Blokzincir ile Gerçek Zamanlı Menkul Kıymet Transferi*

(Blockchain Based Real-time Gross Settlement)

Octabase Blockchain Labs.
labs@octabase.com

Temmuz, 2018

1 Giriş

Octabase, bu kavram kanıtlama çalışması kapsamında; kurumlar arası gerçekleşen kıymet transferlerinin, güvenilir bir üçüncü parti kuruma ihtiyaç olmaksızın, kurumdan kuruma doğrudan yapılabilmesine imkan tanıyacak bir blokzincir mimarisi ve kriptografi protokolü tasarlamış ve prototip geliştirmesini sunmuştur.

Blokzincir üzerinde gerçekleşen işlemler tüm katılımcılar tarafından dağıtık deftere kaydedilmektedir. Blok onay yetkisi olan tüm katılımcılar (*verifiers*), blok içerisinde yer alan transferlerin doğruluğunu kontrol edecektir.

İşlemlerin hangi kullanıcı tarafından yapıldığı, transferin kime gönderildiği, işlem miktarı ve işleme dair menkul kıymetin cinsi şifrelenmiş olarak tutulacaktır. Şifrelenen bu bilgileri yalnızca; işlemi hazırlayan kişi, transferin yapıldığı kişi ve *Auditor* rolünde bulunan katılımcı(lar) deşifre edebilecektir (*block verification*). Bunun sonucu olarak katılımcıların bakiye bilgileri de yalnızca kendilerinin ve denetçi rolünde bulunan katılımcının görebilmesi mümkün olacaktır.

Blokzincirde kullanılabilecek menkul kıymet cinsleri (*asset*) için *Asset Manager* adında rol tanımlanmış olup, bu role sahip kullanıcıların yeni menkul kıymet cinsleri yaratabilmesine olanak tanınmıştır. Yaratılan menkul kıymetlerin dolaşımdaki miktarını düzenleyebilmek için *Issuance Manager* adında bir rol tanımlanmış olup, bu role sahip katılımcıların yeni menkul kıymetleri dolaşıma sokabilmesine yada dolaşımdaki menkul kıymetleri tekrar kullanılamayacak şekilde imha edebilmesine olanak tanınmıştır (*issuance, reissuance, burn*).

*İşbu doküman Octa Blockchain Platformunun işlemlerde kullanmış olduğu kriptografik protokolü açıklamak için hazırlanmıştır. Dokümanda geçen tüm yöntem ve tekniklerin ticari kullanım hakları Octabase lehine saklıdır.

Çalışma kapsamında genel olarak kullanılan teknolojiler; izinli blokzincir mimarisi (*permissioned blockchain*), eliptik eğri şifrelemesi (*ECC, elliptic curve cryptography*) ve harcanmamış işlem çıktısı şeması (*UTXO, unspent transaction output*) ile sıfır bilgi ispatlarını (*zero-knowledge proofs*) içermektedir.

2 Güvenlik

2.1 Güvenlik Seviyesi

2.1.1 Eliptik Eğrilerde Ayrık Logaritma Problemi (*ECDLP*)

Aşağıdaki tabloda(1) eliptik eğrilerdeki ayrık logaritma probleminin çözümü için gerekli olan hesaplama gücü gösterilmiştir. Odlyzko'nun çalışması[2] 10^{14} MIPS yılı sürecek bir hesaplama işleminin 2014 yılı için dünyadaki tüm hesaplama gücü ile ancak 1 yılda tamamlanabileceğini tahmin etmektedir. Bu çalışmada kullanmış olduğumuz eliptik eğri $2^{255}-19$ bitlik bir asal sayı p kullanmakta ve $\log_2 \sqrt{\frac{2^{255}-19}{4}} \simeq 128$ bit güvenlik seviyesi sağlamaktadır. Bu veriler ışığında ve Moore kanununa göre yeniden ölçeklendirdiğimiz güncel hesaplama gücü ile çalışmamızda temel aldığımız eliptik eğri şifrelemesinin kırılabilmesi için dünyadaki tüm hesaplama gücünün 6×10^{11} yıl çalışması gerekmektedir. Kuantum bilgisayarları sonrası kullanılabilecek şifreleme (*post-quantum cryptography*) teknolojileri ve alınabilecek önlemlere ayrıca değinilmiştir.

