CRYPTTECH
CYBER SECURITY INTELLIGENCE

CRYPTTECH
KRİPTOLOJİ LABORATUVARI

BLOK ZİNCİRİ

Büyük yenilikler her zaman teknolojik ve/veya fikri bir gelişmeyi de gerektirirler.
Kripto para uygulamalarının birtakım temel problemlerine çözüm önceden denenmiş birçok yöntemin
yepyeni bir bakış açısıyla birleştirilmesinde bulundu bir anlamda. Bu yeniliklerden biri de pek çok alanda uygulama umudu veren blok zinciridir.

MART — 2018
https://www.crypttech.com/

BLOK ZİNCİRİ
BİR VERİ TABANI YAPISI

Bir şeyi mümkün olan en basit şekilde anlatın; ama asla daha basit değil...

Her şeyin sınırsızca kopyalanabildiği bir yer olan dijital ortamda, koplayanarak çoğalmaması gereken bir olgu olarak paranın var olabilmesi oldukça zordur. Ortaya atılan tüm dijital para teorilerinde paranın kopyalanması probleminin çözümüne dair fikirler, yöntemler vardır. Belki artık zamanı geldiği için belki diğerlerinden daha başarılı olduğu için oldukça popüler olan Bitcoin Kripto Para Sisteminin bulduğu çözüm Blok Zinciri olarak bilinmekte. Ancak bu isimlendirme biraz kafa karıştırmakta. Çünkü aynı terim, arada bir fark gözetilmeden, bazen bir veri tabanı yapısının adı, bazen bir algoritmanın adı, bazen de kolektif otoriteye dayalı bir sistemin ismi olarak kullanılmakta. Blok zinciri aslında bir veri tabanı yapısıdır. Diğer kullanımları, kastettikleri şeylerin bu veri tabanı yapısının bazı avantajlarını kullanıyor olmalarından kaynaklanmakta.

Bir veri tabanı; oluşma zamanı, kayıt numarası, alfabetik dizin ve benzerleri gibi bir kritere göre birbiri ardına sıralanmış verilerden oluşur. Her veri tabanında verilerin korunması gerekir; aksi halde veri tabanının bir anlamı kalmaz. Ufak tefek değişimlerden tamamen silinmeye kadar geniş bir yelpazede gerçekleşebilecek içerik bozulmalarının iki genel nedeni vardır. Birinci neden, kullanıcı veya sistem hatalarıdır. Bu tür hatalara karşı, veri tabanının yedeklenmesi gibi önlemler mevcuttur ve genellikle iyi sonuç verir. İkinci neden ise veri tabanına yönelen saldırılardır.

Bir saldırıda amaç, verilerin tamamen silinmesi olabilir; ancak genellikle istenen, verilerin saldırgana birtakım avantajlar sağlayacak şekilde değiştirilmesidir. Doğal olarak, bu tür saldırılara karşı, verilerin korunması amacına yönelik çeşitli yöntemler de düşünülmüştür: veri tabanına ulaşımın kısıtlanması, verilerin şifrelendikten sonra kaydedilmesi... Fakat gerçekleştirilen saldırılar daha çok sistem açıklarından yararlanma ve veri tabanına ulaşım izni olan kullanıcıların hesaplarının çalınması yöntemiyle olmaktadır. Örneğin; saldırgan, veri tabanında değişiklik yapma izni olan bir kullanıcının isim ve şifresini kullanarak işlem yapmaktadır. Bu durumda veri tabanı üzerinde bir önlem almanın anlamı kalmaz. Çünkü yapılan işlemler yapılmasına izin verilen işlemler olup sadece gerçekleştiren kişi, gerçekte olması gereken, olduğu farz edilen kişi değildir.

Veri tabanları; satır yapılarına, kayıt sistemlerine, içerik analizlerine göre pek çok farklı türe ayrılır. Bazı veri tabanları bazı uygulamalar açısından daha elverişlidir. Ama tüm veri tabanlarının ortak özellikleri vardır ve yukarıda saydığımız saldırı türlerine karşı hiçbiri tam anlamıyla bağışık değildir. Her saldırı mekanizması, her veri tabanına göre uyarlanabilir. Bu yazımızda blok zinciri adı verilen veri tabanı türünü

inceleyeceğiz. Bu veri tabanları hem bazı kriptolojik bilmecelerin zaman kısıtlaması altında çözülebilmelerinin imkânsızlığından hem de kolektif bir otoritenin kontrolünde olduklarından bilinen saldırılara karşı tamamen korunaklıdırlar. Ancak bu sözlerin ardından şunu eklemek zorundayız: Bir protokolü çökertebilmek için saldırganların da çalışmaya, zamana ihtiyacı vardır. Bugün kırılması imkânsız gibi görünen kilitlerin yarın da kırılamayacağının bir garantisi verilemez. Evet, bir sistem bilinen saldırılara karşı matematik olarak tam korunaklıysa *bu saldırılara karşı* her zaman tam korunaklıdır! Fakat henüz bilinmeyen türde bir saldırı her an ortaya çıkabilir...

