajlar .
HSV r, s-1 (v), v, MSV'yi imzalayan dürüst (r, s - 1) -verifiers kümesi olsun.
r, s-1
ben
set
kötü niyetli (r, s - 1) -geçerli bir ileti aldığım doğrulayıcılar ve MSV
r, s-1
ben
(v)
MSV'nin alt kümesi
r, s-1
ben
geçerli bir mesaj imzaladığım kişiden v.
ben ve v, sahibiz
oran
| HSV r, s-1 (v) | + | MSV
r, s-1
ben
(v) |
| HSV r, s-1 | + | MSV
r, s-1
ben
|
>
2
3
.
(1)
İlk gösteriyoruz
| MSV
r, s-1
ben
(v) | ≤ | HSV r, s-1 (v) |.
(2)
Aksi varsayılırsa, parametreler arasındaki ilişkilere göre, ezici olasılıkla
| HSV r, s-1 | > 2 | MSV r, s-1 | ≥ 2 | MSV
r, s-1
ben
| böylelikle
oran <
| HSV r, s-1 (v) | + | MSV
r, s-1
ben
(v) |
3 | MSV
r, s-1
ben
|
<
2 | MSV

r, s-1
ben
(v) |
3 | MSV
r, s-1
ben
|
≤
2
3
,
Eşitsizlikle çelişen 1.
Ardından, Eşitsizlik 1'e göre elimizde
2 | HSV r, s-1 | + 2 | MSV
r, s-1
ben
| <3 | HSV r, s-1 (v) | + 3 | MSV
r, s-1
ben
(v) |
≤ 3 | HSV r, s-1 (v) | + 2 | MSV
r, s-1
ben
| + | MSV
r, s-1
ben
(v) |.
Eşitsizlikle Birleşen 2,
2 | HSV r, s-1 | <3 | HSV r, s-1 (v) | + | MSV
r, s-1
ben
(v) | ≤ 4 | HSV r, s-1 (v) |,
Hangi ima
| HSV r, s-1 (v) | >
1
2
| HSV r, s-1 |.
45

Sayfa 46

Benzer şekilde, i gereksinimleri tarafından / ve v / Elimizdeki
| HSV r, s-1 (v /) | >
1
2
| HSV r, s-1 |.
Dürüst bir doğrulayıcı j ∈ HSV r olduğundan, s-1 geçici gizli anahtarını yok eder.
r, s-1
j
yayılmadan önce
Rakip, mesajından sonra j'nin imzalamadığı bir değer için j'nin imzasını taklit edemez.
j'nin bir doğrulayıcı olduğunu öğrenmek. Dolayısıyla, yukarıdaki iki eşitsizlik | HSV r, s-1 | ≥ | HSV r, s-1 (v) | +
| HSV r, s-1 (v /) | > | HSV r, s-1 |, bir çelişki. Buna göre, istenen i, i / , v, v / mevcut değil ve
Mülkiyet (d) tutmaktadır.
■

5.7 Tamlık Lemması
Lemma 5. 2. [Tamlık Lemması, yeniden ifade edilmiştir] 1–3 Özelliklerini varsayarsak, r − 1 turu için lider ℓ r ezici bir olasılıkla dürüsttür,
• Tüm dürüst kullanıcılar , ℓ r tarafından üretilen ve bir maksimal değeri içeren aynı blok B r üzerinde hemfikirdir.
ℓ r tarafından zamana göre alınan ödeme seti α
r, 1
ℓ r
∈ I r ; ve
• T r + 1 ≤ T r + 8λ + Λ ve tüm dürüst kullanıcılar B r'yi I r + 1 zaman aralığında bilirler .
Kanıt. Endüktif hipotez ve Lemma 5 .5'e göre, her adım s ve doğrulayıcı i ∈ HSV r, s için ,
α
r, s
ben
∈ r . Aşağıda protokolü adım adım analiz ediyoruz.
Adım 1. Tanım gereği, her dürüst doğrulayıcı i ∈ HSV r, 1 istenen mesajı yayar m
r, 1
ben
-de
zaman β
r, 1
ben
= α
r, 1
ben nerede m
r, 1
ben
= (B r
ben , esig ben (H (B r
i)), σ
r, 1
i), B r
i = (r, P AY r
i , SIG ben (Q r-1), H (B r-1)),
ve P AY r
i zamana göre gördüğüm tüm ödemeler arasında en yüksek maaş setidir α
r, 1
i .
Adım 2. Dürüst bir doğrulayıcı i ∈ HSV r, 2'yi keyfi olarak düzeltin . Lemma 5. 5, i oyuncunun işi bittiğinde
zamanında bekliyor β
r, 2
ben
= α
r, 2
ben
+ T 2 , O HSV içinde doğrulayıcı tarafından gönderilen mesajları aldığı r, 1 de dahil olmak üzere
m
r, 1
ℓ r . ℓ r'nin tanımına göre , PK r-k'de kimlik bilgilerinin hash'i olan başka bir oyuncu yoktur.
değer H'den küçüktür (σ
r, 1
ℓ r). Tabii ki, Adversary bozabilir ℓ r gördükten sonra o H (σ
r, 1
ℓ r)

o zaman oyuncu ℓ çok küçük, ancak r onun fani anahtarı ve mesaj m yok etmiştir
r, 1
ℓ r
yayıldı. Böylece doğrulayıcı i oyuncu ℓ olmak kendi liderini belirler r . Buna göre, zamanında β
r, 2
ben ,
verifier m'yi yayar
r, 2
ben
= (ESIG i (v /
i), σ
r, 2
i), nerede v /
i = H (B r
ℓ r). R = 0 olduğunda, tek fark
bu β
r, 2
ben
= t 2 , bir aralıkta olmak yerine. Gelecekteki adımlar için benzer şeyler söylenebilir ve biz
onları tekrar vurgulamayacak.
Adım 3. Dürüst bir doğrulayıcıyı rastgele düzeltin i , HSV r, 3 . Lemma 5. 5, i oyuncunun işi bittiğinde
zamanında bekliyor β
r, 3
ben
= α
r, 3
ben
+ t 3 , HSV r, 2'deki doğrulayıcılar tarafından gönderilen tüm mesajları aldı .
Parametreler arasındaki ilişkilere göre, ezici olasılıkla | HSV r, 2 | >
2 | MSV r, 2 |. Dahası, hiçbir dürüst doğrulayıcı çelişen mesajlar imzalamayacaktır.
dürüst bir doğrulayıcının imzasını, ikincisi karşılık gelen kişiyi yok ettikten sonra taklit edemez.
geçici gizli anahtar. Böylece aldığım tüm geçerli (r, 2) -mesajların 2 / 3'ünden fazlası
dürüst doğrulayıcılar ve form m
r, 2
j
= (ESIG j (H (B r
ℓ r)), σ
r, 2
j) hiçbir çelişki olmadan.
Buna göre, zamanında β
r, 3
ben
Oyuncu m'yi yayar
r, 3
ben
= (ESIG i (v /), σ
r, 3
i), v / = H (B r
ℓ r).

Sayfa 47

Adım 4. Dürüst bir doğrulayıcıyı rastgele düzeltin i ∈ HSV r, 4 . Lemma 5. 5, oyuncu i hepsini aldı
HSV r'de doğrulayıcılar tarafından gönderilen mesajlar , 3 bekleme zamanı bittiğinde β
r, 4

ben
= α
r, 4
ben
+ t 4 . Benzer
3. Adım, aldığım tüm geçerli (r, 3) mesajların 2 / 3'ünden fazlası dürüst doğrulayıcılardan ve formun m
r, 3
j
= (ESIG j (H (B r
ℓ r)), σ
r, 3
j).
Buna göre, oyuncu i, v i = H (B r
ℓ r), g ben = 2 ve b i = 0. β zamanında
r, 4
ben
= α
r, 4
ben
+ t 4 yayılır
m
r, 4
ben
= (ESIG i (0), ESIG i (H (B r
ℓ r)), σ
r, 4
i).
Adım 5. Dürüst bir doğrulayıcıyı rastgele düzeltin i ∈ HSV r, 5 . Lemma 5. 5, sahip olacağım oyuncu HSV r'de doğrulayıcılar tarafından gönderilen tüm mesajları aldı , 4 zamana kadar beklediyse α
r, 5
ben
+ t 5 . Bunu not et
| HSV r, 4 | ≥ t , H .19 Ayrıca HSV r, 4'teki tüm doğrulayıcılarınH (B r
ℓ r).
As | MSV r, 4 | <t H , herhangi bir v / = H (B r
ℓ r) t H tarafından imzalanmış olabilir
SV r, 4'teki doğrulayıcılar (kimin kötü niyetli olması gerekirdi), bu yüzden oyuncu i, ondan önce durmaz.
alınan t H geçerli mesajlar m
r, 4
j
= (ESIG j (0), ESIG j (H (B r
ℓ r)), σ
r, 4
j). T zamanı ne zaman olsun
ikinci olay olur. Bu mesajlardan bazıları kötü niyetli oyunculardan gelebilir, ancak
| MSV r, 4 | <t H , bunlardan en az biri HSV r, 4'teki dürüst bir doğrulayıcıdan geliyor ve bir süre sonra gönderiliyor
T r + t 4 . Buna göre, T ≥ T r + t 4 > T r + λ + Λ ≥ β
r, 1
ℓ r + Λ ve zamanla T oyuncusu i de aldı
mesaj m
r, 1
ℓ r . Protokolün oluşturulmasıyla, oyuncu i zamanında durur β

r, 5
ben
= T olmadan
herhangi bir şeyi yaymak; ayarlar B r = B r
ℓ r ; ve kendi CERT setleri r (r, 4) -İletileri için set olmak
0 ve H (B r
ℓ r) almış olduğu.
Adım s> 5. Benzer şekilde, herhangi bir s> 5 adımı ve herhangi bir doğrulayıcı i ∈ HSV r, s , oyuncu i için
HSV r'de doğrulayıcılar tarafından gönderilen tüm mesajları aldı , 4 zamana kadar beklediyse α
r, s
ben
+ t s . Tarafından
aynı analiz, oyuncu i hiçbir şey yaymadan durur, B r = B r
ℓ r (ve kendi
CERT r uygun şekilde). Elbette, kötü niyetli doğrulayıcılar durmayabilir ve keyfi olarak yayılabilir iletiler, ancak | MSV r, s | <t H , indüksiyonla t H doğrulayıcıları tarafından başka hiçbir
v / imzalanamaz
herhangi bir adımda 4 ≤ s / <s, bu nedenle dürüst doğrulayıcılar yalnızca geçerli t H aldıkları için dururlar
(r, 4) -0 ve H (B r
ℓ r).
Round-r Bloğunun Yeniden İnşası. Adım 5'in analizi genel bir dürüstlük için geçerlidir
kullanıcı i neredeyse hiç değişiklik yapmadan. Aslında, i oyuncusu kendi r turunu I r aralığında başlar ve
T anında yalnızca t H için geçerli (r, 4) -H (B r
ℓ r). Yine çünkü
bu mesajlardan en az biri dürüst doğrulayıcılardan geliyor ve T r + t 4 zamanından sonra gönderiliyor , oyuncu i'nin
ayrıca m aldı
r, 1
ℓ r
T zamanına göre B r = B r
ℓ r uygun sertifika ile r .
Geriye kalan tüm dürüst kullanıcıların r + 1 zaman aralığında r turunu bitirdiğini göstermek için kalır .
Adım 5'in analizine göre, her dürüst doğrulayıcı i ∈ HSV r, 5 B r'yi α'da veya ondan önce bilir.
r, 5
ben
+ t 5 ≤
T r + λ + t 5 = T r + 8λ + Λ. T yana , r + 1 i ilk dürüst kullanıcı zaman r B bilir r , sahip
T r + 1 ≤ T r + 8λ + Λ
istediğiniz gibi. Oyuncunun i Dahası, r, B bilen r , o zaten iletileri çoğaltım yardımcı oldu
onun CERT r . Tüm bu mesajların tüm dürüst kullanıcılar tarafından zamanla λ alınacağını unutmayın.
19 Açıkçası, bu çok yüksek bir olasılıkla gerçekleşir, ancak ille de bunaltıcı değildir. ama, bu olasılık, protokolün çalışma süresini biraz etkiler, ancak doğruluğunu etkilemez. H =% 80 olduğunda, | HSV r, 4 | ≥ t , H olasılık 1 - 10 -8 . Bu olay meydana gelmezse, protokol bir başkası için devam edecektir.
3 adım. Bunun iki adımda meydana gelmeme olasılığı ihmal edilebilir olduğundan, protokol Adım 8'de bitecektir.
Beklenti, o zaman, gereken adım sayısı neredeyse 5'tir.
47

Sayfa 48

Oyuncunun i r onları yaymak için ilk oyuncu oldu. Ayrıca, yukarıdaki analizi takiben elimizde
T r + 1 ≥ T r + t 4 ≥ β

r, 1
ℓ r
+ Λ, böylece tüm dürüst kullanıcılar m aldı
r, 1
ℓ r
zamanla T r + 1 + λ. Buna göre,
tüm dürüst kullanıcılar B r'yi I r + 1 = [T r + 1 , T r + 1 + λ] zaman aralığında bilirler .
Son olarak, r = 0 için aslında T 1 ≤ t 4 + λ = 6λ + Λ olur. Her şeyi bir araya getirmek,
Lemma 5 .2 tutar.
■
5.8 Sağlamlık Lemması
Lemma 5. 3. [Sağlamlık Lemması, yeniden ifade edilmiştir] 1–3 Özellikler varsayıldığında, r - 1 turu için
lider ℓ r olasılığını ezici ile kötü niyetli olarak, tüm dürüst kullanıcılar aynı blokta hemfikir
B r , T r + 1 ≤ T r + (6L r + 10) λ + Λ ve tüm dürüst kullanıcılar B r'yi I r + 1 zaman aralığında bilir .
Kanıt. Protokolün iki bölümü olan GC ve BBA'yı ayrı ayrı ele alıyoruz .
GC. Tümevarımsal hipotez ve Lemma 5. 5, herhangi bir s ∈ {2,3,4} adımı ve herhangi bir dürüst doğrulayıcı i ∈ HSV r, s , oyuncu i zamanında hareket ettiğinde β
r, s
ben
= α
r, s
ben
+ t s , gönderilen tüm mesajları aldı
s / <s adımlarında tüm dürüst doğrulayıcılar tarafından . 4. adım için iki olası durumu birbirinden ayırıyoruz.
Durum 1. Doğrulayıcı yok i ∈ HSV r, 4 set g i = 2.
Bu durumda, tanım gereği b i = 1 tüm doğrulayıcılar için i ∈ HSV r, 4 . Yani bir ile başlarlar ikili BA protokolünde 1 üzerinde anlaşma. Onlar kendi v üzerinde anlaşma olmayabilir i , 'ler ancak bu ikili BA'da göreceğimiz gibi önemli değil.
Durum 2. g î = 2 olacak şekilde bir doğrulayıcı î ∈ HSV r, 4 vardır .
Bu durumda şunu gösteriyoruz
(1) tüm i ∈ HSV r, 4 için g ben ≥ 1 ,
(2) tüm i ∈ HSV r, 4 için v i = v / olacak şekilde bir v / değeri vardır ve
(3) geçerli bir mesaj var m
r, 1
ℓ
bazı doğrulayıcılardan ℓ ∈ SV r, 1 öyle ki v / = H (B r
ℓ).
Nitekim, oyuncu î dürüst olduğundan ve g î = 2 ayarladığından , tüm geçerli mesajların 2 / 3'ünden fazlası m
r, 3
j
aynı değer için v / = ⊥ aldı ve v î = v / ayarladı .
Lemma 5. 5'teki Özellik (d) ile , diğer dürüst (r, 4) -verifi i için, bundan daha fazlası olamaz tüm geçerli mesajların 2 / 3'ünden daha fazla m
r, 3
j
i / ' nin aldığı v // = v / ile aynı değer içindir .
Buna göre, eğer g i = 2 yaparsam, v / için de > 2/3 çoğunluk görmüş olmalıyım ve
v i = v / , istenildiği gibi.
Şimdi , g i <2 olan keyfi bir doğrulayıcı i ∈ HSV r, 4 düşünün . Özellik analizine benzer lemma 5 (d) .5, oyuncu ihas görülen nedeni> v 2/3 çoğunluğu / fazla 1
2
| HSV r, 3 | dürüst

(r, 3) -Verifiers v / imzaladı . Çünkü tüm mesajları dürüst (r, 3) - doğrulayıcılardan aldım.
zaman β
r, 4
ben
= α
r, 4
ben
+ t 4 , özellikle 1'den fazla aldı
2
| HSV r, 3 | onlardan mesajlar
v / . Çünkü | HSV r, 3 | > 2 | MSV r, 3 |, i v / için> 1/3 çoğunluk görmüştür . Buna göre oyuncu
i, g i = 1'i ayarlar ve Özellik (1) tutar.
Oyuncu i zorunlu olarak v i = v / ayarlıyor mu? Farklı bir v // = ⊥ değerinin olduğunu
varsayalım, öyle ki
Oyuncu i ayrıca v // için> 1/3 çoğunluk gördü . Bu mesajlardan bazıları kötü amaçlı olabilir
doğrulayıcılar, ancak bunlardan en az biri dürüst bir doğrulayıcı j ∈ HSV r, 3'ten : gerçekten, çünkü
| HSV r, 3 | > 2 | MSV r, 3 | ve kötü amaçlı yazılım grubu olan HSV r, 3'ten tüm iletileri aldım .
Geçerli bir (r, 3) mesajını aldığım doğrulayıcılar, geçerli tüm mesajların <1 / 3'ü için
aldığı mesajlar.