Bit uzunluğu (n)	$\sqrt{\pi n/4}$	MIPS yılı
160	2^{80}	8.5×10^{11}
192	2^{96}	5.6×10^{16}
224	2^{112}	3.7×10^{21}
256	2^{128}	2.4×10^{26}
384	2^{192}	4.4×10^{45}
521	2^{260}	1.3×10^{66}

Tablo 1: ECDLP'yi çözmek için gerekli hesap gücü (ANSI 9.62b'ye göre [1])

2.1.2 Diffie-Hellman Problemi (*CDH, DDH*)

Çalışmamızda, blokzincir katılımcıları arasında gerçekleşen özel anahtar değişim prosedürleri Diffie-Hellman yöntemini temel almaktadır. *Computational Diffie-Hellman* varsayımı ile $g, g^a, g^b \in \mathbb{G}$ verildiğinde $g^{a \cdot b}$'yi doğrusal zamanda hesaplayabilecek bir algoritmanın var olmadığı kabul edilmektedir.

Buradan yola çıkarak *Decisional Diffie-Hellman* varsayımı ile $g, g^a, g^b, g^c \in \mathbb{G}$ verildiğinde $g^c = g^{a \cdot b}$ eşitliğinin doğruluğunu saptayabilecek ve doğrusal zamanda çalışacak bir algoritmanın var olmadığı kabul edilmektedir.

2.2 Yan Kanal Saldırıları

2.2.1 Zamanlama Saldırıları (*Timing Attacks*)

Herhangi bir kriptosistemde gerçekleşen şifreleme yada deşifre işlemi, kullanılan anahtar veya kapalı-açık metin içeriğinden kaynaklı olarak çalışma zamanında farklı sürelerde tamamlanıyorsa, bu kriptosistemin zamanlama saldırılarına karşı zaafiyet barındırdığı söylenebilir. Öyle ki açık metinde yapılacak değişiklikler ve çalışma zamanının ölçümlenmesi ile istatiksel olarak özel anahtarın açığa çıkarılması mümkün olabilir.

Bu çalışmada bu saldırı yönteminden korunabilmek için ekle ve katla algoritmasının (*add-and-double*) ve Montgomery çarpımının uygulanabildiği bir eliptik eğri kullandık. Bu sayede eliptik eğri üzerinde yapılan en temel işlem olan nokta toplama, dolayısı ile tamsayı ve nokta çarpımı işlemi parametre değerlerinden bağımsız olarak eşlenik hesaplama gücü ile çalışan süreçler olarak implemente edilebildi. Bunun sonucu olarak sistem güvenliğinin temel aldığı ayrık logaritma probleminin, en popüler yan kanal saldırılarından olan zamanlama atakları ile çözülemeyeceği bir kriptosistem elde etmiş olduk.

2.2.2 Önbellek Saldırıları (*Cache Attacks*)

Kriptosistemin çalışması anında kullanmış olduğu sistem belleği ve işlemci önbelleği üzerinde tutulan verilerin okunabilmesi özel anahtarın açığa çıkmasına neden olabilecek bir zaafiyet yaratmaktadır. Bundan korunmak için ve toplamda sağlayacağı pek çok avantajdan faydalanabilmek adına Intel'in SGX (*Security Guard Extension*) özelliğinin kullanılması planlanmaktadır. Bu sayede işlemci üzerinde yalıtılmış bir alanda özel-açık anahtar ikilisi üretilecek, açık anahtar ile şifrelenmiş algoritma işlemcinin yalıtılmış bölgesinde özel anahtar ile deşifre edilerek çalıştırılacak ve çalışma zamanında şifreli bellek erişimi yapacaktır. Bu teknik ile diğer pek çok kazanımın yanında önbellek saldırılarından korunmayı planlamaktayız.