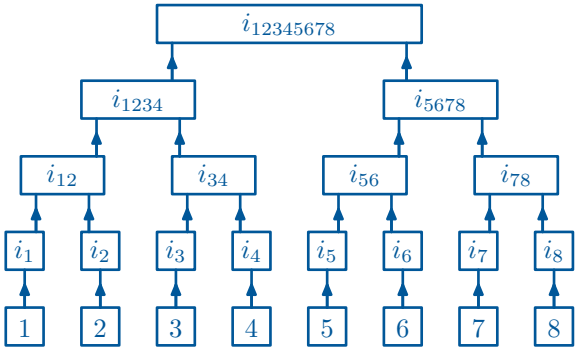
Aşağıdaki paragraf, yazının sonunda bir kez daha yer almaktadır. Burada yer almasının nedeni, okumaya bir yön vermesi; sonda yer almasının nedeni ise yazılanlardan bir özet çıkarmasıdır.

Bir veri tabanındaki verilerin korunması demek, kayıtlı verilerin değiştirilememesi demektir. Bu, bilindik veri tabanlarında, kullanıcı yetkilerinin düzenlenmesi yoluyla gerçekleşir. Sistem, kullanıcının veri değiştirme ve ekleme yetkisini kontrol eder ve yetkisiz kişilerin işlem yapmasına izin vermez. Blok zinciri bu anlamda bir yetki anlayışına sahip değildir. Zincirdeki her blok kriptolojik kilitlerle korunmaktadır. Aslında bu kilitleri aşabilen herkes tüm yetkiye sahiptir. Ancak kilitler, pratik olarak aşılması imkânsız kilitlerdir. Dolayısıyla; istenen yetkiye sahip birinin yetkilerini gasp ederek işlem yapabilmek gibi bir durum söz konusu değildir. Kilitlerin aşılamazlıkları kriptolojik zorluklardan değil, zaman sınırlamasından kaynaklanmaktadır. *Sınırlı işlem gücü ama sınırsız zamanı* veya *sınırlı zamanı fakat sınırsız işlem gücü* olan bir saldırgan tüm zinciri istediği gibi baştan tasarlayabilir. Hiç kimsenin sınırsız bir işlem gücü olmadığı bir olgudur. Çok fazla zamana sahip olma seçeneğiyse yeni blok ekleme hızı gibi sistemin temelinde yer alan zaman sınırlamaları ile bertaraf edilir. Diğer taraftan, zincir örneklerinin çok fazla kişide bulunuyor olması da değişikliklerin kontrol edilebilmesini kolaylaştırır. Saldırganın başarılı olması için hem *tarihi çok hızlı yazması* hem de tüm kopyaların çoğunluğunu değiştirebilecek erişime sahip olması gerekmektedir.

1. Bir veri kaydetme yöntemi olarak blok

Belli sayıda verinin birleştirilerek belli bir algoritma doğrultusunda kriptolojik olarak *kilitlendiği* veri tabanı yaprağına **blok** adını veriyoruz. Bloğun oluşumunda en önemli adımlardan biri, bloğa kaydedilecek olan verilerden bir **Merkle ağacı** oluşturmaktır. Kriptoloji uzmanı Ralph Merkle adına izafeten Merkle ağacı adı verilen bu yapı, ters yönde ilerleyen bir ağaç topolojisine göre elde edilen iz (hash) değerlerden oluşur. En son elde edilen iz değere de **Merkle kök değeri** adı verilir. Verilerden hareketle kök değerin hesaplanması zor bir işlem değildir. Bundan ötürü, kök değer değiştirilmeden verilerin değiştirilebilmeleri mümkün olmaz; sistem, değerleri sürekli bir şekilde kontrol edip değişimi anında raporlayabilir. Bir deyişle, Merkle kök değeri, yaprakta yer alan verileri kriptolojik olarak kilitler.