Sayfa 49

Tanım gereği, j oyuncusu tüm geçerli (r, 2) -mesajları arasında v // için> 2/3 çoğunluğu görmüş
olmalıdır.
aldı. Bununla birlikte, diğer bazı dürüst (r, 3) -doğrulayıcıların gördüklerine zaten sahibiz.
V> 2/3 çoğunluk / (onlar v imzalı çünkü /). Lemma 5 .5 Mülkiyetine (d) göre , bu olamaz
olur ve böyle bir v // değeri yoktur. Bu nedenle, i oyuncusu istendiği gibi v i = v / ayarlamalıdır ,
ve Mülk (2) tutar.
Son olarak, bazı dürüst (r, 3) -doğrulayıcıların v / için> 2/3 çoğunluk gördükleri göz önüne
alındığında , bazıları (aslında,
Dürüst (r, 2) -doğrulayıcıların yarısından fazlası v / için imza attı ve mesajlarını yaydı .
Protokolün oluşturulmasıyla, dürüst (r, 2) - doğrulayıcıların geçerli bir
mesaj m
r, 1
ℓ
bazı oyunculardan ℓ ∈ SV r, 1 ile v / = H (B r
ℓ), dolayısıyla Özellik (3) tutar.
BBA ⋆ . Yine iki durumu birbirinden ayırıyoruz.
Durum 1. Tüm doğrulayıcılar i ∈ HSV r, 4 , b i = 1'e sahiptir.
Bu, GC Durum 1'in ardından gerçekleşir. As | MSV r, 4 | <t H , bu durumda SV r, 5'de doğrulayıcı yok
0. bit için tH geçerli (r, 4) -mesajları toplayabilir veya oluşturabilir. Bu nedenle, HSV r, 5'te dürüst bir
doğrulayıcı yok
dururdu çünkü boş olmayan bir B r bloğu biliyor .
Ayrıca, bit 1 için en az tH geçerli (r, 4) -mesajları olmasına rağmen , s / = 5,
s / - 2 ≡ 1 mod 3, dolayısıyla HSV'de hiçbir dürüst doğrulayıcı r, 5 durmayacaktır çünkü B r = B r
ϱ .
Bunun yerine, her doğrulayıcı i ∈ HSV r, 5 aynı anda hareket eder β
r, 5
ben
= α
r, 5
ben
+ t 5 , hepsini aldığında
Lemma 5 .5'ten sonra HSV r, 4 tarafından gönderilen mesajlar Böylece oyuncu 1 için> 2/3 çoğunluk
gördüm

ve b i = 1'i ayarlar .
Bir Madeni Para-Sabit-1 adımı olan 6. Adımda, s / = 5, s / - 2 ≡ 0 mod 3'ü karşılasa da,
yok t H geçerli (r, 4) -bit 0 için mesajlar, dolayısıyla HSV r, 6'da hiçbir doğrulayıcı durmayacaktır
çünkü
boş olmayan bir B r bloğunu bilir . Bununla birlikte, s / = 6 ile s / - 2 ≡ 1 mod 3 ve var
| HSV r, 5 | ≥ t H geçerli (r, 5) - HSV r, 5'ten bit 1 için mesajlar .
Her doğrulayıcı için i ∈ HSV r, 6 , Lemma 5.5'ten sonra , zamanında veya öncesinde α
r, 6
ben
+ t 6 oyuncu i
HSV r, 5'ten tüm mesajları aldı , bu nedenle hiçbir şey yaymadan durur ve ayarlar
B r = B r
ϱ . CERT r , t H geçerli (r, 5) -mesajlar m kümesidir.
r, 5
j
= (ESIG j (1), ESIG j (v j), σ
r, 5
j)
durduğunda onun tarafından alındı.
Sonra, oyuncunun ya adım s> 6'da dürüst bir doğrulayıcı ya da genel bir dürüst kullanıcı olmasına izin
verin (yani,
doğrulayıcı olmayan). Lemma 5 .2'nin ispatına benzer şekilde , oyuncu i, B r = B r
ϱ ve kendi
CERT r , t H geçerli (r, 5) -mesajlar m
r, 5
j
= (ESIG j (1), ESIG j (v j), σ
r, 5
j) sahip
Alınan.
Son olarak, Lemma 5.2'ye benzer şekilde ,
T r + 1 ≤ dk
i∈ HSV r, 6
α
r, 6
ben
+ t 6 ≤ T r + λ + t 6 = T r + 10λ + Λ,
ve tüm dürüst kullanıcılar B r + 1 zaman aralığında bilir , çünkü ilk dürüst kullanıcı
B bilen r onun CERT içinde -İletileri (5 r) çoğaltım yardımcı oldu etmiştir r .
Durum 2. Bir doğrulayıcı var î ∈ HSV r, 4 ve b î = 0.
Bu, GC'nin 2. Durumunu takiben gerçekleşir ve daha karmaşık bir durumdur. GC analizi ile,
bu durumda geçerli bir mesaj var m
r, 1
ℓ
öyle ki v i = H (B r
ℓ) tüm i ∈ HSV r için, 4 . Not
HSV içinde doğrulayıcıları o r, 4 onların b üzerinde anlaşma olmayabilir i 'ler.
Herhangi bir adım s ∈ {5, ..., m + 3} ve doğrulayıcı i ∈ HSV r, s için Lemma 5 .5 oyuncusundan
HSV r, 4 ∪ ⋯ ∪ HSV r, s-1 içindeki tüm dürüst doğrulayıcılar tarafından gönderilen tüm iletileri
aldı eğer beklediyse
zaman için t s .

Sayfa 50

Şimdi aşağıdaki E olayını ele alıyoruz: bir s ∗ ≥ 5 adımı vardır, öyle ki, ilk

zaman ikili BA, bazı oyuncu $i^* \in SV\ r, s^*$ (ister kötü ister dürüst olsun) durmalıdır
hiçbir şey yaymadan. Biz gerçeği vurgulamak için "durdurmalıdır" kullanan, oyuncu i eğer *
kötü niyetli ise, o zaman protokole göre durmaması gerektiğini iddia edebilir ve
Düşmanın tercih ettiği mesajları yaymak.
Ayrıca protokolün oluşturulmasıyla,
(Ea) i^* en az t H geçerli mesaj toplayabilir veya üretebilir m
$r, s'-1$
j
$= (ESIG\ j\ (0), ESIG\ j\ (v),$
σ
$r, s'-1$
j
) aynı v ve s' için , $5 \leq s' \leq s^*$ ve $s' - 2 \equiv 0 \bmod 3$ ile; veya
(Eb) i^* en az t H geçerli mesaj toplayabilir veya üretebilir m
$r, s'-1$
j
$= (ESIG\ j\ (1), ESIG\ j\ (v\ j\ ),$
σ
$r, s'-1$
j
) aynı s' için , $6 \leq s' \leq s^*$ ve $s' - 2 \equiv 1 \bmod 3$ ile.
Çünkü dürüst (r, s' - 1) -mesajlar tüm dürüst (r, s') doğrulayıcıları tarafından onlardan önce alınır.
s' adımında beklemek bitmiştir ve Düşman her şeyi en geç
dürüst kullanıcılar, genelliği kaybetmeden $s' = s^*$ elde ederiz ve i^* oyuncusu kötü niyetli. Bunu not et
Ea'daki v değerinin geçerli bir bloğun karması olmasını şart koşmadık: netleşeceği gibi
analizde, $v = H\ (B\ r$
ℓ) bu alt etkinlikte.
Biz Aşağıda ilk etkinlik E aşağıdaki Davasını 2 analiz ve sonra s değeri olduğunu
göstermektedir * temelde
L r'ye göre dağıtılır (dolayısıyla E olayı, m + 3 Adımından önce çok büyük
parametreler için ilişkiler verilen olasılık). Başlamak için, herhangi bir adım için $5 \leq s < s^*$,
Her dürüst $i \in HSV$ doğrulayıcı r, s bekledi süresi t ler ve set v ı çoğunluğu oy olmak
geçerli (r, s − 1) - aldığı mesajlar. Oyuncu, tüm dürüst (r, s − 1)-mesajları aldığından beri
Lemma 5'i takiben 5, HSV r, 4'teki tüm dürüst doğrulayıcılar H (B r
ℓ) Aşağıdaki Dava
GC'nin 2'si ve $|\ HSV\ r, s-1\ | > 2\ |\ MSV\ r, s-1\ |$ her bir s için, tümevarım yoluyla şu oyuncuya
sahibiz i
ayarlandı
v ben = H (B r
ℓ).
Aynı şey, yayılmadan durmayan her dürüst doğrulayıcı $i \in HSV\ r, s^*$ için de geçerlidir.
herhangi bir şey. Şimdi s^* Adımını ele alıyoruz ve dört alt durumu birbirinden ayırıyoruz.
Durum 2.1.a. Olay Ea olur ve ben doğrulayıcı dürüst vardır $' \in HSV\ s, r$ * should
ayrıca hiçbir şey yaymadan durun.
Bu durumda, $s^* - 2 \equiv 0 \bmod 3$'ümüz var ve Adım s^* , Madeni Para 0'a Sabit bir adımdır. Tarafından
tanım, oyuncu i $'$ en az t H geçerli (r, $s^* - 1$) -formun mesajlarını almıştır
$(ESIG\ j\ (0), ESIG\ j\ (v), \sigma$
$r, s^* - 1$
j
). HSV r'deki tüm doğrulayıcılar , $s^* - 1\ H\ (B\ r$
ℓ) ve
$|\ MSV\ r, s^* - 1\ | < t\ H$, $v = H\ (B\ r$
ℓ).
En az t H - $|\ MSV\ r, s^* - 1\ |$ 0 ve v için i $'$ tarafından alınan (r, $s^* - 1$) -mesajların ≥ 1'i

doğrulayıcılar tarafından HSV'de gönderilir r, s ∗ −1 T r + t s ∗ −1 ≥ T r + t 4 ≥ T r + λ + Λ ≥ β
r, 1
ℓ
+ Λ,
oyuncu i ′ m aldı
r, 1
ℓ
o (r, s ∗ −1) -mesajlarını aldığında. Böylece oyuncu
i ′ hiçbir şey yaymadan durur; ayarlar B r = B r
ℓ ; ve kendi CERT setleri r olmak
0 ve v için aldığı geçerli (r, s ∗ - 1) -mesajları kümesi.
Sonra, diğer herhangi bir doğrulayıcı i ∈ HSV r, s ∗ ' nin ya B r = B r ile durduğunu gösteriyoruz.
ℓ veya
b i = 0 olarak ayarlanmış ve yayılmıştır (ESIG i (0), ESIG i (H (B r
ℓ)), σ
r, s
i). Gerçekten, çünkü Adım s ∗
ilk kez bazı doğrulayıcıların herhangi bir şey yaymadan durması gerektiğidir,
bir adım vardır s ' <s * s ' - 2 ≡ 1 mod 3 t, öyle ki , H (R, S ' - 1) -verifiers 1 imzalamıştır.
Buna göre, HSV r'de doğrulayıcı yok , s ∗ B r = B r ile durur
ϱ .
50

Sayfa 51

Dahası, {4,5, ..., s ∗ - 1} adımlarındaki tüm dürüst doğrulayıcılar H (B r
ℓ), var
bir basamak s bulunamadı ' ≤ s * s ile ' -2 ≡ 0 mod 3, t o , H (R, S ' -verifiers imzalayan -1)
biraz v ′ ′ = H (B r
ℓ) —dizelendi, | MSV r, s ′ −1 | <T , H . Buna göre, HSV'de doğrulayıcı yok , s ∗ durur
B r = B r ile
ϱ ve B r = B r
ℓ . Yani, bir i ∈ HSV r, s ∗ oyuncusu olmadan durduysa
herhangi bir şeyi yaymak için B r = B r ayarlamış olmalı
ℓ .
Bir oyuncu i ∈ HSV r, s ∗ ise
t s zaman bekledi ∗
ve zamanında bir mesaj yaydı
β
r, s ∗
ben
= α
r, s ∗
ben
+ t s ∗ , HSV r, s ∗ −1'den en az
t H - | MSV r, s ∗ −1 | bunlardan 0 ve v. 1 için> 2/3 çoğunluk gördüysem, o
1 için 2'den fazla (t H - | MSV r, s ∗ −1 |) geçerli (r, s ∗ - 1) -mesaj gördü.
2t'den H - 3 | MSV r, s ∗ −1 | Dürüst (r, s ∗ - 1) -doğrultuculardan bunlardan. Ancak bu şu anlama gelir:
| HSV r, s ∗ −1 | ≥ t , H - | MSV R, S * -1 | + 2t H -3 | MSV R, S * -1 | > 2n − 4 | MSV r, s ∗ −1 |, çelişen
gerçek şu ki
| HSV r, s ∗ −1 | + 4 | MSV r, s ∗ −1 | <2n,
parametreler için ilişkilerden gelir. Buna göre> 2/3 görmüyorum
1 için çoğunluk, ve b i = 0 olarak ayarlıyor çünkü s ∗ Madeni Para 0'a Sabit bir adımdır. Sahip olduğumuz gibi