2.3 Donanımsal Güvenlik Modülleri (*HSM*)

Özel anahtar kullanılarak yapılan anahtar değişimi, deşifre ve imzalama işlemlerinin, bilgisayarın dışında başka bir cihaz ile yapılması ve özel anahtarın cihaz içerisinden çıkarılmasının önünde fiziksel engeller bulunması nedeni ile HSM'lerin kullanımı gün geçtikçe yaygınlaşmaktadır. Çalışmamızda zincire eklenecek blokları onaylayan katılımcılar ve transferi gerçekleştiren cüzdan sahipleri yoğun olarak özel anahtarlarına bağlı işlemler yapmaktadırlar. Bu

işlemlerin bir HSM cihazında yapılabilmesi teknik olarak mümkün olup, gereksinim halinde kullandığımız standart algoritmaları destekleyen bir HSM ile entegrasyonu hızlıca sağlanabilir.

2.4 Kuantum Bilgisayarları Sonrası Kriptografi

Günümüzde gelişim hızı ivme kazanan kuantum bilgisayarları modern kritolojinin temelleri olan imzalama ve özel anahtar değişim algoritmaları için büyük bir tehdit oluşturmaktadır. Günümüz yaşantısının ayrılmaz unsurları haline gelen mobil iletişim, kredi kartları, bankacılık işlemleri ve diğer pek çok uygulama kuantum bilgisayarlarının gelişimi karşısında bu tehlike ile karşı karşıya bulunmaktadır. Shor algoritması ile ayrık logaritma problemlerinin doğrusal zamanda çözülebilmesi sebebi ile yeni kriptosistem çalışmalarının kuantum dirençli algoritmalar yönünde hız kazanmasına yol açmıştır. Aşağıdaki tabloda(2) Shor algoritmasının RSA ve eliptik eğri şifrelemesinde kullanılan ayrık logaritma probleminin çözümü için ihtiyaç duyduğu sistem kaynakları gösterilmiştir.

Şema	Anahtar uzunluğu	Kübit Sayısı	Toffoli Kapı Sayısı
RSA	3072	6146	5.2×10^{12}
RSA	15360	30722	2.87×10^{15}
ECC	256	2330	1.26×10^{10}
ECC	521	4719	1.14×10^{12}

Tablo 2: Açık anahtar şifrelemesinin kırılması için gerekli kuantum kaynağı (ECRYPT 2018'e göre [3])

Bu veriler ışığında bu çalışmada kullandığımız kriptoloji şemasının 2330 kübit ve 1.26×10^{10} toffoli kapısına sahip bir kuantum bilgisayarı tarafından kırılabileceğini söyleyebiliriz. Shor algoritması evrensel kuantum bilgisayarına (*universal quantum computer*) ihtiyaç duymaktadır. 2018 Temmuz ayı itibari ile dünyadaki en gelişmiş evrensel kuantum bilgisayarı 72 kübit ile rekoru elinde tutan Bristlecone[4] projesinde yapılmıştır. D-Wave isimli şirketin 2000 kübitlik kuantum bilgisayarları olsa da amaca yönelik hazırlanmış ve kuantum tavlama (*quantum annealing*) tekniği ile çalışan sistemler oldukları için modern kriptoloji için doğrudan bir tehdit unsuru olarak görülmemektedir.

Blokzincir teknolojisi ile gerçek zamanlı kıymet transferinin uygulanması dağıtık defter üzerinde şifrelenmiş kayıtların sistem terkedilene dek saklanması anlamına gelmektedir. Bir transferin gerçekleşebilmesi için gönderen kişinin ilgili varlığa sahip olduğunu dijital imza ile kanıtlaması gerekmektedir. Sahip olduğu varlık miktarının, gönderdiği varlık miktarına eşit olduğunu, daha az olmadığını ise taahhüt şemaları ile kanıtlamaktadır. Bu çalışmada

bu amaç doğrultusunda kullanmış olduğumuz Pedersen taahhüt şeması mükemmel gizleme (*perfect hiding*), hesaplanabilir bağlama (*computationally binding*) ilkesi ile çalışmaktadır. Bunun anlamı bir taahhütün hangi miktardaki varlık için verildiğinin ayrık logaritma problemini çözülerek dahi tespit edilememesi, ancak ayrık logaritma problemini çözen birisinin başka bir miktardaki varlık için aynı taahhütü üretebilmesine imkan bulunmasıdır.