Bir blok genellikle iki bölümden oluşur. Birinci bölüm, verilerin yer aldığı **içerik** adlı bölümdür. İkinci bölüme ise **başlık** adı verilir. Bir blokta, bloğun kullanım amacına göre, veriler dışında, değiştirilmemesi gereken başka birtakım bilgiler de bulunabilir. Örneğin; bloğun oluşturulma tarihi ve zamanı, bloğu oluşturan kişinin adı, bloğun oluşturulma nedeni gibi. Eğer bu tür bilgiler varsa, Merkle kök değeri ile birlikte, bloğun başlık bölümünde yer alırlar. Başlıktaki bilgiler birleştirilerek iz değerleri alınır. Bu değer, bloğu temsil eden iz değerdir. Bu değerin oluşumunda kullanılan veri ve bilgiler o şekilde birleştirilmiştir ki, bir tekinin dahi değişmesi iz değerin değişmesine neden olur. Dolayısıyla, bu değeri elde etmek için kullanılan değerler kriptolojik olarak kilitlenmişler, korumaya alınmışlardır—tekil olarak değiştirilebilmeleri mümkün değildir.

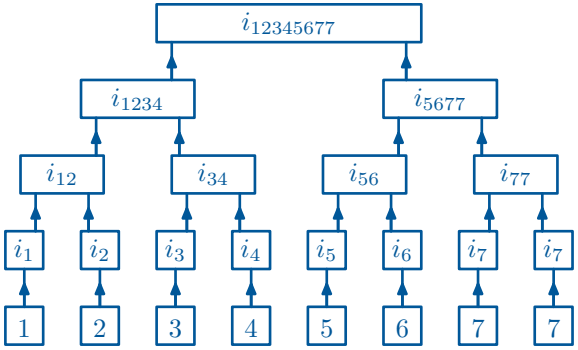


Şekil 1. Sekiz yapraklı bir Merkle ağacı

Yukarıdaki blok örneğimizde, 8 tane veri kullanacağız; ancak bu sayının istendiği gibi artırılıp azaltılabileceği açıktır. Şekilde görülen blokta veriler belli bir ölçüte göre 1'den 8'e doğru sıralanmıştır. En alt sırada verilerin kendilerini temsil eden kareler görülmektedir. Verilerin kendileri rakamlarla gösterilmiştir. Blok oluşturulurken bu verilerin iz değerleri kullanılır. n rakamı ile temsil edilen verinin iz değerini i_n şeklinde gösteriyoruz. İkinci sırada, bu iz değerleri temsil eden kareler yer almaktadır.

Verilerin iz değerlerini ikişer ikişer birleştirilerek bir kez daha iz değerleri elde ediyoruz. Örneğin; 1 ve 2 numaralı verilerin iz değerleri olan i_1 ve i_2 değerlerini birleştirerek elde ettiğimiz bit dizisinden yeni bir iz değer elde ediyoruz ve bu değeri de i_{12} şeklinde göstermekteyiz. Tüm verilerin bu şekilde iz değerlerini alırsak i_{12}, i_{34}, i_{56} ve i_{78} gibi dört tane iz değer elde ederiz. Şekilde, bu iz değerleri üçüncü sıradaki dikdörtgenler temsil etmektedir. Yine aynı şekilde devam ederek bu dört iz değeri de ikişer ikişer birleştirip iz değerlerini alıyoruz. Bu noktada elimizde iki yeni iz değer mevcuttur: i_{1234} ve i_{5678}. Son adımımız bu iki değerin izini almak: $i_{12345678}$. Elde ettiğimiz bu son değer, blokta kaydedilmiş olan sekiz verinin Merkle kök değeridir. Şekilde açıkça görülmektedir ki, ele alınan sekiz veriden bir tanesi bile değiştirilse Merkle kök değeri değişecektir. Diğer taraftan, i_1 ve i_2 iz değerlerinin birleştirilme şekli; yani, $i_1 i_2$ şeklinde veya $i_2 i_1$ şeklinde birleştirilmesi bu birleşimin üreteceği iz değeri değiştireceği için Merkle kökü, aynı zamanda, verilerin sıralamasını da koruma altına alan bir değerdir.