görülen, v i = H (B r
ℓ). Böylece (ESIG i (0), ESIG i (H (B r
ℓ)), σ
r, s
i) istediğimiz gibi
göstermek.
Adım s için * + 1, oyuncunun beri ı ' onun CERT mesajları çoğaltım yardımcı oldu r
zamanında veya öncesinde α
r, s ∗
ben ′
+ t s ∗ , HSV r, s ∗ + 1'deki tüm dürüst doğrulayıcılar en azından
t H geçerli (r, s ∗ - 1) -bit 0 ve H değeri için mesajlar (B r
ℓ) bitiminde veya yapılmadan önce
bekliyor. Ayrıca, HSV r, s ∗ +1 içindeki doğrulayıcılar bunları almadan önce durmayacaktır (r,
s ∗ −1) -
mesajlar, çünkü bit 1 için başka t H geçerli (r, s ′ −1) -mesajları olmadığından
s ' - 2 ≡ 1 mod 3 ve 6 ≤ s ' ≤ s * + 1, Aşama tanımına * . Özellikle Step
s ∗ + 1'in kendisi bir Coin-Fixed-To-1 adımıdır, ancak HSV'de hiçbir dürüst doğrulayıcı r,
s ∗ yayılmamıştır
1 için bir mesaj ve | MSV r, s ∗ | <T , H .
Bu nedenle, HSV r, s ∗ + 1'deki tüm dürüst doğrulayıcılar hiçbir şey yaymadan durur ve B r =
B r
ℓ : daha önce olduğu gibi, m aldılar
r, 1
ℓ
istenen (r, s ∗ −1) -mesajlarını almadan önce .20
Aynı şey, gelecekteki adımlarda tüm dürüst doğrulayıcılar ve genel olarak tüm dürüst kullanıcılar için
söylenebilir.
Özellikle, hepsi bilirler B r = B r
ℓ I r + 1 zaman aralığında ve
T r + 1 ≤ α
r, s ∗
ben ′
+ t s ∗ ≤ T r + λ + t s ∗ .
Durum 2.1.b. Olay Eb olur ve ben doğrulayıcı dürüst vardır ' ∈ HSV s, r * should
ayrıca hiçbir şey yaymadan durun.
Bu durumda s ∗ −2 ≡ 1 mod 3'ümüz var ve Adım s ∗ bir Sikke-Sabit-1 adımıdır. Analiz
Durum 2.1.a'ya benzer ve birçok ayrıntı atlanmıştır.
20 Eğer ℓ kötü niyetli ise, m r, 1 gönderebilir
ℓ
geç, bazı dürüst kullanıcıların / doğrulayıcıların m r almamış olmasını umarak , 1
ℓ
hala
bunun için istenen sertifikayı aldıklarında. Ancak, doğrulayıcı î ∈ HSV r, 4 , b î = 0 ve v î = H (B r
ℓ) olarak
Dürüst doğrulayıcılar i more HSV r, 3'ün yarısından fazlasına sahip olmadan önce , v i = H (B r
ℓ). Bu ayrıca daha fazlasını ima eder
dürüst doğrulayıcıların yarısından fazlası i ∈ HSV r, 2 v i = H (B r
ℓ) ve bu (r, 2) -doğrulayıcıların tümü m r, 1 almış
ℓ
. Olarak
Düşman, bir doğrulayıcıyı, doğrulamayan kişiden ayırt edemez, m r'nin yayılmasını hedefleyemez , 1
ℓ
için (r, 2) -verifiers

Doğrulayıcı olmayanların görmesine gerek kalmadan. Aslında, yüksek olasılıkla, yarıdan fazla (veya iyi bir sabit kesir)
Dürüst kullanıcılar arasından m r, 1
ℓ
kendi turlarının başından itibaren t 2'yi bekledikten sonra r. Buradan itibaren
m r, 1 için gerekli zaman λ ′
ℓ
kalan dürüst kullanıcılara ulaşmak Λ'den çok daha küçüktür ve basit olması için
analize yaz. 4λ ≥ λ ′ ise, analiz herhangi bir değişiklik olmaksızın devam eder: 4.Adımın sonunda, hepsi
dürüst kullanıcılar m r alırdı , 1
ℓ
. Bloğun boyutu çok büyük olursa ve 4λ <λ ′ ise 3. ve 4. Adımlarda,
protokol her doğrulayıcıdan 2λ yerine λ ′ / 2 beklemesini isteyebilir ve analiz tutmaya devam eder.
51

Sayfa 52

Daha önce olduğu gibi, i ′ oyuncusu en az t H geçerli (r, s ∗ - 1) -formun mesajlarını almış olmalıdır
(ESIG j (1), ESIG j (v j), σ
r, s ∗ −1
j
). Yine s ∗ tanımına göre , bir adım yoktur
5 ≤ s ' <s * s ile ' -2 ≡ 0 mod 3, en az bir T , H (R, S ' -1) -verifiers 0 imzalanmış ve adres
aynı v. Dolayısıyla i ′ oyuncusu hiçbir şey yaymadan durur; ayarlar B r = B r
ϱ ; ve setler
Kendi CERT r (r s geçerli kümesi olmak * - 1) o aldığını bit 1 için -İletileri.
Dahası, herhangi bir başka doğrulayıcı i ∈ HSV r, s ∗ ya B r = B r ile durmuştur.
ϱ , veya ayarlayan b i =
1 ve yayılmış (ESIG i (1), ESIG i (v i), σ
r, s ∗
ben
). İ ′ oyuncusu yayılmaya yardımcı olduğundan
(R, S * - 1), onun Cert içinde -İletileri r zaman a ile
r, s ∗
ben ′
+ t s ∗ , yine tüm dürüst doğrulayıcılar
HSV r, s ∗ +1 hiçbir şey yaymadan durur ve B r = B r olarak ayarlayın
ϱ . Benzer şekilde, hepsi dürüst
kullanıcılar bilir B r = B r
ϱ I r + 1 zaman aralığında ve
T r + 1 ≤ α
r, s ∗
ben ′
+ t s ∗ ≤ T r + λ + t s ∗ .
Durum 2.2.a. Ea olayı gerçekleşir ve dürüst bir doğrulayıcı yoktur i ′ ∈ HSV r, s ∗ kim
ayrıca hiçbir şey yaymadan durmalıdır.
Bu durumda, i ∗ oyuncunun geçerli bir CERT r'ye sahip olabileceğine dikkat edin.
i * t oluşan H , istenen
(r, s ∗ - 1) - Düşmanın toplayabileceği veya oluşturabileceği mesajlar. Ancak, kötü niyetli
doğrulayıcılar bu mesajların yayılmasına yardımcı olamayabilir, bu nedenle dürüst olanın
kullanıcılar bunları zamanında alacaktır λ. Aslında | MSV r, s ∗ −1 | Bu mesajların arasında şunlar
olabilir:
kötü niyetli (r, s ∗ - 1) - mesajlarını hiç yaymayan ve yalnızca gönderen doğrulayıcılar
bunları s ∗ adımındaki kötü niyetli doğrulayıcılara iletin .

Durum 2.1.a'ya benzer şekilde, burada $s^* - 2 \equiv 0 \bmod 3$'ümüz var, Adım s^* Madeni Para 0'a Sabit bir adımdır,
ve (R, S * CERT'in içinde -İletileri - 1) r
i^* bit 0 içindir ve v = H (B r
ℓ). Doğrusu, hepsi dürüst
$(r, s^* - 1)$ -düzenleyiciler v işaretler, bu nedenle Düşman t H geçerli $(r, s^* - 1)$ -mesajlar oluşturamaz farklı bir v ′ için .
Dahası, tüm dürüst (r, s^*) doğrulayıcıları t s^* süresini beklediler ve> 2/3 çoğunluğu görmediler bit 1 için, yine çünkü $|HSV\ r, s^* - 1| + 4 |MSV\ r, s^* - 1| < 2n$. Böylece her dürüst doğrulayıcı $i \in HSV\ r, s^*$ ayarlar b ben = 0, v ben = H (B r
ℓ) oy çokluğu ile ve m propagandası
r, s *
ben
=
(ESIG i (0), ESIG i (H (B r
ℓ)), σ
r, s *
ben
) zamanında α
r, s *
ben
+ t s^* .
Şimdi Adım $s^* + 1$'deki (Coin-Fixed-To-1 adımı) dürüst doğrulayıcıları düşünün . Eğer Adversary aslında CERT mesajlar gönderir r
i^* bazılarına ve
dur, sonra Durum 2.1.a'ya benzer şekilde, tüm dürüst kullanıcılar bilir B r = B r
ℓ zaman aralığı içinde
I r + 1 ve
$T\ r + 1 \leq T\ r + \lambda + t\ s^* + 1$.
Aksi takdirde, Adım $s^* + 1$ ' deki tüm dürüst doğrulayıcılar 0 için tüm (r, s^*) -mesajlarını almış ve H (B r
ℓ) HSV'den r, s^* bekleme süresinden sonra t $s^* + 1$, bu da> 2/3 çoğunluğa yol açar, çünkü $|HSV\ r, s^*| > 2 |MSV\ r, s^*|$. Böylece HSV r, $s^* + 1$ içindeki tüm doğrulayıcılar mesajlarını 0 ve H (B r
ℓ) buna göre. HSV r, $s^* + 1$ içindeki doğrulayıcıların B r = B r ile bitmediğini unutmayın.
ℓ ,
çünkü Adım $s^* + 1$, Madeni Para 0'a Sabit adım değildir.
Şimdi, Adım $s^* + 2$'deki dürüst doğrulayıcıları düşünün (bu bir Coin-Genuinely-Flipped adımıdır). Adversary CERT iletileri gönderirse r
i * , bunlardan bazılarına ve durdurmak için bunları neden olur
sonra yine tüm dürüst kullanıcılar bilir B r = B r
ℓ I r + 1 zaman aralığında ve
$T\ r + 1 \leq T\ r + \lambda + t\ s^* + 2$.

Sayfa 53

Aksi takdirde, Adım $s^* + 2$'deki tüm dürüst doğrulayıcılar için tüm $(r, s^* + 1)$ -mesajlarını almışlardır.
0 ve H (B r
ℓ) HSV'den r, $s^* + 1$ bekleme süresinden sonra t $s^* + 2$, bu da> 2/3 çoğunluğa yol açar.
Böylece hepsi mesajlarını 0 ve H (B r
ℓ) buna göre: onlar yaparlar
bu durumda “yazı tura atmayın”. Yine, yayılmadan durmadıklarını unutmayın,
çünkü Adım $s^* + 2$, Madeni Para 0'a Sabit adım değildir.

Son olarak, Adım s ∗ + 3'teki dürüst doğrulayıcılar için (bu da 0'a sabitlenmiş bir jeton adımıdır), hepsi
Bunlardan en az t alacağı H 0 ve H (B için geçerli iletileri r
ℓ) HSV s ∗ +2'den ,
eğer gerçekten beklerlerse t s ∗ +3 . Böylece, Düşmanın mesajları gönderip göndermeyeceği CERT'de r
ben ∗
bunlardan herhangi birine, HSV r'deki tüm doğrulayıcılar , s ∗ + 3 , B r = B r ile durur
ℓ , olmadan
her şeyi yaymak. Düşmanın nasıl davrandığına bağlı olarak, bazılarının
Kendi Cert r o oluşan (R, S * CERT'in olarak -İletileri -1) r
i ∗ ve diğerleri var
Kendi Cert r o oluşan (R, S * + 2) -İletileri. Her durumda, tüm dürüst kullanıcılar
bilmek B r = B r
ℓ I r + 1 zaman aralığında ve
T r + 1 ≤ T r + λ + t s ∗ +3 .
Durum 2.2.b. Eb olayı gerçekleşir ve dürüst bir doğrulayıcı yoktur i ′ ∈ HSV r, s ∗ kim ayrıca hiçbir şey yaymadan durmalıdır.
Bu durumda analiz, Durum 2.1.b ve Durum 2.2.a'dakilere benzer, dolayısıyla birçok ayrıntı ihmal edildi. Özellikle, CERT r
i ∗ t H istenen (r, s ∗ - 1) -mesajlarından oluşur
Düşmanın toplayabildiği veya üretebildiği 1. bit için, s ∗ - 2 ≡ 1 mod 3, Adım s ∗ bir Coin-Fixed-To-1 adım ve hiçbir dürüst (r, s ∗) -verifier 0 için> 2/3 çoğunluk görmezdi.
Böylece, her doğrulayıcı i ∈ HSV r, s ∗ , b i = 1'i ayarlar ve m'yi yayar.
r, s ∗
ben
= (ESIG i (1), ESIG i (v i),
σ
r, s ∗
ben
) zamanında α
r, s ∗
ben
+ t s ∗ . Durum 2.2.a'ya benzer şekilde, en fazla 3 adımda (yani protokol ulaştığında adım s * , +3, bir Para-Sabit için-1 adım) olan tüm dürüst kullanıcı B bilmek r = B r
ϱ
zaman aralığı içinde r + 1 . Ayrıca, T r + 1 ≤ T r + λ + t s ∗ +1 veya ≤ T r + λ + t s ∗ +2 olabilir , veya ≤ T r + λ + t s ∗ +3 , dürüst bir doğrulayıcının ilk kez ne zaman durabileceğine bağlı olarak yayılmadan.
Dört alt durumu birleştirdiğimizde, tüm dürüst kullanıcıların zaman aralığı içinde B r'yi bildiğini görüyoruz.
I r + 1 ile
Durum 2.1.a ve 2.1.b'de T r + 1 ≤ T r + λ + t s ∗ ve
T r + 1 ≤ T r + λ + t s ∗ +3 Durum 2.2.a ve 2.2.b'de.
Üst sınır s ∗ ve dolayısıyla Durum 2 için T r + 1 olarak kalır ve bunu nasıl yapacağımızı düşünerek yaparız.
Çoğu zaman Coin-Genuinely-Flipped adımları aslında protokolde yürütülür: yani,
bazı dürüst doğrulayıcılar aslında yazı tura attılar.
Özellikle, bir Coin-Genuinely-Flipped step s ′ (yani, 7 ≤ s ′ ≤ m + 2 ve
s ' - 2 ≡ 2 mod 3) ve izin ℓ '
arg min
j∈ SV r, s′ − 1 H (σ
r, s ′ −1
j
). Şimdilik s ′ <s ∗ varsayalım ,

Aksi hiçbir dürüst doğrulayıcı aslında Adım ler bir para çevirir çünkü ' önceki göre,
tartışmalar.
SV r, s ′ −1 tanımına göre , ℓ ′ kimlik bilgilerinin hash değeri de en küçük olanıdır.
PK r − k içindeki tüm kullanıcılar . Karma işlevi rastsal kahin, ideal olarak oyuncu ℓ olduğundan ' ile
dürüst
olasılık en az h. Daha sonra göstereceğimiz gibi, Düşman elinden gelenin en iyisini yapsa bile
rastgele oracle çıktısı ve olasılığı eğin, oyuncu ℓ ′ hala olasılık konusunda dürüst
53

Sayfa 54

en az p h = h 2 (1 + h - h 2). Aşağıda bunun gerçekten olduğu durumu ele alıyoruz: Yani,
ℓ ′ ∈ HSV r, s ′ −1 .
Her dürüst doğrulayıcı i ∈ HSV r, s ″ nin HSV r, s ′ −1'den gelen tüm mesajları aşağıdaki yollarla
aldığına dikkat edin :
zaman α
r, s ′
ben
+ t s ′ . Eğer oyuncunun yazı tura atması gerekiyorsa (yani,> 2/3 çoğunluğu görmemiştir.
aynı bit b ∈ {0,1}), sonra b i = lsb (H (σ
r, s ′ −1
ℓ ′
)). Başka bir dürüst varsa
doğrulayıcı i ′ ∈ HSV r, s ′ bir bit için> 2/3 çoğunluğu gören b ∈ {0,1}, sonra Mülk tarafından
(d) lemma 5 .5, HSV hiçbir dürüst doğrulayıcı r, s ' görmüş olur> biraz 2/3 çoğunluğu
b ′ = b. Lsb (H (σ
r, s ′ −1
ℓ ′
)) = b, 1/2 olasılıkla, HSV r, s ′ erişimindeki tüm dürüst doğrulayıcılar
b üzerinde 1/2 olasılıkla bir anlaşma. Tabii ki, böyle bir doğrulayıcı i ′ yoksa, o zaman hepsi
HSV r'deki dürüst doğrulayıcılar , s ′ lsb (H (σ
r, s ′ −1
ℓ ′
)) olasılıkla 1.
ℓ olasılığını birleştiren ' HSV ∈ r, s ' -1 , HSV dürüst kontrol hizmeti olduğunu var r, s '
en azından olasılıkla bir bit b ∈ {0,1} üzerinde bir anlaşmaya varmak
p h
2
=
h 2 (1 + h − h 2)
2
. Dahası,
daha önce olduğu gibi oy çokluğu üzerine tümevarım, HSV tüm dürüst kontrol hizmeti r, s ' onların v
var ben s set'
H olmak (B r
ℓ). B anlaşma içinde ulaşıldığında Böylece, Aşama s ' , T , r + 1 olduğu
ya ≤ T r + λ + t s ′ +1 veya ≤ T r + λ + t s ′ +2 ,
b = 0 veya b = 1 olmasına bağlı olarak, Durum 2.1.a ve 2.1.b'nin analizini takiben. İçinde
özellikle, başka bir Coin-Genuinely-Flipped adımı yürütülmeyecek:
bu tür adımlar yine de doğrulayıcı olduklarını kontrol eder ve bu nedenle bekler, ancak hepsi olmadan
dururlar.
her şeyi yaymak. Buna göre, s ∗ Adımından önce , Coin-Genuinely-
Çevrilen adımlar, L r rastgele değişkenine göre dağıtılır . Letting Adım s ′
Protokolün oluşturulmasıyla L r'ye göre son Coin-Genuinely-Flipped adımı olun
sahibiz
s ′ = 4 + 3L r .