Hem mükemmel gizleme hem de mükemmel bağlama özelliğine sahip bilinen bir taahhüt şeması bulunmamaktadır. Kuantum bilgisayarlarının ayrık logaritma problemini pratikte çözebildiği bir olasılık gerçekleştiğinde bu duruma hazır olmak için kriptosistemi şimdiden bu risklere göre tasarlamamız gerekmektedir. Bu amaç doğrultusunda DLP'nin çözülebildiği bir zamanda bu çalışmada önerilen sistemi kullanan bir katılımcının sahip olmadığı miktardaki bir varlık için taahhüt oluşturmasının önüne geçilmesi adına bazı önlemler planlanmıştır. Pedersen yerine, hesaplanabilir gizlilik ve mükemmel bağlama ilkesi ile çalışan ElGamal taahhüt şemasının kullanımı değerlendirilmiştir. Bu sayede DLP çözülebildiğinde katılımcıların sahip olmadıkları miktarda transfer gerçekleştirmesinin önüne geçilmiş ancak DLP çözümü ile sahip olunan varlık miktarının tespit edilebileceği bir yöntem elde etmekteyiz. Ancak ElGamal şemasında oluşturulan bir taahhütün boyutu Pedersen şemasında oluşturulan taahhütlerin iki katı olmakta ve bu durum dağıtık defter boyutunun iki kat artmasına sebep olmaktadır. Bu durumun önüne geçmek adına DLP'nin çözülebildiği bir durum mümkün olduğu takdirde Pedersen şemasından ElGamal şemasına geçişi mümkün kılacak bir yöntemin[5] kullanılması öngörülmüştür.

Kuantum sonrası şifreleme alanı hızla gelişmekte ve her geçen gün yeni teknikler önerilmektedir. Bu çalışmada kullanılan eliptik eğri, imza algoritmaları ve özel anahtar değişim algoritmaları yukarıdaki tabloda(2) belirtilen özelliklere sahip olan kuantum bilgisayarlarına karşı güvenli değildir. Kuantum bilgisayarlarına karşı ilk önlemimiz gelişmiş taahhüt şemaları ile sistem içerisinde sahte kıymet üretilmesinin önüne geçmektir.

İlerleyen süreçte hash temelli imzalar, latis tabanlı şifreleme, süper tekil eliptik eğrilerdeki izojeni gibi teknikleri sisteme dahil ederek mevcut güvenlik seviyesinin kuantum bilgisayarlara karşı korunması hedeflenmektedir.

2.5 Sözde Rastsal Sayı Üreteci (*PRNG*)

Modern kriptosistemlerin temel bileşenleri rastsal üretildiği varsayılan sayılara dayanmaktadır. Oluşturulan özel-açık anahtarlar, üretilen dijital imzalar, simetrik şifrelemede kullanılan başlangıç değerleri gibi pek çok süreç rastgele üretilen sayıları temel almaktadır. Rastgele üretildiği varsayılan sayıların bir değer etrafında kümeleşmesi, sayıların belli bir desene göre üretilmiş olması yada sürekli aynı sayının üretilmiş olması kriptosistemde kullanılan güvenlik önlemlerini bertaraf edecektir. Örneğin aynı özel anahtar ile oluşturulmuş iki farklı imzanın kullanmış olduğu rastsal sayının aynı olması

durumunda $x = \frac{e_1 - e_2}{s_1 - s_2}$ eşitliğinden sadece imzaya bakarak özel anahtara ulaşılabilmektedir. Bu yöntem kullanılarak, Ağustos 2013'de Android işletim sisteminde bulunan bir hatadan dolayı üretilen rastsal sayıların bir noktada kümeleşmeye başlaması ile aynı rastsal sayının kullanıldığı iki farklı bitcoin imzası tespitleri yapılmış ve 53 adet bitcoin çalınmıştır. Alanında öncü olan bir oyun konsolu üreticisinin konsolunda çalışmasına izin verdiği oyunları imzalarken rastsal sayı yerine sabit bir sayı kullanmış olması, bunu tespit eden saldırganların üretici firmanın özel anahtarını açığa çıkarması ve onun adına oyunları imzalaması ile sonuçlanmıştır.