Kullandığımız örnekte Merkle kökünü 8 veriden oluşturduk. $8 = 2^3$ eşitliği gereği, her ikili iz değer alma işleminin sonucunda yine 2'nin katı sayıda iz değer elde ettik. Ancak bunun her zaman böyle olması şartı yoktur. Eğer blokta kaydedilecek olan veri sayısı 2^n şeklinde ifade edilemiyorsa en az bir noktada tek sayıda iz değerle karşılaşılacaktır.



Şekil 2. Yedi veriden elde edilen Merkle ağacı

Örneğin; 10 veriden yola çıkılarak oluşturulan bir Merkle ağacında ikinci sırada 5 iz değer elde edilecektir. Böyle bir durumda yapılacak olan en sonda kalan iz değeri bir kez daha yazıp çift sayıda iz değer elde etmektir. Yukarıdaki şekilde 7 veriden elde edilen bir Merkle ağacı gö-

rülmektedir.

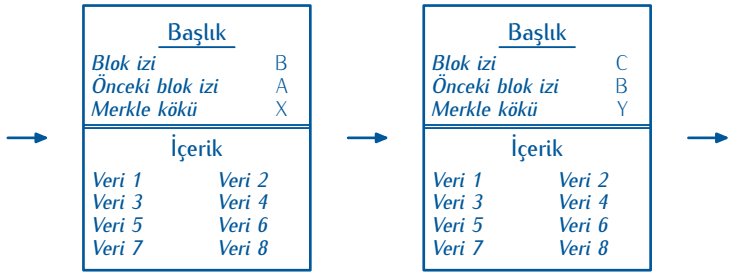
Yukarıda söylediğimizi tekrar edersek: Merkle kök değeri bloktaki tüm verileri kilitlemek için yeterlidir. Ancak bloğun oluşturulma amacına göre, Merkle kök değeri ile birlikte başka bilgilerin de yer aldığı bir blok başlığı bölümü oluşturulabilir ve bu bölümdeki bilgilerin bir iz değeri alınabilir. Bu iz değer bloğu temsil eden iz değer olacaktır.

2. Blok Zinciri: Bir veri tabanı olarak

Çok sayıda bloğu bir araya toplarsak bir veri tabanı oluşturmuş oluruz. Eğer bloklar belli bir sıraya göre dizilirler ve bu sıranın değişmesi de bir şekilde engellenirse elde edilen veri tabanı sayfa numaraları olan bir kitabı andırır. Bu şekilde oluşturulmuş veri tabanlarına **blok zinciri** adı vermekteyiz.

Blok zincirinde yer alan bloklar yapısal olarak iki bölümden oluşurlar [1]. Birinci bölüm, bloğa kaydedilmiş olan verilerden oluşan ve bir Merkle kök değeri ile kilitlenen **içerik** bölümüdür. İkinci bölüm ise, veriler dışında kalan ve değişmeden saklanması istenen bilgilerin yer aldığı başlık bölümüdür. Blokların oluşturulma amacı bir veri tabanı elde etmek olduğuna göre, başlık kısmında yer alacak olan bilgiler de bu doğrultuda seçilmelidir. Buna göre, başlık kısmında Merkle kök değeri ve bir önceki bloğun iz değeri başlıkta yer almalıdır. Bir önceki bloğun iz değerinin başlıkta yer alması, sayfa numarası eklemeye benzer bir işlemdir; bu değere bakılarak bloğun hangi bloktan sonra geleceği belirlenmiş olacaktır.

Dikkat edilirse bloğun iz değeri, sadece kendi içindeki verileri değil, kendinden önceki blokta yer alan verileri de değişmemek üzere kilitlemektedir. Çünkü, iz değerin elde edilmesinde kullanılan bir önceki bloğa ait iz değer, o blokta yer alan verilerden elde edilmiş olan Merkle kök değerini de içinde barındırmaktadır. Bu düzen bir silsile şeklinde geriye doğru işleyeceğinden ötürü, her blok kendinden önceki tüm blokları kilitlemektedir. Buna göre, bir blokta değişiklik yapmak demek o bloktan sonra gelen tüm blokları değiştirmeyi gerektirir. Buna izin verilip verilmeyeceği sistem yönetiminin kararıdır.



Şekil 3. Bir veri tabanı olarak blok zinciri

Yukarıdaki şekilde her biri 8 veri barındıran bir blok zincirinin temsili görülmektedir. Sol tarafta yer alan bloğun iz değeri (B), sağ tarafta yer alan blokta *"önceki blok izi"* olarak görülmektedir.