Rakip, T r + 1'i şu kadar geciktirmek istiyorsa ∗ Adımını ne zaman yapmalıdır?
mümkün? Düşmanın L r'nin gerçekleşmesini önceden bildiğini bile varsayabiliriz . Eğer
s ∗ > s ′ o zaman faydasızdır, çünkü dürüst doğrulayıcılar şu anda bir anlaşmaya varmışlardır.
Adım s ′ . Emin olmak için, bu durumda s ∗ , yine b = 0 olup olmamasına bağlı olarak s ′ +1 veya s ′ +
2 olacaktır.
veya b = 1. Ancak, bu aslında Durum 2.1.a ve 2.1.b'dir ve sonuçta ortaya çıkan T r + 1 tam olarak
bu durumda olduğu gibi. Daha kesin,
T r + 1 ≤ T r + λ + t s ∗ ≤ T r + λ + t s ′ +2 .
S Eğer * <s ' - -yani 3, s * sondan ikinci bozuk parayla Genuinely-Flipped tarafından daha sonra üvey
öncedir
Durum 2.2.a ve 2.2.b'nin analizi,
T r + 1 ≤ T r + λ + t s ∗ +3 <T r + λ + t s ′ .
Yani, Düşman aslında B r üzerindeki anlaşmanın daha hızlı gerçekleşmesini sağlıyor.
Eğer s ∗ = s ′ - 2 veya s ′ - 1 ise - yani, Coin-Fixed-To-0 adımı veya Coin-Fixed-To-1 adımı
hemen önce Basamak s ' - o zaman dört alt vakaların analizi ile, dürüst kontrol hizmeti içinde
Adım s ′ artık bozuk paraları çeviremezsiniz, çünkü ya yayılmadan durmuşlardır,
veya aynı bit için> 2/3 çoğunluk görmüşler b. Bu nedenle biz var
T r + 1 ≤ T r + λ + t s ∗ +3 ≤ T r + λ + t s ′ +2 .
54

Sayfa 55

Özetle, s ∗ ne olursa olsun , bizde
T r + 1 ≤ T r + λ + t s ′ +2 = T r + λ + t 3L r +6
= T r + λ + (2 (3L r + 6) - 3) λ + Λ
= T r + (6L r + 10) λ + Λ,
göstermek istediğimiz gibi. En kötü durum, s ∗ = s ′ - 1 ve Durum 2.2.b'nin gerçekleşmesidir.
İkili BA protokolünün Durum 1 ve 2'sini birleştiren Lemma 5 .3 tutar.
■
Tohum Q 5.9 Güvenlik r Dürüst Lider ve Olasılık
Lemma 5 .4'ü kanıtlamaya devam ediyor . R turundaki doğrulayıcıların PK r − k ve
Q r − 1 miktarına göre seçilir . Yeniden inceleme parametresini kullanıma sunmanın nedeni k
R - k turunda, Düşmanın yeni kötü niyetli kullanıcılar ekleyebildiğinden emin olmaktır.
PK r − k'ye göre , ihmal edilebilir olasılık dışında Q r − 1 miktarını tahmin edemez . Unutmayın ki
karma işlevi rastgele bir oracle'dır ve Q r − 1 , r turu için doğrulayıcıları seçerken girdilerinden biridir.
Böylelikle, Adversary'ın bakış açısından, PK r-k'ye ne kadar kötü niyetli kullanıcılar
eklenirse eklensin, her biri
bunlardan biri gerekli olasılıkla p (veya
Adım 1 için p 1). Daha doğrusu, aşağıdaki lemmaya sahibiz.
Lemma 5.6. K = O (log 1/2 F) ile, her r turu için, ezici bir olasılıkla Düşman
r - k turundaki rastgele oracle'a Q r − 1'i sorgulamadı .
Kanıt. Tümevarımla ilerliyoruz. Her γ <r turu için, Düşmanın sorgulamadığını varsayın
Q γ − 1 , γ - k turunda geri dönen rastgele kahine.21 Aşağıdaki zihinsel oyunu düşünün
R - k turundaki Rakip, Q r − 1'i tahmin etmeye çalışıyor .
Her turun 1. Adımında γ = r - k, ..., r - 1, rastgele sorgulanmayan belirli bir Q γ − 1 verildiğinde
oracle, oyunculara H hash değerlerine göre i has PK γ − k sipariş ederek (SIG i (γ, 1, Q γ − 1))
artan bir şekilde, PK γ − k üzerinde rastgele bir permütasyon elde ederiz . Tanım olarak, lider
ℓ y olduğunu
permütasyondaki ilk kullanıcı ve olasılıkla dürüst h. Dahası, PK γ − k büyük olduğunda
yeterli, herhangi bir x ≥ 1 tamsayısı için permütasyondaki ilk x kullanıcının tümünün olma olasılığı
kötü niyetli ancak (x + 1) st dürüsttür (1 - h) x h.
ℓ Eğer γ dürüst, daha sonra Q, γ = H (SIG ℓ γ (S γ-1), γ). Düşman imzayı taklit edemeyeceği için
ℓ γ , Q γ , Düşmanın bakış açısından rastgele tekdüze olarak dağıtılır ve hariç
üssel olarak küçük olasılıkla,22 , r - k turunda H'ye sorgulanmadı. Her biri
Sırasıyla Q γ + 1 , Q γ + 2 , ..., Q r − 1 , Q γ , Q γ + 1 , ..., Q r − 2 girişlerinden biri olan H'nin
çıkışıdır ,

hepsi Düşmana rastgele görünüyor ve Düşman, Q r − 1'den H'ye kadar sorgulayamazdı.
yuvarlak r - k.
Buna göre, Düşmanın , raundda iyi bir olasılıkla Q r prob 1'i tahmin edebildiği tek durum
r − k, tüm liderlerin ℓ r − k , ..., ℓ r − 1 kötü niyetli olduğu zamandır. Yine bir γ ∈ {r − k ..., r − 1}
ve karşılık gelen karma değerlerle indüklenen PK γ − k üzerindeki rastgele permütasyon . Bazıları için
x ≥ 2, permütasyondaki ilk x - 1 kullanıcıların tümü kötü niyetli ve x-th dürüst, sonra
Adversary S x olası seçenek vardır γ ya (SIG bir şekilde, H: i (S γ-1 i biridir, γ)),
21 k küçük bir tam sayı olduğundan, genellik kaybı olmaksızın protokolün ilk k turlarının çalıştırıldığı
varsayılabilir.
güvenli bir ortamda ve tümevarımsal hipotez bu turlar için geçerlidir.
22 Yani, H'nin çıktısının uzunluğunda üsteldir. Bu olasılığın F'den çok daha küçük olduğuna dikkat
edin.

Sayfa 56

i oyuncusunu − turunun lideri yaparak ilk x − 1 kötü niyetli kullanıcılar; veya H (Q γ − 1 , γ) ile
zorlama B γ = B
γ
ǫ . Aksi takdirde, γ turunun lideri permütasyondaki ilk dürüst kullanıcı olacaktır.
ve Q r-1 , Düşman için tahmin edilemez hale gelir.
Rakip, yukarıdaki x Q γ seçeneklerinden hangisini takip etmelidir? Düşmana yardım etmek için
bu soruyu cevaplayın, zihinsel oyunda onu aslında ondan daha güçlü yapıyoruz
Şöyleki. Her şeyden önce, gerçekte, Düşman, dürüst bir kullanıcının karmasını hesaplayamaz.
imza, bu nedenle her Q γ için başlangıçta kötü niyetli kullanıcıların x (Q γ) sayısına karar veremez
γ + 1 turundaki rastgele permütasyonun Q γ tarafından indüklendiği . Zihinsel oyunda ona veriyoruz
sayılar x (Q γ) ücretsiz. İkincisi, gerçekte, permütasyondaki ilk x kullanıcının tümü
kötü niyetli olmak, hepsinin lider haline getirilebileceği anlamına gelmez, çünkü hash
imzalarının değerleri de p 1'den küçük olmalıdır . Bu kısıtlamayı zihinsel olarak görmezden geldik.
Oyun, Düşmana daha da fazla avantaj sağlıyor.
Zihinsel oyunda görmek kolaydır, Q ile gösterilir hasım için optimal seçenek, y ,
rastgele başlangıcında en uzun kötü niyetli kullanıcı dizisini üreten
γ + 1. turdaki permütasyon Nitekim, belirli bir Q γ verildiğinde , protokol Q γ − 1'e bağlı değildir.
artık ve Rakip, yalnızca γ + 1 turundaki yeni permütasyona odaklanabilir.
başlangıçta kötü niyetli kullanıcı sayısı için aynı dağıtım. Buna göre her turda
γ, yukarıda bahsedilen ˆQ γ ona Q γ + 1 için en fazla sayıda seçeneği verir ve böylece maksimize eder
Ardışık liderlerin kötü niyetli olma olasılığı.
Bu nedenle, zihinsel oyunda Düşman, r - k turundan bir Markov Zincirini takip ediyor.
r - 1 durum uzayı {0} ∪ {x: x being 2} olmak üzere yuvarlamak için. Durum 0, şu gerçeği temsil eder:
γ geçerli raunddaki rastgele permütasyondaki ilk kullanıcı dürüsttür, bu nedenle Rakip,
Q r − 1'i tahmin etmek için oyun ; ve x ≥ 2 durumlarının her biri, içindeki ilk x - 1 kullanıcının
permütasyon kötü amaçlıdır ve x-th dürüsttür, bu nedenle Düşman'ın Q γ için x seçeneği vardır . The
geçiş olasılıkları P (x, y) aşağıdaki gibidir.
• Herhangi bir y ≥ 2 için P (0,0) = 1 ve P (0, y) = 0'dır.
permütasyondaki kullanıcı dürüst olur.
• P (x, 0) = h x herhangi bir x ≥ 2 için. Yani, h x olasılıkla , tüm x rastgele permütasyonlarının sahip
olduğu
ilk kullanıcıları dürüst olur, bu nedenle Adversary sonraki turda oyunu başarısız olur.
• Herhangi bir x ≥ 2 ve y ≥ 2 için, P (x, y) x rastgele permütasyonlar arasında olasılıktır.
Q γ ' nin x seçeneğinden kaynaklanan , en uzun kötü niyetli kullanıcı dizisi
bazıları y - 1'dir, dolayısıyla Düşman'ın sonraki turda Q γ + 1 için y seçeneği vardır . Yani,
P (x, y) = (
y − 1
∑
i = 0
(1 - h) i h)

x
- (
y − 2
∑
i = 0
(1 - h) i h)
x
= (1 - (1 - h) y) x - (1 - (1 - h) y − 1) x .
0 durumunun, P geçiş matrisindeki ve diğer tüm durumlardaki benzersiz soğurma durumu olduğuna dikkat edin.
x'in 0'a gitme olasılığı pozitiftir.
Markov Zincirinin çok büyük bir olasılıkla 0'a yakınsaması için gerekli mermi: yani hayır
Zincir hangi durumda başlarsa başlasın, rakip büyük bir olasılıkla oyunu kaybeder.
ve r - k turunda Q r − 1'i tahmin edemiyor .
P (2) geçiş matrisini düşünün
İki turdan sonra P · P. P (2) (0,0) = 1 olduğunu görmek kolaydır.
ve P (2) (0, x) = 0 herhangi bir x ≥ 2 için. Herhangi bir x ≥ 2 ve y ≥ 2 için, P (0, y) = 0 olarak,
P (2) (x, y) = P (x, 0) P (0, y) + ∑
z≥2 P (x, z) P (z, y) = ∑
z≥2
P (x, z) P (z, y).