PRNG algoritmaları deterministik olarak çalışmaktadır. Rastsallık için bir entropi kaynağını temel alırlar. Linux çekirdeğine sahip çalışma ortamlarında sabit disk, klavye, ağ trafiği gibi pek çok girdi kullanılarak bir entropi havuzu oluşturulur. Girdilerin katacağı entropi oranının görece düşük olması rastsal sayı üretiminin buna bağlı olarak yavaşlamasına sebep olmaktadır. Bu çalışmada ChaCha20[14] şifreleme algoritması üzerinden blokajlı Linux entropi havuzu kullanılmış olup, sunucu sistemlerinde entropinin arttırılması için donanımsal entropi kaynaklarının kullanılabilmesine imkan tanınmıştır. Entropi kaynağından okunacak veri miktarının düşürülmesi sayesinde donanımsal entropi kaynağı olarak uygun maliyetli yarı-iletken birleşim gürültü tespit cihazlardan, fotonların kuantum özelliklerini kullanan kurumsal ürünlere kadar geniş bir yelpazede seçim yapılabilir.

3 Mimari

3.1 Mutabakat ve Ölçeklenebilirlik

Blokzincir mimarilerinde gayri merkezi olarak veri bütünlüğünü korumak üzere çeşitli mutabakat protokolleri kullanılmaktadır. Kamu erişimine açık ve dileyen her katılımcının blok onaylayabildiği platformlarda sanal olarak yaratılmış çeşitli problemlerin çözümü iş ispatı olarak istenmekte ve kötü niyetli katılımcıların geçersiz işlemler barındıran blokları zincire eklemeleri engellenmeye çalışılmaktadır. Bu durum blokzincirin işlem kapasitesini düşürmekte ve hesaplama gücünün gereksiz kullanımına sebep olmaktadır. Bir diğer yaklaşım ise blokzincir üzerindeki varlık sahiplerinin varlıklarını korumak isteyecekleri olgusuna dayalı, oylama temelli blok onay hakkı sunan mutabakat sistemleridir.

Blokzincir ağına dahil olup blokları onaylayacak katılımcıların denetlenebildiği ortamlar için daha verimli mutabakat alternatifleri geliştirilmiştir. İş yada hisse ispatı gerekmeksizin belirlenmiş katılımcıların blok onaylayabildiği, dileyen herkesin blokzincire katılabildiği mimarilere "izinli blokzincir mimarileri" adı verilmekte olup bu çalışma kapsamında kullandığımız mimarinin temelini teşkil etmektedir.

Transfer işlemlerinin onaylanması için gerekli hesaplama gücü ve işlemlerin dağıtık defterde kapladığı alan miktarı, kullanılan mutabakat yöntemi

ile birlikte ölçeklenebilirlik probleminin temelini oluşturmaktadır.

3.1.1 Performans

Blok onaylayan katılımcıların transfer işlemlerinin doğruluğunu kontrol etmek için çok sayıda kriptolojik işlem yapması gerekmektedir. Bu nedenle kullandığımız eliptik eğrinin gerektirdiği matematiksel işlevlerin, güncel bilişim sistemlerinin sunmuş olduğu yetenekler ile güvenli ve performanslı bir şekilde implemente edilebilmesi büyük önem arz etmektedir.

Bu çalışmada, parametreleri güncel işlemci mimarilerinin imkanları gözetilerek seçilen Curve25519[8] ile Edwards eliptik eğrisi kullanılmıştır. Bu eğride temsil edilen noktaların güvenli ve sıkıştırılmış olarak dağıtık deftere yazılabilmesi için Decaf[9] şemasının bir varyantı olan Ristretto[10] algoritması kullanılmıştır. Intel işlemci ailesinin AVX2 ve AVX512 özellikleri sayesinde 256 bit'lik modüler aritmatik hesaplamaları yüksek verimle çalışacak şekilde implemente edilmiştir.

Sistem seviyesinde Rust, C ve Assembly, orta katmanda Go ve son kullanıcı katmanında JavaScript kullanılmıştır.

3.1.2 Uygulamalı Bizans Hata Toleransı (*PBFT*)

Bu çalışma kapsamında sunmuş olduğumuz izinli blokzincir mimarisinde kullanılan mutabakat yöntemi *PBFT*'dir. Blok onaylayacak katılımcıların 1/3'üne kadar art niyetli olması durumunda dahi tutarlılığını yitirmeyen ve çifte harcama ataklarının önüne geçebilen DLS[6] protokolünün bir varyantı olan Tendermint[7]'i kullanıyoruz.