3. Blok Zinciri: Bir protokol olarak

Yukarıda bahsettiğimiz gibi, blok zinciri şeklinde oluşturulan bir veri tabanında herhangi bir bilginin, diğer bilgilere hiç dokunulmadan değiştirilebilmesi mümkün değildir. Herhangi bir blokta yer alan bir bilginin değiştirilebilmesi için o bloğun ve ardından oluşturulmuş bütün blokların yeniden yazılması gerekir—Satoshi Nakamoto'nun deyişiyle *tarihin yeniden yazılması*dır bu. Fakat, yine yukarıda bahsettiğimiz

1: Bir blokta başlık bölümü olması şartı yoktur; fakat blok zincirinde yer alan bloklarda başlık bölümü mutlak surette bulunur.

gibi, verilerin iz değerlerinin hesaplanması çok da zor olmayan bir iştir.

Şimdi bir felaket senaryosu yazalım. Bir saldırganın, bin bloktan oluşan bir blok zincirinin ellinci bloğunda yer alan bir bilgiyi değiştirmek istediğini farz edelim. Bunun için yapması gereken ellinci bloktaki (neredeyse) tüm iz değerleri yeniden hesap etmek ve bunun ardından da tam 950 tane blok için başlıktaki bilgilerle blok izi hesaplamaktır. Bu büyük bir iştir. Diğer taraftan, bloklar tek bir kişi tarafından değil de birkaç kişi tarafından oluşturuluyorsa bahsedilen tüm işin, diğer kişiler yeni bir blok oluşturmadan önce yapılması gerekir. Yani, yeni blok (1001. blok) en son bloğun (1000. bloğun) iz değerini kapsayacağı için yeni blok oluşturulmadan en son bloğun *yeni* iz değeri belirlenmiş olmalıdır. Dolayısıyla, *tarihin yeniden yazılma hızı*, tarihin doğal yazılma hızının önüne geçmelidir. Bu, daha da büyük bir iştir; fakat, bazı önlemler alınmazsa, yapılamayacak kadar büyük bir iş değildir.

Bu işi yapılamaz kılmanın bir yolu blok zincirini birkaç farklı noktada muhafaza etmek; yani, birçok kopyasını alıp herhangi bir kopyadaki farklılığı diğer kopyalarla kıyaslama yoluyla değerlendirmektir. Bu güvenilir bir yol olsa da eğer ki, değişikliği gerçekleştirmek isteyen, tekil bir saldırgan değil de bir şebeke ise bu işin üstesinden gelebilir. Şebeke, kopyaların en az %51 çoğunluğunu değiştirmede başarılı olursa gerçek kopyaların bu kopyalar olduklarını iddia eder ve hatta bununla da kalmayarak iddiasını kabul ettirebilir.

İkinci bir yöntem de *iş ispatı* adı verilen yöntemdir. Hatırlayalım ki, iz değerler ve dolayısıyla blok izleri on altı tabanlı sayı sisteminde bir sayı ifade ederler. İş ispatı yönteminde, her bir blok izinin belli bir sayıdan daha küçük olması istenir. Bu sayı, her bir blok için ayrıca hesaplanan bir değerdir. Bir bloğun başlık kısmında yer alan bilgiler, belli verilere işaret ettiklerinden ötürü, değiştirilebilir bilgiler değildirler. Dolayısıyla, başlıkta yer alan bilgilerin verecekleri iz değer belli olup bu izin sayısal değeri de bellidir; yani, belli bir sayıdan küçük veya büyük olması ayarlanabilir bir şey değildir. Bu ayarlamanın yapılabilmesi için başlık kısmına bir değişken değer eklenir. İngilizcede *nonce* adı verilen, bizim **başlık değişkeni** diyeceğimiz bu değişken değerin tek amacı, blok izinin istenen değerden küçük olmasını sağlamaktır. İz değerin, başlık değişkeninin hangi değerleri için istenenden daha küçük olacağı, iz fonksiyonların yapısı nedeniyle bilinemez; öğrenmenin tek yolu deneme–yanılmadır. Buna göre, başlık değişkenine çeşitli değerler vererek iz değerin istenen değerine ulaşılmaya çalışılır. (Başlık değişkeni, bu değeri verebilecek olan pek çok sayıda değer alabilir; yalnızca bir tanesini bulmak bloğu kilitlemek için yeterlidir.) Bu denemeler, doğal olarak zaman alır ve bu zaman bir zorluk derecesi olarak düşünülebilir. Oluşturulmuş bir bloğun istenen değerde olmasını sağlayan başlık değişkeninin değeri blok başlığına yazılır. Dolayısıyla blok izinin doğruluğu isteyen herkes tarafından kontrol edilebilir. Diğer taraftan, tarihi yeniden yazmak isteyen saldırgan her bir blok için başlık değişkeninin değerini yeniden hesaplamalıdır. Çünkü değiştirilen blokta Merkle kökü, takip eden bloklarda ise önceki blok izi değiştiği için bloğun ilk oluşumunda kullanılan başlık değişkeni değeri kullanılamaz; kullanılması, istenen sonucu vermez. Buna göre, tarih ne kadar eskiden başlanarak yeniden yazılacaksa iş o kadar büyüyecektir.