Sayfa 57

¯h bırakma
1 - h, biz var
P (x, y) = (1 -
¯
h y) x - (1 - ¯h y − 1) x
ve
P (2) (x, y) = ∑
z≥2
[(1 -
¯
h z) x - (1 - ¯h z − 1) x] [(1 -
¯
h y) z - (1 - ¯h y − 1) z].
Aşağıda limitini hesaplıyoruz
P (2) (x, y)
P (x, y)
h 1'e giderken - yani ¯h 0'a gider.
P (x, y) 'deki ¯h mertebesi, x katsayısı ile ish y − 1'dir . Buna göre,
lim
h → 1
P (2) (x, y)
P (x, y)
= lim
¯h → 0
P (2) (x, y)
P (x, y)
= lim
¯h → 0
P (2) (x, y)
x¯h y − 1 + O (¯h y)
= lim

¯h → 0 ∑ z≥2 [x¯h z − 1 + O (¯h z)] [z¯h y − 1 + O (¯h y)]
x¯h y − 1 + O (¯h y)
= lim
¯h → 0
2x¯h y + O (¯h y + 1)
x¯h y − 1 + O (¯h y)
= lim
¯h → 0
2x¯h y
x¯h y − 1
= lim
¯h → 0
2¯h = 0.
H 1'e yeterince yakın olduğunda,23 sahibiz
P (2) (x, y)
P (x, y)
≤
1
2
herhangi bir x ≥ 2 ve y ≥ 2 için. Tümevarım yoluyla, herhangi bir k> 2 için, P (k)
P k öyle ki
• Herhangi bir x ≥ 2 için P (k) (0,0) = 1, P (k) (0, x) = 0 ve
• herhangi bir x ≥ 2 ve y ≥ 2 için,
P (k) (x, y) = P (k − 1) (x, 0) P (0, y) + ∑
z≥2
P (k − 1) (x, z) P (z, y) = ∑
z≥2
P (k − 1) (x, z) P (z, y)
≤ ∑
z≥2
P (x, z)
2 k − 2
· P (z, y) =
P (2) (x, y)
2 k − 2
≤
P (x, y)
2 k − 1
.
P (x, y) ≤ 1 olarak, 1 − log 2 F yuvarlamasından sonra, herhangi bir y ≥ 2 durumuna geçiş olasılığı ihmal edilebilir,
herhangi bir x ≥ durumu ile başlayarak 2. y gibi birçok durum olmasına rağmen, bunu görmek kolaydır.
lim
y → + ∞
P (x, y)
P (x, y + 1)
= lim
y → + ∞
(1 -
¯

h y) x - (1 - ¯h y − 1) x
(1 -
¯

h y + 1) x - (1 - ¯h y) x

= lim
y → + ∞
h y − 1 - ¯h y
h y - ¯h y + 1
=
1
h
=
1
1 - s
.
Bu nedenle, P geçiş matrisinin her x satırı, 1 oranlı geometrik bir dizi olarak azalır.
1 − s
> 2
y yeterince büyük olduğunda ve aynı durum P (k) için de geçerlidir . Buna göre, k yeterince büyük olduğunda ancak yine de
log 1/2 F mertebesinde , ≥ y≥2 P (k) (x, y) <F herhangi bir x ≥ 2 için
Düşman oyunu kaybeder ve r - k turunda Q r − 1'i tahmin edemez. H ∈ (2/3, 1] için bir daha karmaşık analiz, yeterli olacak şekilde 1 / 2'den biraz daha büyük bir sabit C olduğunu gösterir.
k = O (log C F) almak için. Böylece Lemma 5 .6 geçerli.
■
Lemma 5.4 . (yeniden düzenlenmiş) Verilen Özellikler r'den önceki her raunt için 1–3, L r için p h = h 2 (1 + h - h 2) ,
ve lider ℓ r olasılıkla dürüsttür, en azından p h .
23 Örneğin, belirli parametre seçimlerinin önerdiği gibi h =% 80.
57

Sayfa 58

Kanıt. Lemma 5 .6'yı takiben , Rakip, Q r − 1'i r - k turunda geri tahmin edemez.
ihmal edilebilir olasılık. Bunun dürüst bir liderin olasılığının h için olduğu anlamına gelmediğini unutmayın.
her turda. Aslında, Q r − 1 verildiğinde , başlangıçta kaç kötü niyetli kullanıcının olduğuna bağlı olarak
PK r − k'nin rastgele permütasyonu , Rakip, Q r için birden fazla seçeneğe sahip olabilir ve böylelikle r + 1 turunda kötü niyetli bir liderin olma olasılığını artırabilir - yine de ona
Analizi basitleştirmek için Lemma 5. 6'daki gibi bazı gerçekçi olmayan avantajlar .
Ancak, R - k turunda Rakip tarafından H'ye sorgulanmayan her Q r − 1 için,
herhangi bir x ≥ 1, (1 - h) x − 1 h olasılıkla ilk dürüst kullanıcı sonuçta x konumunda ortaya çıkar rasgele PK r − k permütasyonu . X = 1 olduğunda, r + 1 turunda dürüst bir liderin olasılığı gerçekten h; x = 2 olduğunda, Düşmanın Q r için iki seçeneği vardır ve sonuçta ortaya çıkan olasılık h 2 . Sadece bu iki durumu göz önünde bulundurarak, yuvarlakta dürüst bir lider olma olasılığına sahibiz.
r + 1, istendiği gibi en az h · h + (1 - h) h · h 2 = h 2 (1 + h - h 2) 'dir.
Yukarıdaki olasılığın sadece r − k turundan gelen protokoldeki rastgeleliği dikkate aldığına dikkat edin.
r yuvarlak Yuvarlak r yuvarlak 0 ile her rastgele dikkate alındığında, Q, R-1 olan
Düşman için daha az tahmin edilebilir ve r + 1 turunda dürüst bir lider olma olasılığı şu şekildedir:
en az h 2 (1 + h - h 2). R + 1'i r ile değiştirmek ve her şeyi bir tur geriye kaydırmak, lider ℓ r
en az h 2 (1 + h - h 2) olasılıkla dürüsttür .
Benzer şekilde, her Coin-Genuinely-Flipped adımında, o adımın "lideri" - doğrulayıcı budur.
kimlik bilgileri en küçük hash değerine sahip SV r'de, en az h 2 (1 +
h - h 2). Böylece L r ve Lemma 5 .4 için p h = h 2 (1 + h - h 2) geçerlidir.

■
6 Algorand ′
2
Bu bölümde, aşağıdaki varsayım altında çalışan bir Algorand ′ sürümü oluşturuyoruz .
Kullanıcıların Dürüst Çoğunluğu Varsayımı: Her PK r'deki kullanıcıların 2 / 3'ünden fazlası dürüsttür.
Bölüm 8'de , yukarıdaki varsayımı istenen Dürüst Çoğunluk ile nasıl değiştireceğimizi gösteriyoruz.
Para varsayımı.
6.1 Algorand için Ek Gösterimler ve Parametreler ′
2
Notasyonlar
• μ ∈ Z + : çok büyük bir olasılıkla, adım sayısının pragmatik bir üst sınırı,
aslında bir turda alınacak. (Göreceğimiz gibi, μ parametresi kaç tane kısa ömürlü olduğunu kontrol eder.
kullanıcının her tur için önceden hazırladığı tuşlar.)
• L r : her biri 1'i görmek için gereken Bernoulli denemelerinin sayısını temsil eden rastgele bir değişken
deneme 1 olasılıkla
p h
2
. L r , oluşturmak için gereken süreyi üst sınırlamak için kullanılacaktır.
blok B r .
• t H : r turunun s> 1 adımında dürüst doğrulayıcıların sayısı için bir alt sınır, öyle ki
ezici bir olasılık (n ve p verildiğinde), SV r, s'de > t H dürüst doğrulayıcılar vardır .
Parametreler
• Çeşitli parametreler arasındaki ilişkiler.
- R turunun her s> 1 adımı için n seçilir, öyle ki, çok büyük bir olasılıkla,

---

Sayfa 59

| HSV r, s | > t H
ve
| HSV r, s | + 2 | MSV r, s | <2t , H .
Yukarıdaki iki eşitsizliğin birlikte | HSV r, s | > 2 | MSV r, s |: işte burada
seçilen doğrulayıcılar arasında 2/3 dürüst çoğunluktur.
H'nin değeri 1'e ne kadar yakınsa, n'nin o kadar küçük olması gerekir. Özellikle, kullanıyoruz (varyantlar
of) İstenilen koşulların ezici bir olasılıkla geçerli olmasını sağlamak için Chernoff sınırları.
• Önemli parametrelerin örnek seçimleri.
- F = 10 −18 .
- n ≈ 4000, t H ≈ 0.69n, k = 70.
6.2 Algorand'de Geçici Anahtarları Uygulama
′
2
Bir doğrulayıcı i ∈ SV r'nin mesajını dijital olarak imzaladığını hatırlayın.
r, s
ben
r turundaki s adımının,
geçici bir genel anahtar pk
r, s
i , geçici bir gizli anahtar sk kullanarak
r, s
ben
O derhal yok eder
kullandıktan sonra. Bir turun atabileceği olası adımların sayısı belirli bir
tamsayı μ, biz zaten geçici anahtarları pratik olarak nasıl kullanacağımızı görmüştük. Örneğin biz

Algorand'da açıkladı ′
1 (burada μ = m + 3), tüm olası geçici anahtarlarını işlemek için,
yuvarlak R ' yuvarlak r ' + 10 6 , i bir çift (PMK SMK), PMK ortak ana üretir
bir kimlik tabanlı imza şemasının anahtarı ve SMK ona karşılık gelen gizli ana anahtar. Kullanıcı i
PMK'yı duyurur ve her olası geçici genel anahtarın gizli anahtarını oluşturmak için SMK'yı kullanır
(ve bunu yaptıktan sonra SMK'yı yok eder). İlgili anahtarlar için i'nin geçici genel anahtarları
mermi S = {i} × {r ′ , ..., r ′ + 10 6 } × {1, ..., μ} şeklindedir. (Tartışıldığı gibi, r ′ +
10 6 turu yaklaştıkça,
i çiftini (PMK, SMK) "yeniler".)
Pratikte, μ yeterince büyükse, bir tur Algorand ′
2 , μ adımdan fazlasını almayacaktır. İçinde
ilke, bununla birlikte, bazı raundlar için adımların sayısının çok uzak bir olasılığı vardır.
gerçekte alınan μ değerini geçecektir. Bu olduğunda mesajını imzalayamayacağım m
r, s
ben
için
herhangi bir adım s> μ, çünkü r turu için önceden sadece μ gizli anahtarlar hazırlamıştır. Üstelik o
daha önce tartışıldığı gibi, yeni bir geçici anahtar zulası hazırlayıp tanıtamadı. Aslında yapmak
bu nedenle, yeni bir bloğa yeni bir genel ana anahtar PMK ′ eklemesi gerekir . Ama r yuvarlamalıdır
daha fazla adım atın, yeni bloklar oluşturulmayacaktır.
Ancak çözümler var. Örneğin, r, pk turunun son geçici anahtarını kullanabilirim.
r, μ
ben
,
aşağıdaki gibi. (1) başka bir
ana anahtar çifti (PMK, SMK); (2) bu çifti başka bir, örneğin 10 6 geçici anahtar oluşturmak için
kullanmak,
sk
r, μ + 1
ben
, ..., sk
r, μ + 10 6
ben
Adımlar μ + 1 karşılık gelen, ..., μ + 10 6 tur r; (3) sk kullanarak
r, μ
ben
dijital olarak
PMK işareti (ve eğer i ∈ SV r, μ ise herhangi bir (r, μ) -mesaj), pk'ye göre
r, μ
ben
; ve (4) SMK ve sk silme
r, μ
ben
.
S ∈ {1, ..., 10 6 } ile μ + s adımında doğrulayıcı olmalı mıyım , sonra dijital olarak onun (r, μ + s) -
mesaj m
r, μ + s
ben
yeni anahtarına göre
r, μ + s
ben
= (i, r, μ + s). Tabii ki, bu imzayı doğrulamak için
i'de, diğerlerinin bu genel anahtarın i'nin yeni genel ana anahtarının PMK'sına karşılık geldiğinden
emin olması gerekir.
Böylece, bu imzaya ek olarak, PMK'nın dijital imzasını pk'ye göre iletiyorum.

r, μ
ben
.
Tabii ki, bu yaklaşım gerektiği kadar tekrar edilebilir, r turu devam ederse
daha fazla adım için! Son geçici gizli anahtar, yeni bir ana halkın kimliğini doğrulamak için kullanılır
anahtar ve dolayısıyla r turu için başka bir geçici anahtar zulası. Ve bunun gibi.

Sayfa 60

6.3 Gerçek Protokol Algoritması ve ′
2
Bir r turunun her adımında, bir doğrulayıcı i ∈ SV r'nin uzun vadeli kamu sırrını kullandığını tekrar hatırlayın.
kimlik bilgilerini üretmek için anahtar çifti, σ
r, s
ben
SIG i (r, s, Q r − 1) ve s = 1 durumunda SIG i (Q r − 1).
Doğrulayıcı i, geçici anahtar çiftini kullanır (pk
r, s
ben , sk
r, s
i), olabilecek başka herhangi bir mesajı imzalamak
gereklidir. Basit olması için sig pk r, s yerine esig i (m) yazıyoruz
ben
(m), i'nin gerçek geçici olduğunu belirtmek için
Bu adımda m imzası ve SIG pk r, s yerine ESIG i (m) yazın
ben
(m)
(ben, m, esig i (m)).
1. Adım: Teklifi Engelleyin
Her kullanıcı için talimatlar i ∈ PK r − k : Kullanıcı i, sahip olduğu anda kendi r turunun 1. Adımını başlatır.
CERT r − 1 , i'nin H (B r − 1) ve Q r − 1'i açık bir şekilde hesaplamasına izin verir .
• Kullanıcı i, i ∈ SV r, 1 olup olmadığını kontrol etmek için Q r − 1 kullanır . İ / ∈ SV r, 1 ise Adım 1 için hiçbir şey yapmaz.
• Eğer i ∈ SV r, 1 ise , yani eğer i potansiyel bir lider ise, o zaman aşağıdakileri yapar.
(a) B 0 , ..., B r − 1'i gördüysem (herhangi bir B j = B
j
ọ hash değerinden kolayca türetilebilir
CERT j'de ve bu nedenle "görüldüğü" varsayılır), daha sonra
şimdiye kadar kendisine iletildi ve maksimum maaş seti P AY r
ben onlardan.
(b) Henüz tüm B 0 , ..., B r − 1'i görmediysem , P AY r
i = ∅ .
(c) Sonra, "aday bloğunu" B r hesaplıyorum
i = (r, P AY r
i , SIG ben (Q r − 1), H (B r − 1)).
(c) Son olarak, m mesajını hesaplıyorum
r, 1
ben
= (B r
ben , esig ben (H (B r
i)), σ
r, 1
i), geçici olanını yok eder

gizli anahtar sk
r, 1
i ve sonra iki mesajı yayar, m
r, 1
ben
ve (SIG i (Q r − 1), σ
r, 1
i),
ayrı ayrı ama aynı anda.a
a i lider olduğunda, SIG i (Q r − 1) başkalarının Q r = H (SIG i (Q r − 1), r) hesaplamasına izin verir .

Sayfa 61

Seçici Yayılma
Adım 1'in küresel uygulamasını ve tüm turu kısaltmak için, (r, 1) -
mesajlar seçilerek yayılır. Yani, sistemdeki her j kullanıcısı için,
• Aldığı ve başarıyla doğruladığı ilk (r, 1) -message için,bir içerip içermediği
bir blok veya sadece bir kimlik bilgisi ve Q r-1 imzası ise , j oyuncusu bunu her zamanki gibi yayar.
• J oyuncusunun aldığı ve başarıyla doğruladığı tüm diğer (r, 1) -mesajlar için,
sadece içerdiği kimlik bilgisinin karma değeri, karma değerler arasında en küçükse
Aldığı ve başarıyla doğruladığı tüm (r, 1) mesajlarında bulunan kimlik bilgilerinin
Irak.
• Ancak, j, m biçiminde iki farklı mesaj alırsa
r, 1
ben
aynı oyuncudan ben,b o
i'nin kimlik bilgisinin karma değeri ne olursa olsun ikincisini atar.
Seçici yayılma altında, her bir potansiyel liderin kendi
kimlik bilgisi σ
r, 1
ben
m'den ayrı
r, 1
i :c bu küçük mesajlar bloklardan daha hızlı seyahat eder,
m'nin zamanında yayılması
r, 1
i içerdiği kimlik küçük karma değerlerini sahip olduğu 'iken, s
büyük hash değerlerine sahip olanların hızla kaybolmasını sağlayın.
a Yani tüm imzalar doğrudur ve eğer m r biçimindeyse , 1
ben
hem blok hem de hash geçerlidir
- Bununla birlikte j, dahil edilen ödeme setinin i için maksimum olup olmadığını kontrol etmez.
b Bu da kötü niyetli olduğum anlamına gelir.
c Bunu önerdiği için Georgios Vlachos'a teşekkür ederiz.