3.1.3 Harcanmamış İşlem Çıktısı Modeli (*UTXO*)

Çalışmamız kapsamında blokzincir üzerinde gerçekleşen transferlerin işlenebilmesi, bloklara yerleştirilmesi ve dağıtık defter üzerinde saklanabilmesi için kullanılan iki yöntemden birisi olan UTXO yöntemi kullanılmıştır. Bitcoin'in kullandığı bu model sayesinde işlemlerin doğrulukları eş zamanlı olarak kontrol edilebilmekte ve blokzincir ağının birim zamanda işleyebileceği transfer miktarı arttırılmaktadır.

Ethereum'un kullanmış olduğu bakiye takibi ve *world state* yönetimi gerek tarihçeli işlem sorgularının zor olması, gerekse dağıtık defterin son durumu üzerinde sürekli mutabakat sağlanmasını gerektirmesi nedeni ile tercih edilmemiştir. RTGS uygulamalarında ihtiyaç duyulabilecek akıllı sözleşme kurgularının UTXO programları olarak implemente edilebileceğini öngörmekteyiz.

3.2 Transfer İşleminin Yapısı

Katılımcıların ellerinde tuttukları varlıklar Pedersen taahhütleri ile ifade edilmektedir. Bu taahhütler diğer kimselere gönderilebilir ve bölünebilir niteliktedirler. Varlığın ihraç edildiği andan itibaren içerisinde bulunduğu temel form bir Pedersen taahhütü olarak $C = g^v h^r \in \mathbb{G}$ şeklinde ifade edilebilir. Bu ifade içerisinde v varlığın miktarını, r ise bu bilgiyi gizlemek için kullandığımız rastsal bir sayıyı(*blinding factor*) temsil etmektedir. C ile taahhütün karşılığı olan nokta değeri ifade edilmekte olup, g varlık cinsine göre üretilmiş ve h "H" harfinin hash değerinden eğrideki bir noktaya aktarılmış *basepoint*'e göre ayrık logaritması kimse tarafından bilinmeyen iki sabit noktadır.

Pedersen taahhütleri[15] toplamsal olarak eşgeçişgenlik (*additive homomorphism*) gösterirler.

$$C_1 = g^{v_1} h^{r_1},\ C_2 = g^{v_2} h^{r_2};\ C_1 + C_2 = g^{v_1+v_2}\ h^{r_1+r_2}$$

Sahip olduğu varlığı bir başkasına göndermek isteyen katılımcı, göndereceği varlığın ifade edildiği taahhütte kullanılan v ve r değerlerini bilmelidir.

Bir transfer işlemi girdi taahhütleri ve çıktı taahhütlerinden oluşmaktadır. Girdi taahhütlerinin ifade ettiği varlık miktarının toplamı, çıktı taahhütlerinin ifade ettiği varlık miktarının toplamına eşit olmalıdır. Blok onaylayan katılımcılar bu eşitiğin kontrolünü miktar bilgisini görmeksizin, sıfır bilgi ispatı ile yapmaktadırlar.

$$\sum_{i=0}^{n} C_{input_i} - \sum_{i=0}^{m} C_{output_i} = \sum_{i=0}^{n} h^r_{input_i} - \sum_{i=0}^{m} h^r_{output_i}$$

Transferi gerçekleştirecek olan kullanıcı kalan h^r'nin ayrık logaritmasını ispat etmelidir. Bunun ispatı için Schnorr[11] protokolünü kullanıyoruz. Etkileşimsiz sıfır bilgi ispatı için Fiat–Shamir heuristic[12] modelini kullanıyoruz.

$$\begin{aligned}
&x = \textstyle\sum_{i=0}^{n} r_{input_i} - \sum_{i=0}^{m} r_{output_i} \\
&r = \xleftarrow{r} \in \mathbb{Z}_p \\
&e = H(\, g^r \parallel g^x) \\
&s = r - x \cdot e \bmod n \\
&\mathcal{P} \rightarrow \mathcal{V} : s, e
\end{aligned}$$

Transfer işleminin yer aldığı bloku onaylayacak katılımcılar ayrık logaritma ispatını doğrularlar.

$$p = \sum_{i=0}^{n} h^r_{input_i} - \sum_{i=0}^{m} h^r_{output_i} \rightarrow e \stackrel{?}{=} H(\, g^s p^e \parallel p)$$

Bu adımdan sonra transferin girdi ve çıktılarının toplamlarının sıfır olduğu ve göndericinin sahip olduğundan daha fazla varlık transferi yapmadığı kanıtlanmış olur.