Şimdiye kadar bahsedilen kriptolojik kilitlerin zorluğudur. Fakat eğer saldırganın (sınırsız işlem gücü olamayacağı için) sınırsız zamanı varsa bu kilitleri aşabilir.

Saldırganın kilitleri aşmasını önlemenin yolu, zamanına sınırlama getirmektir. Eğer zincire belli bir zaman aralığıyla sürekli bir şekilde yeni bloklar eklenirse saldırganın zamanı kısıtlanmış olur. Blok zincirinin kriptolojik kilitlerinin aşılamamasının nedenlerinden biri budur. Öte yandan zincirin kopyalarının sistemdeki herkese dağıtılmış olması da çok ciddi bir önlemdir. Çünkü saldırgan, zincir kopyalarının çoğunluğunu değiştirmediği takdirde başarılı olamaz. Kopyaların birbirleri ile kıyaslanmaları gerçeği ortaya çıkaracaktır.

Başlık değişkeninin değerinin hesaplanmasının ne kadar zaman alacağı, ihtimal hesabı ile aşağı yukarı bulunabilir. Bu zaman ne kadar uzun olursa tarihi yeniden yazmak o kadar güçleşir; ama bu, aynı zamanda, zincire yeni blok eklemeyi de zorlaştırır. Fakat amaca ve uygulama koşullarına bakılarak optimum bir değer bulmak mümkündür.

Bir veri tabanındaki verilerin korunması demek, kayıtlı verilerin değiştirilememesi demektir. Bu, bilindik veri tabanlarında, kullanıcı yetkilerinin düzenlenmesi yoluyla gerçekleşir. Sistem, kullanıcının veri değiştirme ve ekleme yetkisini kontrol eder ve yetkisiz kişilerin işlem yapmasına izin vermez. Blok zinciri bu anlamda bir yetki anlayışına sahip değildir. Zincirdeki her blok kriptolojik kilitlerle korunmaktadır. Aslında bu kilitleri aşabilen herkes tüm yetkiye sahiptir. Ancak kilitler, pratik olarak aşılması imkânsız kilitlerdir. Dolayısıyla; istenen yetkiye sahip birinin yetkilerini gasp ederek işlem yapabilmek gibi bir durum söz konusu değildir. Kilitlerin aşılamazlıkları kriptolojik zorluklardan değil, zaman sınırlamasından kaynaklanmaktadır. *Sınırlı işlem gücü ama sınırsız zamanı* veya *sınırlı zamanı fakat sınırsız işlem gücü* olan bir saldırgan tüm zinciri istediği gibi baştan tasarlayabilir. Hiç kimsenin sınırsız bir işlem gücü olmadığı bir olgudur. Çok fazla zamana sahip olma seçeneğiyse yeni blok ekleme hızı gibi sistemin temelinde yer alan zaman sınırlamaları ile bertaraf edilir. Diğer taraftan, zincir örneklerinin çok fazla kişide bulunuyor olması da değişikliklerin kontrol edilebilmesini kolaylaştırır. Saldırganın başarılı olması için hem *tarihi çok hızlı yazması* hem de tüm kopyaların çoğunluğunu değiştirebilecek erişime sahip olması gerekmektedir.

Yararlanılan Kaynaklar

1. Mastering Bitcoin (2nd Edition)
Andreas M.Antonopoulos
O'Reilly 2017 - ISBN-13: 978-1-491-95438-6

2. Decentralized Applications
Siraj Raval
O'Reilly 2016 - ISBN-13: 978-1-491-92454-9

3. Blockchain Basics
Daniel Drescher
Apress 2017 - ISBN-13: 978-1-4842-2604-9