Sayfa 62

Adım 2: Dereceli Konsensüs Protokolü GC'nin İlk Adımı
Her kullanıcı için talimatlar i ∈ PK r − k : Kullanıcı i, sahip olduğu anda kendi r turunun 2. Adımını başlatır.
CERT r − 1 .
• Kullanıcı i maksimum süre bekler t 2
λ + Λ. Beklerken aşağıdaki gibi davranırım.
1. 2λ süresini bekledikten sonra, H (σ

r, 1
ℓ
) ≤ H (σ
r, 1
j) hepsi için
kimlik bilgileri σ
r, 1
j
başarıyla doğrulanmış (r, 1) - aldığı mesajların bir parçası olan
şimdiye kadar.a
2. H (B r − 1) hash değeriyle eşleşen bir B r − 1 bloğu almışsa
CERT r − 1'de bulunan ,b ve eğer from'dan geçerli bir mesaj almışsa m
r, 1
ℓ
=
(B r
ℓ , esig ℓ (H (B r
ℓ)), σ
r, 1
ℓ
),c sonra beklemeyi bırakıp v ′ ayarlar
ben
(H (B r
ℓ), ℓ).
3. Aksi takdirde, t 2 süresi dolduğunda, v 'yi
ben
⊥ .
4. v ′ değeri
i ayarlandı, CERT r − 1'den Q r − 1'i hesaplar ve
i ∈ SV r, 2 veya değil.
5. i ∈ SV r, 2 ise , i m mesajını hesaplar
r, 2
ben
(ESIG i (v ′
i), σ
r, 2
i),d onun geçici olduğunu yok eder
gizli anahtar sk
r, 2
i ve sonra m'yi yayar
r, 2
i . Aksi takdirde yayılmadan dururum
herhangi bir şey.
a Esasen, i kullanıcısı, r turunun liderinin kullanıcı ℓ olduğuna özel olarak karar verir.
b Elbette, eğer CERT r − 1 , B r − 1 = B r − 1 olduğunu gösterirse
ϙ
, o zaman B r − 1'i aldığı anda zaten "aldım"
CERT r − 1 .
c Yine, oyuncunun imzaları ve karmalarının tümü başarıyla doğrulandı ve P AY r
ℓ
B r'de
ℓ geçerli bir maaş setidir
round r - i, P AY r olup olmadığını kontrol etmese de
ℓ , ℓ için maksimaldir veya değildir. Eğer B r
ℓ boş bir maaş seti içeriyorsa

i B görmek gerek aslında yoktur r-1 B olmadığını doğrulayarak önce r
ℓ geçerli veya değil.
d Mesaj m r, 2
ben
i oyuncunun v'nin ilk bileşenini düşündüğüne işaret eder ′
i sonraki bloğun karması olmak veya
sonraki bloğun boş olduğunu düşünür.
62

Sayfa 63

Adım 3: GC'nin İkinci Adımı
Her kullanıcı için talimatlar i ∈ PK r − k : Kullanıcı i, sahip olduğu anda kendi r turunun 3. Adımını başlatır.
CERT r − 1 .
• Kullanıcı i maksimum süre bekler t 3
t 2 + 2λ = 3λ + Λ. Beklerken, gibi davranıyorum
takip eder.
1. En az t H geçerli mesaj alacak şekilde bir v değeri varsa m
r, 2
j
nın-nin
form (ESIG j (v), σ
r, 2
j) herhangi bir çelişki olmaksızın,a sonra beklemeyi bırakır ve ayarlar
v ′ = v.
2. Aksi takdirde, t 3 zamanı bittiğinde, v ′ = ⊥ olur.
3. v ′ değeri ayarlandığında, i CERT r − 1'den Q r − 1'i hesaplar ve
i ∈ SV r, 3 veya değil.
4. Eğer i ∈ SV r, 3 ise , o zaman m mesajını hesaplar
r, 3
ben
(ESIG i (v ′), σ
r, 3
i), onu yok eder
geçici gizli anahtar sk
r, 3
i ve sonra m'yi yayar
r, 3
i . Aksi takdirde, olmadan dururum
her şeyi yaymak.
a Yani, sırasıyla ESIG j (v) ve farklı bir ESIG j (v) içeren iki geçerli mesaj almamışsa ,
bir oyuncudan j. Dürüst bir oyuncu
belirli bir biçimde mesajlar istiyorsa, birbiriyle çelişen mesajlar asla sayılmaz veya geçerli sayılmaz.
63

Sayfa 64

Adım 4: GC'nin Çıktısı ve BBA'nın İlk Adımı ⋆
Her kullanıcı için talimatlar i ∈ PK r − k : Kullanıcı i, kendi r turunun 4. Adımını başlatır.
kendi 3. Adımını bitirir.
• Kullanıcı i maksimum süre 2λ bekler.a Beklerken aşağıdaki gibi davranırım.
1. GC'nin çıktısı olan v i ve g i'yi aşağıdaki gibi hesaplar .
(a) En az t H geçerli mesaj alacak şekilde bir v ′ = ⊥ değeri varsa
m
r, 3

j
= (ESIG j (v ′), σ
r, 3
j), sonra beklemeyi bırakır ve v i
v ′ ve g i
2.
(b) En az t H geçerli mesaj almışsa m
r, 3
j
= (ESIG j (⊥), σ
r, 3
j), sonra durur
bekliyor ve ayarlar v i
⊥ ve g i
0.b
(c) Aksi takdirde, 2λ süresi bittiğinde, bir v ′ = ⊥ değeri varsa ,
en az ⌈ t H alındı
2
⌉ geçerli mesajlar m
r, j
j
= (ESIG j (v ′), σ
r, 3
j), sonra v i'yi ayarlar
v ′
ve g ben
1.c
(d) Aksi takdirde, 2λ süresi bittiğinde, v i
⊥ ve g i
0.
2. v i ve g i değerleri ayarlandığında, i aşağıdaki gibi BBA ⋆ girdisi olan b i'yi hesaplar :
b ben
0 eğer g ben = 2 ve b ben
Aksi takdirde 1.
3. i CERT r − 1'den Q r − 1'i hesaplar ve i ∈ SV r, 4 olup olmadığını kontrol eder .
4. Eğer i ∈ SV r, 4 ise , m mesajını hesaplar
r, 4
ben
(ESIG i (b i), ESIG i (v i), σ
r, 4
i), onu yok eder
geçici gizli anahtar sk
r, 4
i ve m'yi yayar
r, 4
i . Aksi takdirde yayılmadan dururum
herhangi bir şey.
a Dolayısıyla, r turunun 1. Adımına başladığımdan beri maksimum toplam süre t 4 olabilir.
t 3 + 2λ = 5λ + Λ.
b Adım (b) 'nin protokolde olup olmaması onun doğruluğunu etkilemez. Ancak, Adım (b) 'nin varlığı
Yeterince çok sayıda 3. Adım doğrulayıcısının "⊥ imzalaması" durumunda 4. Adımın 2λ'dan daha
kısa sürede bitmesini sağlar.
c Bu durumda, varsa , v ′ ' nin benzersiz olması gerektiği kanıtlanabilir .

Adım s, 5 ≤ s ≤ m + 2, s - 2 ≡ 0 mod 3: BBA'nın 0'a Sabit Bir Madeni Para Adımı ⋆
Her kullanıcı için talimatlar i ∈ PK r − k : Kullanıcı i, kendi r aşamasını başlatır.
kendi Adım s - 1'i bitirir.
• Kullanıcı i maksimum süre 2λ bekler.a Beklerken aşağıdaki gibi davranırım.
- Bitiş Koşulu 0: Herhangi bir noktada bir v = ⊥ dizisi ve bir s ′ adımı varsa, öyle ki
(a) 5 ≤ s ′ ≤ s, s ′ - 2 ≡ 0 mod 3 — yani, Adım s ′ , Madeni Para 0'a Sabit bir adımdır,
(b) en az t H geçerli mesaj aldım m
r, s ′ −1
j
= (ESIG j (0), ESIG j (v), σ
r, s ′ −1
j
),b
ve
(c) i geçerli bir mesaj aldı (SIG j (Q r − 1), σ
r, 1
j) j ikinci olmak üzere
v bileşeni,
daha sonra, beklemeyi bırakırım ve Adım s'yi kendi yürütmesini bitiririm (ve aslında r turu)
herhangi bir şeyi a (r, s) -verifier olarak yaymadan hemen; H (B r) 'yi ilk olarak ayarlar
v bileşeni; ve kendi CERT setleri r mesajları m kümesi olmak
r, s ′ −1
j
(b) adımının
(SIG j (Q r − 1), σ ile birlikte
r, 1
j).c
- Durum 1 Bitiş: herhangi bir noktada var ise adım s ' şekildedir
(a ') 6 ≤ s ′ ≤ s, s ′ - 2 ≡ 1 mod 3 — yani, Adım s ′ bir Madeni Para 1'e Sabit adımdır ve
(b ') en az t H geçerli mesaj aldım m
r, s ′ −1
j
= (ESIG j (1), ESIG j (v j),
σ
r, s ′ −1
j
),d
daha sonra, beklemeyi bırakıyorum ve Adım s'yi (ve aslında r turunun) kendi yürütmesini bitiriyorum.
herhangi bir şeyi a (r, s) -verifier olarak yaymadan uzaklaştırmak; ayarlar B r = B r
ǫ ; ve kendisininkini ayarlar
CERT r , mesaj seti olacak m
r, s ′ −1
j
alt adımın (b ').
- Herhangi bir noktada en az t H almışsa
geçerli m
r, s − 1
j
formun
(ESIG j (1), ESIG j (v j), σ
r, s − 1
j
), sonra beklemeyi bırakır ve b i ayarlar.
1.

- Herhangi bir noktada en az t H almışsa
geçerli m
r, s − 1
j
formun
(ESIG j (0), ESIG j (v j), σ
r, s − 1
j
), ancak aynı v üzerinde anlaşmazlar, sonra durur
bekliyor ve b i ayarlar
0.
- Aksi takdirde, 2λ süresi bittiğinde, b i
0.
- b i değeri ayarlandığında, i CERT r − 1'den Q r − 1'i hesaplar ve
i ∈ SV r, s .
- eğer i ∈ SV r, s , i mesaj m'yi hesaplar
r, s
ben
(ESIG i (b i), ESIG i (v i), σ
r, s
i) v ben olmak
4. Adımda hesapladığı değer, geçici gizli anahtar skalasını yok eder
r, s
ben ve sonra
m'yi yayar
r, s
i . Aksi takdirde, hiçbir şey yaymadan dururum.
a Dolayısıyla, r turunun 1. Adımına başladığımdan beri maksimum toplam süre t s olabilir.
t s − 1 + 2λ =
(2s - 3) λ + Λ.
b Oyuncu j'den gelen böyle bir mesaj, i oyuncusu 1'e imza atan j'den bir mesaj almış olsa bile sayılır.
Bitiş Koşulu 1 için de benzer şeyler. Analizde gösterildiği gibi, bu, tüm dürüst kullanıcıların
Cert r birbirinden zaman X içinde.
c Kullanıcı i artık H (B r) 'yi ve kendi raund r bitişlerini biliyor . O sadece aslında blok B kadar
beklemek gerekiyor r olan
ona yayıldı, bu biraz daha zaman alabilir. Hala genel bir kullanıcı olarak mesajların yayılmasına
yardımcı oluyor.
ancak bir (r, s) -verifiyeri olarak herhangi bir yayılma başlatmaz. Özellikle, tüm mesajların
protokolümüz için yeterli olan CERT r . Ayrıca b i ayarlaması gerektiğini unutmayın
İkili BA protokolü için 0, ancak
b i zaten bu durumda gerekli değildir. Gelecekteki tüm talimatlar için benzer şeyler.
d Bu durumda, v j'lerin ne olduğu önemli değildir .

Sayfa 66

Aşama s, 6 ≤ s ≤ m + 2, s - 2 ≡ 1 mod 3: BBA A Düğme Sabit için-1 Aşama ⋆
Her kullanıcı için talimatlar i ∈ PK r − k : Kullanıcı i, kendi r aşamasını başlatır.
kendi Adım s - 1'i bitirir.
• Kullanıcı i maksimum süre 2λ bekler. Beklerken aşağıdaki gibi davranırım.
- Bitiş Koşulu 0: Madeni Para 0'a Sabit adımındaki talimatların aynısı.
- Bitiş Koşulu 1: Madeni Para 0'a Sabit adımındaki talimatların aynısı.
- Herhangi bir noktada en az t H almışsa
geçerli m
r, s − 1
j

formun
(ESIG j (0), ESIG j (v j), σ
r, s − 1
j
), sonra beklemeyi bırakır ve b i ayarlar.
0.a
- Aksi takdirde, 2λ süresi bittiğinde, b i
1.
- b i değeri ayarlandığında, i CERT r − 1'den Q r − 1'i hesaplar ve
i ∈ SV r, s .
- eğer i ∈ SV r, s , i mesaj m'yi hesaplar
r, s
ben
(ESIG i (b i), ESIG i (v i), σ
r, s
i) v ben olmak
4. Adımda hesapladığı değer, geçici gizli anahtar skalasını yok eder
r, s
ben ve sonra
m'yi yayar
r, s
i . Aksi takdirde, hiçbir şey yaymadan dururum.
Bir alıcı t o Not * H geçerli (R, S - 1) Durum 1 Bitiş anlamına gelir 1 için oturum -İletileri.
Adım s, 7 ≤ s ≤ m + 2, s - 2 ≡ 2 mod 3: BBA'nın Gerçekten Ters Çevrilmiş Bir Madeni Para Adımı ⋆
Her kullanıcı için talimatlar i ∈ PK r − k : Kullanıcı i, kendi r aşamasını başlatır.
kendi s - 1 adımını bitirir.
• Kullanıcı i maksimum süre 2λ bekler. Beklerken aşağıdaki gibi davranırım.
- Bitiş Koşulu 0: Madeni Para 0'a Sabit adımındaki talimatların aynısı.
- Bitiş Koşulu 1: Madeni Para 0'a Sabit adımındaki talimatların aynısı.
- Herhangi bir noktada en az t H almışsa
geçerli m
r, s − 1
j
formun
(ESIG j (0), ESIG j (v j), σ
r, s − 1
j
), sonra beklemeyi bırakır ve b i ayarlar.
0.
- Herhangi bir noktada en az t H almışsa
geçerli m
r, s − 1
j
formun
(ESIG j (1), ESIG j (v j), σ
r, s − 1
j
), sonra beklemeyi bırakır ve b i ayarlar.
1.
- Aksi takdirde, 2λ süresi bittiğinde, SV
r, s − 1
ben
aşağıdaki (r, s - 1) -verifiers kümesi
kime geçerli bir mesaj almış m
r, s − 1

j
İ b setleri i
lsb (min
j∈ SV r, s − 1
ben
H (σ
r, s − 1
j
)).
- b i değeri ayarlandığında, i CERT r − 1'den Q r − 1'i hesaplar ve
i ∈ SV r, s .
- eğer i ∈ SV r, s , i mesaj m'yi hesaplar
r, s
ben
(ESIG i (b i), ESIG i (v i), σ
r, s
i) v ben olmak
4. Adımda hesapladığı değer, geçici gizli anahtar skalasını yok eder
r, s
ben ve sonra
m'yi yayar
r, s
i . Aksi takdirde, hiçbir şey yaymadan dururum.
Açıklama. Prensip olarak, alt bölüm 6 .2'de ele alındığı gibi , protokol keyfi olarak birçok
bir turda adımlar. Bu durumda, tartışıldığı gibi, s> μ olan bir i ∈ SV r, s kullanıcısı tükenmiştir
66