Sistem genelinde hash yöntemi olarak 256 bit SHA3 kullanılmaktadır. Random oracle modeli için SHA3 XOF ve eliptik eğride rastgele nokta seçmek için 512 bit SHA3 ile Elligator[13] algoritmaları kullanılmaktadır.

3.3 İşlemlerde Taşma Kontrolü

Varlık transferlerinde gönderilen Pedersen taahhütlerinde ifade edilen varlık miktarı Ristretto şemasında kullanılan *prime order*[1] üst limiti ile sınırlıdır. Tamsayılar üzerinde yapılan tüm aritmatik işlemler bu moduloda gerçekleştirilir.

Transfer edilen varlık taahhütlerinin toplanarak girdi taahhütleri ile karşılaştırılması aşamasında toplanan taahhütlerin ifade ettiği miktar bilgisinin *prime order*'ı aşması durumunda modüler çalışmadan dolayı sıfırdan devam edecektir. Bu durum kötü niyetli katılımcıların sahip olduklarından daha fazla varlığı transfer edebilmelerine olanak tanır.

Bu sorunu ortadan kaldırmak için blok onaylayan katılımcıların her bir Pedersen taahhütü içerisinde ifade edilen miktar bilgisinin toplamının *prime order*'ı geçmediğini, ifade edilen miktar bilgisini görmeden onaylaması gerekmektedir.

Bu çalışmada gönderilecek azami varlık miktarının $2^{64} - 1$ birim olabileceğini varsaydık. Bu varsayım ile *prime order*'a ulaşabilmek için 3.9×10^{56} adet taahhütün aynı transfer içerisinde kullanılması gerektiğini hesaplayabiliriz. Böylece blok onaylayan katılımcıların, her bir taahhütün ifade ettiği miktar bilgisinin 64 bit ile sınırlı olduğunu doğrulaması ile art niyetli katılımcıların sahip olduklarından daha fazla varlık gönderebilmesinin önüne geçilmiştir.

3.3.1 Alternatif Yöntemler

Taahhütlerin ifade ettiği değerin taşma olmaksızın 64 bit ile ifade edilebildiğinin kanıtının sıfır bilgi ispatı ile oluşturmanın ve doğrulamanın alternatif yöntemleri mevcuttur.

Blokzincir mimarisinde kullanmış olduğumuz mutabakat özelliği sayesinde *honest verifier non-interactive zero-knowledge proofs* yöntemlerini kullanabilmekteyiz. Bu noktada *CRS* ve $\sum$ protokolü temelli iki farklı yaklaşım kullanılmaktadır. Sigma protokolünü temel alan algoritmalar Fiat–Shamir heuristic[12] yaklaşımı ile *non-interactive* biçimde implemente edilebilmektedir. Ortak referans değerini (*CRS*) temel alan algoritmalar ise güvenli kurulum fazına ihtiyaç duyarlar. Bu durum genellikle güvenilir bir üçüncü parti gereksinimi ortaya çıkartır. Bu çalışmada kullanmış olduğumuz etkileşimsiz sıfır bilgi ispatı teknikleri sigma protokolünü temel almaktadır ve güvenilir üçüncü parti yada kurulum fazına ihtiyaç duymamaktadır.