Sayfa 67

önceden oluşturulmuş geçici anahtarların zulası ve (r, s) -mesajının kimliğini doğrulaması gerekir.
r, s
ben
tarafından
Geçici anahtarların “basamaklandırılması”. Böylece i'nin mesajı biraz uzar ve bunları daha uzun süre iletir
mesajlar biraz daha zaman alacak. Buna göre, belirli bir turun pek çok adımından sonra,
λ parametresi otomatik olarak biraz artacaktır. (Ama yeni bir kez orijinal λ'ya geri döner.
blok üretilir ve yeni bir tur başlar.)
Round-r Bloğunun Doğrulayıcı Olmayanlar Tarafından Yeniden İnşası
Sistemdeki her i kullanıcısı için talimatlar: Kullanıcı i, sahip olduğu anda kendi turunu başlatır.
CERT r − 1 .
• Protokolün her adımının talimatlarını izler, tüm
mesajlar, ancak içinde doğrulayıcı değilse bir adımda herhangi bir yayılma başlatmaz.
• i kendi r turunu bazılarında Bitiş Koşulu 0 veya Bitiş Koşulu 1 girerek bitirir.
adım, karşılık gelen CERT ile r .
• Bundan sonra, gerçek B r bloğunu almayı beklerken r + 1 turuna başlar (
hash'i H (B r) CERT r tarafından sabitlenmiş olan onu zaten aldı) . Yine, eğer
CERT r , B r = B r olduğunu gösterir
ϱ , B r'yi CERT r'ye sahip olduğu anda bilir .
6.4 Algorand Analizi ′
2
Algorand ′ analizi
2 Algorand'ınkinden kolayca türetilir ′
1 . Esasen, Algorand'da ′
2 , ile
ezici bir olasılık, (a) tüm dürüst kullanıcılar aynı blok B r üzerinde hemfikir ; yeninin lideri

blok dürüsttür ve olasılıkla en az p h = h 2 (1 + h - h 2).
7 Çevrimdışı Dürüst kullanıcıları yönetme
Dediğimiz gibi, dürüst bir kullanıcı, çevrimiçi olma dahil olmak üzere tüm talimatlarını takip eder.
ve protokolü çalıştırmak. Algorand'da hesaplama ve hesaplamadan bu yana bu büyük bir yük değildir.
Dürüst bir kullanıcının ihtiyaç duyduğu bant genişliği oldukça mütevazı. Yine de, Algorand'in
Dürüst kullanıcıların çevrimdışı olmasına izin verilen iki modelde çalışacak şekilde kolayca değiştirilebilir
harika numaralar.
Bu iki modeli tartışmadan önce, dürüst oyuncuların yüzdesi
% 95 idi, Algorand hala h =% 80 olduğu varsayılarak tüm parametreleri ayarlayarak çalıştırılabilirdi.
Buna göre, Algorand dürüst oyuncuların en fazla yarısı bile olsa düzgün bir şekilde çalışmaya devam edecektir.
çevrimdışı olmayı seçti (aslında, büyük bir "devamsızlık" durumu). Aslında, herhangi bir zamanda, en azından
Çevrimiçi oyuncuların% 80'i dürüst olacaktır.
Sürekli Katılımdan Tembel Dürüstlüğe Gördüğümüz gibi, Algorand ′
1 ve Algorand ′
2 seçim
yeniden inceleme parametresi k. Şimdi k'yi uygun şekilde büyük seçmenin birinin kaldırılmasını sağladığını gösterelim.
Sürekli Katılım şartı. Bu gereklilik, önemli bir özelliği garanti eder: yani,
temeldeki BA protokolü BBA ⋆ uygun bir dürüst çoğunluğa sahiptir. Şimdi ne kadar tembel olduğunu açıklayalım
dürüstlük, bu mülkü tatmin etmek için alternatif ve çekici bir yol sağlar.
67

Sayfa 68

Bir kullanıcının tembel ama dürüst olduğunu hatırlayın, eğer (1) tüm talimatlarını yerine getirirse, protokole katılması istenir ve (2) sadece protokole katılması istenir
çok nadiren - örneğin, haftada bir - uygun bir ön bildirimle ve potansiyel olarak önemli miktarda
Katıldığı zaman ödüller.
Algorand'in bu tür oyuncularla çalışmasına izin vermek için, sadece "
Çok daha erken bir turda zaten sistemde bulunan kullanıcılar arasında mevcut tur. " Gerçekten, hatırla şunu
bir r turu için doğrulayıcılar r - k turundaki kullanıcılardan seçilir ve seçimler temel alınarak yapılır.
Q r − 1 miktarında . Haftanın yaklaşık 10.000 dakikadan oluştuğunu unutmayın ve bir
Tur kabaca (örneğin ortalama) 5 dakika sürer, bu nedenle haftada kabaca 2.000 tur vardır. Varsaymak
Bir noktada, zamanını planlamak ve olup olmayacağını bilmek istediğim bir kullanıcı
önümüzdeki hafta bir doğrulayıcı. Protokol artık bir tur için doğrulayıcıları seçiyor.
r - k - 2.000 tur ve seçimler Q r − 2.001'e dayanmaktadır . R turunda, tanıdığım oyuncu
aslında blok zincirinin parçası oldukları için Q r − 2.000 , ..., Q r − 1 değerleri. Sonra, her bir M için
1 ile 2.000 arasında, i, ancak ve ancak
.H (SIG i (r + M, s, Q r + M − 2,001)) ≤ s.
Bu nedenle, sonraki 2000 turda doğrulayıcı olarak çağrılıp çağrılmayacağını kontrol etmek için, hesaplamak σ
Hanım
ben
= SIG i (r + M, s, Q r + M − 2.001) M = 1 ila 2.000 ve her adım s için ve kontrol edin
olsun .H (σ
Hanım
ben
) ≤ p bazıları için. Dijital imza hesaplamak bir milisaniye sürüyorsa,
Bu işlemin tamamı onu yaklaşık 1 dakikalık hesaplama sürecini alacak. Doğrulayıcı olarak seçilmediyse

Bu turların herhangi birinde, o zaman "dürüst bir vicdan" ile çevrimdışı duruma geçebilir. Sürekli olsaydı
katılırsa, sonraki 2.000 raundda aslında 0 adım atmış olacaktı! Bunun yerine,
bu turlardan birinde doğrulayıcı olarak seçilir, ardından kendini hazırlar (örneğin, tüm
gerekli bilgi) uygun turda dürüst bir doğrulayıcı olarak hareket etmek.
Böyle davranarak, tembel ama dürüst bir potansiyel doğrulayıcı sadece yayılmaya katılmayı özlüyorum
mesajların. Ancak mesaj yayılımı tipik olarak sağlamdır. Dahası, ödeyenler ve alacaklılar
yakın zamanda yayılmış ödemelerin, ödemelerine ne olduğunu izlemek için çevrimiçi olması bekleniyor,
ve böylece dürüst olurlarsa mesajın yayılmasına katılırlar.
Paranın Dürüst Çoğunluğu ile 8 Protokol Algorand ′
Son olarak, Kullanıcıların Dürüst Çoğunluğu varsayımını çok daha fazlasıyla nasıl değiştireceğimizi gösteriyoruz.
Paranın anlamlı Dürüst Çoğunluğu varsayımı. Temel fikir şudur (bir teminat kanıtı çeşidinde)
"İle orantılı bir ağırlığa (yani karar gücü) sahip SV r, s'ye ait bir kullanıcı i PK r − k seçmek için
i. 'nin sahip olduğu para miktarı. "[24]
HMM varsayımımıza göre, bu miktarın r −k turunda sahiplenilip sahiplenilmeyeceğini seçebiliriz
veya (başlangıcında) r turunda. Sürekli katılımı önemsemediğimizi varsayarak,
ikinci seçenek. (Sürekli katılımı kaldırmak için eski seçimi tercih ederdik.
Daha iyisi, r - k - 2.000 turunda sahip olunan para miktarı için.)
Bu fikri uygulamanın birçok yolu var. En basit yol, her bir tuşa sahip olmaktır.
en fazla 1 birim para ve sonra rasgele n kullanıcı i PK r − k arasından seçin, öyle ki a
(r)
ben
= 1.
24 Sürekli katılımın yerini alması için PK r − k − 2.000 demeliyiz . Basit olması için, bir kişi gerektirmek isteyebileceğinden
her halükarda sürekli katılım, bir daha az parametre taşımak için PK r − k'yi eskisi gibi kullanıyoruz.

Sayfa 69

Sonraki En Basit Uygulama
Bir sonraki en basit uygulama, her bir genel anahtarın maksimum bir miktara sahip olmasını talep etmek olabilir.
M, bazı sabit M için M değeri, toplam miktarına kıyasla yeterince küçüktür.
sistemdeki para, öyle ki bir anahtarın birden fazla doğrulayıcı kümesine ait olma olasılığı
adım adım —söy— k turları ihmal edilebilir. Sonra, bir miktar paraya sahip olan bir anahtar i ∈ PK r − k
(r)
ben
r turunda, SV r'ye ait olarak seçilir , eğer
.H (SIG i (r, s, Q r − 1)) ≤ p · a (r)
ben
M
.
Ve hepsi eskisi gibi ilerliyor.
Daha Karmaşık Bir Uygulama
Son uygulama “sistemdeki zengin bir katılımcıyı birçok anahtara sahip olmaya zorladı”.
Aşağıda açıklanan alternatif bir uygulama, statü kavramını genelleştirir ve
her bir kullanıcı i, her biri bağımsız olarak doğrulayıcı olarak seçilen K + 1 kopyalardan (i, v) oluşacak,
ve kendi geçici anahtarına sahip olacak (pk
r, s
ben, v , sk

r, s
i, v) r turunun s adımında. K değeri bağlıdır
para miktarına göre a
(r)
ben
r turunda i'ye ait.
Şimdi böyle bir sistemin nasıl çalıştığını daha ayrıntılı olarak görelim.
Kopya Sayısı Her bir doğrulayıcı kümesinin hedeflenen beklenen önceliği n olsun ve bir
(r)
ben
r turunda i kullanıcısı tarafından sahip olunan para miktarı. A r sahip olunan toplam para miktarı olsun
PK r − k'deki kullanıcılar tarafından r turunda, yani,
Bir r = ∑
i∈ PK r − k
a
(r)
i .
Eğer i PK r − k'de bir kullanıcı ise , i'nin kopyaları (i, 1), ..., (i, K + 1), burada
K = ⌊
n · a
(r)
ben
A r
⌋ .
Misal. O zaman n = 1.000, bir R = 10 9 , ve
(r)
ben
= 3,7 milyon. Sonra,
K = ⌊
· 10 3 · (3,7 · 10 6)
10 9
⌋ = ⌊ 3.7⌋ = 3.
Doğrulayıcılar ve Kimlik Bilgileri Bırakın PK r − k'de K + 1 kopyaları olan bir kullanıcı olalım .
Her v = 1, ..., K için copy (i, v) otomatik olarak SV r, s'ye aittir . Yani, kimlik bilgisi
σ
r, s
i, v
SIG i ((i, v), r, s, Q r − 1), ancak karşılık gelen koşul olur .H (σ
r, s
i, v) ≤ 1, ki
herzaman doğru.
Kopya için (i, K + 1), r turunun her s Adımında,
.H (SIG i ((i, K + 1), r, s, Q r − 1)) ≤ bir (r)
ben
n
A r
- K.

Sayfa 70

Eğer öyleyse, kopya (i, K + 1) SV r, s'ye aittir . Kanıtlamak için kimlik bilgilerini yayıyorum
σ
r, 1
i, K + 1
= SIG ben ((i, K + 1), r, s, Q r − 1).

Misal. Önceki örnekte olduğu gibi, n = 1K olsun, a
(r)
ben
= 3,7M, A r = 1B ve b'de 4
kopyalar: (i, 1), ..., (i, 4). Daha sonra ilk 3 kopya otomatik olarak SV r'ye aittir . Dördüncüsü için, kavramsal olarak, Algorand ′ , Tura olasılığı 0.7 olan taraflı bir parayı bağımsız olarak yuvarlar. Kopyala
(i, 4), ancak ve ancak yazı tura atılması durumunda seçilir.
(Elbette, bu önyargılı yazı tura atma, karma oluşturma, imzalama ve karşılaştırma yoluyla gerçekleştirilir.
onun sonucunu kanıtlamam için bu yazıda başından beri yaptım.)
Olağan Olarak İş Doğrulayıcıların nasıl seçildiğini ve kimlik bilgilerinin nasıl olduğunu açıklamak bir r turunun her adımında hesaplandığında, bir turun yürütülmesi daha önce açıklanana benzerdir.
9 Taşıma Çatalları
Çatal olasılığını 10 −12 veya 10 −18'e düşürdükten sonra, elleçlemek neredeyse gereksizdir onları meydana gelme ihtimalleri çok uzaktır. Algorand, ancak, çeşitli çatallar da kullanabilir. iş kanıtı olsun veya olmasın çözüm prosedürleri.
Kullanıcılara çatalları çözme talimatı vermenin olası bir yolu şudur:
• Bir kullanıcı birden fazla zincir görürse en uzun zinciri takip edin.
• Birden fazla en uzun zincir varsa, sonunda boş olmayan bloğu olanı takip edin. Eğer hepsinin sonunda boş bloklar var, ikinci-son bloklarını düşünün.
• Sonunda boş olmayan bloklara sahip birden fazla en uzun zincir varsa, diyelim ki zincirler r uzunluğunda, r bloğunun lideri en küçük kimlik bilgisine sahip olanı takip edin. Eğer bağ varsa, r bloğunun kendisi en küçük hash değerine sahip olanı takip edin. Hala bağlar varsa, takip edin r bloğu sözlükbilimsel olarak birinci sırada olan blok.
10 Ağ Bölümlerini Kullanma
Söylediği gibi, mesajların ağdaki tüm kullanıcılar arasında yayılma sürelerinin daha yüksek olduğunu varsayıyoruz.
λ ve Λ ile sınırlıdır. Günümüz İnternet'i hızlı ve sağlam olduğundan bu güçlü bir varsayım değildir ve bu parametrelerin gerçek değerleri oldukça makul. Burada, bize bu Algorand işaret izin '
2
İnternet ara sıra ikiye bölünse bile çalışmaya devam eder. Durum ne zaman
İnternet ikiden fazla bölüme ayrılmışsa benzerdir.
10.1 Fiziksel Bölümler
Her şeyden önce, bölüm fiziksel nedenlerden kaynaklanıyor olabilir. Örneğin, büyük bir deprem olabilir
Avrupa ile Amerika arasındaki bağın tamamen kopmasıyla sonuçlanır. Bu durumda
kötü niyetli kullanıcılar da bölümlenir ve iki bölüm arasında iletişim yoktur. Böylece

---

Sayfa 71

biri 1. bölüm ve diğeri 2. bölüm için olmak üzere iki Düşman olacak.
protokolü kendi bölümünde bozmak.
Bölmenin r turunun ortasında gerçekleştiğini varsayın. Daha sonra her kullanıcı hala bir Doğrulayıcı , önceki ile aynı olasılıkla, PK r − k'ye dayanır . HSV edelim
r, s
ben
ve MSV
r, s
ben
sırasıyla
i ∈ {1,2} bölümündeki bir adımda dürüst ve kötü niyetli doğrulayıcılar olun. Sahibiz
| HSV
r, s
1 | + | MSV

r, s
1 | + | HSV
r, s
2 | + | MSV
r, s
2 | = | HSV r, s | + | MSV r, s |.
| HSV r, s | + | MSV r, s | <| HSV r, s | + 2 | MSV r, s | Çok büyük olasılıkla <2t H.
Bir parçam varsa | HSV
r, s
ben
| + | MSV
r, s
ben
| ≥ t H ihmal edilemez olasılıkla, örneğin% 1, sonra
olasılık | HSV
r, s
3 − i | + | MSV
r, s
3 − i | ≥ t H çok düşüktür, örneğin F = 10 −18 olduğunda 10 −16 . Bu durumda,
daha küçük kısmı çevrimdışı olarak ele alabiliriz, çünkü içinde yeterince doğrulayıcı olmayacaktır.
t Bir bloğu onaylamak için H imzaları oluşturmak için bu bölüm .
Büyük bölümü, diyelim ki 1. bölümü genelliği kaybetmeden ele alalım. | HSV r, s | <
t H , her adımda ihmal edilebilir olasılıkla s, ağ bölümlendiğinde, | HSV
r, s
1 | olabilir
daha t daha H bazı göz ardı edilemeyecek bir olasılık ile. Bu durumda, Düşman, bazılarıyla
diğer ihmal edilemez olasılık, ikili BA protokolünü r turunda bir çatala zorla
boş blok B r ve boş blok B r
ọ her ikisi de t H geçerli imzaya sahip.25 Örneğin,
Coin-Fixed-To-0 adımı, HSV'deki tüm doğrulayıcılar
r, s
1
bit 0 ve H (B r) için imzalandı ve bunların
mesajlar. MSV'deki tüm doğrulayıcılar
r, s
1
ayrıca 0 ve H (B r) imzaladı , ancak mesajlarını sakladı. Çünkü
| HSV
r, s
1 | + | MSV
r, s
1 | ≥ t H , sistem B r'yi onaylamak için yeterli imzaya sahiptir . Ancak,
kötü niyetli doğrulayıcılar imzalarını sakladılar, kullanıcılar bir Coin-Fixed-To- olan s + 1 adımına
girerler.
1 adım. Çünkü | HSV
r, s
1 | <t H bölüm nedeniyle, HSV'deki doğrulayıcılar
r, s + 1
1
t H görmedim
bit 0 için imzalar ve hepsi bit 1 için imzalandı. MSV'deki tüm doğrulayıcılar
r, s + 1
1
Aynı şeyi yaptı. Çünkü
| HSV

r, s + 1
1
| + | MSV
r, s + 1
1
| ≥ t H , sistem B r'yi onaylamak için yeterli imzaya sahip
ϱ . Düşman
daha sonra MSV'nin imzalarını serbest bırakarak bir çatal oluşturur
r, s
1
0 ve H (B r) için.
Buna göre, karşılık gelen r yuvarlak blokları tarafından tanımlanan iki Q r ' olacaktır . Ancak,
çatal devam etmez ve r + 1 turunda iki daldan sadece biri büyüyebilir.
Algorand için Ek Talimatlar ′
2
. Boş olmayan bir B r bloğu ve boş
blok B r
ϱ , boş olmayan olanı (ve onun tarafından tanımlanan Q r'yi) takip edin.
Gerçekten de, kullanıcılara protokoldeki boş olmayan bloğun büyük olması durumunda gitmeleri
talimatını vererek
PK r + 1 − k'deki dürüst kullanıcıların miktarı r + 1 turunun başında bir çatal olduğunu fark eder, sonra
boş blok yeterli takipçiye sahip olmayacak ve büyümeyecek. Düşmanın başardığını varsayın
Dürüst kullanıcıları, bazı dürüst kullanıcıların B r (ve belki de B r
ϱ) ve bazıları yalnızca
B r
ϱ . Çünkü rakip, hangisinin B r'yi takiben doğrulayıcı olacağını ve hangilerinin
B r'yi takiben bir doğrulayıcı olacak
ϱ , dürüst kullanıcılar rastgele bölümlenir ve her biri hala
doğrulayıcı olur (B r'ye göre veya B r'ye göre)
ϱ) s> 1 adımında olasılıkla
s. Kötü niyetli kullanıcılar için, her birinin doğrulayıcı olmak için iki şansı olabilir.
B r ve diğeri B r ile
ϱ , her biri bağımsız olarak p olasılığa sahiptir.
HSV edelim
r + 1, s
1; B r
B r'yi izleyen r + 1 turunun s adımlarında dürüst doğrulayıcılar kümesi olun . Diğer gösterimler
HSV gibi
r + 1, s
1; B r
ϱ
, MSV
r + 1, s
1; B r
ve MSV
r + 1, s
1; B r
ϱ
benzer şekilde tanımlanmıştır. Chernoff'a bağlı olarak, bu çok kolay
25 Boş olmayan iki bloğa sahip bir çatala sahip olmak, önemsiz durumlar dışında, bölmeli veya
bölmesiz mümkün değildir.
olasılık.
71

---

bunu çok büyük bir olasılıkla görmek için
| HSV
r + 1, s
1; B r
| + | HSV
r + 1, s
1; B r
ϱ
| + | MSV
r + 1, s
1; B r
| + | MSV
r + 1, s
1; B r
ϱ
| <2t , H .
Buna göre, iki dalın her ikisinin de yuvarlak için bir bloğu onaylayan H uygun imzaları olamaz.
r + 1 aynı adımda s. Dahası, iki adım s ve s ′ için seçim olasılıkları ,
aynı ve seçimler bağımsızdır, aynı zamanda ezici bir olasılıkla
| HSV
r + 1, s
1; B r
| + | MSV
r + 1, s
1; B r
| + | HSV
r + 1, s ′
1; B r
ϱ
| + | MSV
r + 1, s ′
1; B r
ϱ
| <2t H ,
herhangi iki adım için s ve s ′ . F = 10 −18 olduğunda , Düşman yapamadığı sürece sendika sınırına göre
Dürüst kullanıcıları uzun süre bölümlere ayır (10 4 adım diyelim ki bu 55 saatten fazla λ = 10 saniye26), yüksek olasılıkla (1−10 −10 diyelim) en fazla bir dalın t H uygun imzası olacaktır.
r + 1 turundaki bir bloğu onaylamak için.
Son olarak, fiziksel bölüm kabaca aynı boyutta iki parça oluşturduysa,
olasılık | HSV
r, s
ben
| + | MSV
r, s
ben
| ≥ t H her i parçası için küçüktür. Benzer bir analizi takiben,
Düşman, her bölümde ihmal edilemez bir olasılıkla bir çatal oluşturmayı başarsa bile
r turu için, dört daldan en fazla biri r + 1 turunda büyüyebilir.
10.2 Çekişmeli Bölme
İkincisi, bölünmeye Karşı Taraf neden olabilir, böylece mesajlar yayılır.
Dürüst kullanıcılar tarafından bir kısımda dürüst kullanıcılara diğer kısımda doğrudan ulaşmayacak, ancak
Düşman, mesajları iki parça arasında iletebilir. Yine de, birinden bir mesaj
diğer tarafta dürüst bir kullanıcıya ulaşırsa, ikincisinde her zamanki gibi yayılır. Eğer

Düşman çok fazla para harcamaya isteklidir,
İnternet ve bir süre bu şekilde bölümleyin.
Analiz, yukarıdaki fiziksel bölümdeki daha büyük bölüm için olana benzer (daha küçük
Bölüm, popülasyona sahip olarak düşünülebilir 0): Düşman bir çatal oluşturabilir ve
her dürüst kullanıcı, şubelerden yalnızca birini görür, ancak en fazla bir şube büyüyebilir.
Toplamda 10.3 Ağ Bölümleri
Ağ bölümleri olabilir ve bölümler altında bir turda bir çatal oluşabilir, ancak orada
kalıcı bir belirsizlik değildir: çatal çok kısa ömürlüdür ve aslında en fazla tek bir tur sürer. İçinde
en fazla bir tanesi hariç bölümün tüm bölümleri, kullanıcılar yeni bir blok oluşturamaz ve bu nedenle
(a) ağda bir bölüm olduğunun farkına varın ve (b) asla "kaybolacak" bloklara güvenmeyin.
Teşekkürler
İlk olarak, bahsedilen Democoin sisteminin ortak yazarı Sergey Gorbunov'a teşekkür etmek istiyoruz.
En içten teşekkürler, birçok aydınlatıcı tartışma için Maurice Herlihy'ye işaret ettiği için.
ardışık düzenlemenin Algorand'ın üretim performansını iyileştireceğini ve
26 Bir kullanıcı bir adım bitmesi Not o azından ton gördü yalnızca 2λ süre beklemeden s H için
imzalar
aynı mesaj. Yeterli imza olmadığında, her adım 2λ süre sürecektir.

Sayfa 73

bu yazının önceki bir versiyonunun açıklaması. Sergio Rajsbaum'a yorumları için çok teşekkürler
bu yazının daha önceki bir versiyonu. Vinod Vaikuntanathan'a derin tartışmalar için çok teşekkürler
ve içgörüler. Yossi Gilad, Rotem Hamo, Georgios Vlachos ve Nickolai Zeldovich'e çok teşekkürler
bu fikirleri test etmeye başlamak ve birçok yararlı yorum ve tartışma için.
Silvio Micali, sayısız tartışma ve rehberlik için Ron Rivest'e şahsen teşekkür eder.
30 yılı aşkın süredir kriptografik araştırmada, belirtilen mikroödeme sistemini birlikte yazmak için
Algorand'ın doğrulayıcı seçim mekanizmalarından birine ilham vermiştir.
Bu teknolojiyi bir sonraki seviyeye taşımayı umuyoruz. Bu arada seyahat ve arkadaşlık
çok eğlenceliyiz, bunun için minnettarız.
Referanslar
[1] Bitcoin Hesaplama Atıkları, http://gizmodo.com/the-worlds-most-powerful-computer-network-is-being-was-504
2013.
[2] Bitcoinwiki. Teminat Kanıtı . http://www.blockchaintechnologies.com/blockchain-applications
5 Haziran 2016 itibarıyla.
[3] Coindesk.com.
Bitcoin:
Bir
Eşler arası
Elektronik
Nakit
Sistem
http://www.coindesk.com/ibm-reveals-proof-concept-blockchain-powered-internet-things/
Haziran 2016 itibarıyla.
[4] Ethereum. Ethereum. https://github.com/ethereum/. 12 Haziran 2016 itibarıyla.
[5] HowStuffWorks.com.
Nasıl
çok
gerçek para
dır-dir
Orada
içinde
the
dünya ?,
https://money.howstuffworks.com/how-much-money-is-in-the-world. htm. 5 itibariyle

Haziran 2016.
[6] en.wikipedia.org/wiki/Sortition.
[7] M. Ben-Or. Serbest seçimin bir başka avantajı: Tamamen eşzamansız anlaşma protokolleri. Proc. 2.Yıllık Dağıtık Hesaplama İlkeleri Sempozyumu, ACM, New York, 1983, s. 27-30.
[8] M. Castro ve B. Liskov. Pratik Bizans Hata Toleransı, Üçüncü Bildiri İşletim Sistemleri Tasarımı ve Uygulaması Sempozyumu. New Orleans, Louisiana, ABD, 1999, s. 173–186.
[9] DL Chaum, Rastgele Örnek Seçimler , https://www.scribd.com/mobile/document/236881043/Random-Samp
[10] B. Chor ve C. Dwork. Bizans mutabakatında, Rastgelelikte ve Hesaplama. S. Micali, ed., JAI Press, Greenwich, CT, 1989, s. 433-498.
[11] C. Decker ve R. Wattenhofer. Bitcoin Ağında Bilgi Yayılımı. 13-th IEEE Peer-to-Peer Computing Konferansı, 2013.
[12] D. Dolev. Bizans Generalleri Yeniden Grevde. J. Algorithms, 3, (1982), s. 14-30.
[13] D. Dolev ve HR Strong. Bizans anlaşması için doğrulanmış algoritmalar. SIAM Dergisi Hesaplama 12 (4), 656-666.

Sayfa 74

[14] C. Dwork ve M. Naor. İşleme veya Önemsiz Postayla Mücadele yoluyla Fiyatlandırma. Gelişmeler Cryptology, CRYPTO'92: Bilgisayar Bilimleri Ders Notları No. 740. Springer: 139-147.
[15] P. Feldman ve S. Micali. Senkron Bizans İçin Optimal Olasılık Algoritması Anlaşma. (STOC 88'deki ön sürüm.) SIAM J. on Computing, 1997.
[16] M. Fischer. Güvenilmez dağıtılmış sistemlerde fikir birliği sorunu (kısa bir anket). Proc. Uluslararası Hesaplamanın Temelleri Konferansı, 1983.
[17] S. Goldwasser, S. Micali ve R. Rivest. Uyarlanabilirliğe Karşı Güvenli Bir Dijital İmza Şeması Seçilmiş Mesaj Saldırısı. SIAM Journal of Computing, 17, No. 2, Nisan 1988, s.281-308
[18] S. Gorbunov ve S. Micali. Democoin: Herkese Açık Olarak Doğrulanabilir ve Ortak Hizmet Verilen Kripto para. https://eprint.iacr.org/2015/521, 30 Mayıs 2015.
[19] J. Katz ve CY Koo. Bizans Anlaşması için Beklenen Sabit-Yuvarlak Protokoller Üzerine. https://www.cs.umd.edu/~jkatz/papers/BA.pdf.
[20] A. Kiayias, A. Russel, B. David ve R. Oliynycov .. Ouroburos: Kanıtlanabilir bir güvenli kanıtı protokolü. Cryptology ePrint Arşivi, Rapor 2016/889, 2016. http://eprint.iacr.org/2016/889.
[21] S. King ve S. Nadal. PPCoin: Proof-of-Stake ile Eşler Arası Kripto Para Birimi, 2012.
[22] D. Lazar ve Y. Gilad. Kişisel iletişim.
[23] N. Lynch. Dağıtık Algoritmalar. Morgan Kaufmann Publishers, 1996.
[24] S. Micali. Algorand: Verimli Genel Muhasebe . https://arxiv.org/abs/1607.01341.
[25] S. Micali. Hızlı Ve Öfkeli Bizans Anlaşması. Teorik Bilgisayar Biliminde Yenilik 2017. Berkeley, CA, Ocak 2017. Tek sayfalık özet.
[26] S. Micali. Bizans Anlaşması, Önemsiz Yapıldı . https://people.csail.mit.edu/silvio/SelectedScientifi
[27] S. Micali, M. Rabin ve S. Vadhan. Doğrulanabilir Rastgele İşlevler. 40. Temelleri Computer Science (FOCS), New York, Ekim 1999.
[28] S. Micali ve RL Rivest. Mikroödemeler Yeniden Ziyaret Edildi. Bilgisayar Bilimi Ders Notları, Cilt. 2271, s. 149-163, Springer Verlag, 2002.
[29] S. Nakamoto. Bitcoin: Bir

Eşler arası
Elektronik
Nakit
Sistem.
http://www.bitcoin.org/bitcoin.pdf, Mayıs 2009.
[30] R. Pass ve E. Shi. Uzlaşmanın Uykulu Modeli. Cryptology ePrint Archive, Şubat 2017, Rapor 2017/918.
[31] M. Pease, R. Shostak ve L. Lamport. Hataların varlığında anlaşmaya varmak. J. Assoc. Comput. Mach., 27 (1980), s. 228-234.
[32] M. Rabin. Randomize Bizans generalleri. Bilgisayar Biliminin 24. Temelleri (FOCS), IEEE Computer Society Press, Los Alamitos, CA, 1983, s. 403-409.

Sayfa 75

[33] R. Turpin ve B. Coan. İkili Bizans anlaşmasının çok değerli Bizans'ı kapsayacak şekilde genişletilmesi
anlaşma. Bilgi vermek. İşlem. Lett., 18 (1984), s. 73-76.