[1] $2^{252} + 27742317777372353535851937790883648493$

Halka İmzaları (*Ring Signatures*) Aralık ispatı için hali hazırda Monero gibi kriptopara platformları tarafından kullanılan, taahhütün ifade ettiği miktardaki her bir hanenin alabileceği tüm değerleri temsilen oluşturulan imza halkalarını temel alırlar. Bu yöntemde oluşturulan kanıtların boyutu, 32 bit ile sınırlandırılmış ortalama bir taahhüt için dahi, 2-4 KB aralığında olmaktadır. UTXO yapısı gereği en küçük transfer işleminin bir adet girdi, bir adet çıktı ve genelde bir adet iade çıktı taahhütü bulunur. Bu durumda en küçük transfer için dahi 4-8 KB aralığında bir kanıt oluşturulması ve dağıtık defterde saklanması gerekir. Pratikte kullanılan bir yöntem olmasına karşın son derece verimsiz bir seçenektir.

zk-SNARKs CRS ve *bilinear pairing-based* kriptografi temellerinde geliştirilen ve zCash kriptopara platformu tarafından kullanılan sıfır bilgi ispatı yöntemidir. Güvenilir kurulum gerektirmesi en büyük zayıflığı olup, standart model dışında yeni sayılabilecek kriptolojik kabüllerin bulunduğu şifreleme teknikleri kullanmaktadır. Oluşturulan kanıtların 500-600 byte gibi küçük ve sabit büyüklükte olması ve doğrulayıcıların çalışma zamanının doğrusal olması önemli avantajlarındandır.

zk-STARKs[16] Güvenilir kurulum gerektirmeyen ve kuantum bilgisayarlara karşı dirençli bir etkileşimsiz sıfır bilgi ispatı yöntemidir. Sabit ve küçük boyutlu kanıtlara sahip olması, kanıt doğrulama algoritmasının doğrusal zamanda çalışması önemli avantajlarıdır. Mevcut hali ile doğrulama aşamasında pratikte uygulanamayacak kadar bellek gerektirmesi ve uzun çalışma zamanında tamamlanabilen kanıt hazırlama algoritması nedeni ile şimdilik pratik bir kullanımı bulunmamaktadır. Gelecekte kuantum bilgisayarlara karşı alınacak önlemler içerisinde bu seçeneği değerlendirmeyi planlıyoruz.

Bulletproofs[17] Güvenilir kurulum gerektirmeyen ve standart model içerisinde kriptolojik kabüller ile geliştirilmiş ve çalışmamızda da kullandığımız sıfır bilgi ispatı algoritmasıdır. Sigma protokolünü temel alır ve random oracle model üzerinden etkileşimsiz olarak implemente edilebilir. Oluşturulan kanıtlar $(2 \times log_2(n) + 4)$ nokta ve 5 adet tamsayı değer içermektedir. Curve25519 ve Ristretto şeması ile bu verinin 64 bit üzerinden hesaplanan bir kanıt için 672 byte olduğunu söyleyebiliriz.

Kanıt boyutunun doğrusal arttığı, oluşturma ve doğrulama algoritmalarının doğrusal zamanda çalıştığı bu yöntemin en büyük avantajı oluşturulan kanıtların birleşebilmesi özelliğidir. Bu sayede çok sayıdaki taahhüt için tek bir kanıt oluşturulabilmekte, saklanabilmekte ve doğrulanabilmektedir. Bu özelliği aynı zamanda güvenilir çok partili hesaplama (*secure multi-party computation*) imkanı da sunmaktadır. Bu özellikleri ve aritmatik devreleri

kullanarak ileride mahremiyet odaklı akıllı sözleşmeleri geliştirmeyi planlamaktayız.

3.3.2 Kullanılan Yöntem

Pedersen taahhütünün ifade ettiği varlık miktarı bilgisinin taşma olmaksızın n bit ile ifade edilebildiği aşağıdaki ilişki üzerinden gösterilebilir;

$$\{(g,\ h,\ C \in \mathbb{G},\ n;\ v,\ r\ \in \mathbb{Z}_p) :\ C = g^v h^r\ \ \wedge v \in [0,\ 2^n - 1]\}$$

Bu ilişkinin etkileşimsiz sıfır bilgi ispatı ile kanıtlanması ve doğrulanması için aşağıdaki eşitliklerin sıfır bilgi ispatlarını sunuyoruz;

$$\langle \mathbf{a}_L, \mathbf{2}^n \rangle = v\ \ \wedge\ \ \mathbf{a}_L \circ \mathbf{a}_R = \mathbf{0}^n\ \ \wedge\ \ \mathbf{a}_R = \mathbf{a}_L - \mathbf{1}^n$$

Bu eşitliklerin ispatı için random oracle'dan z ve y adında iki değer alıyoruz;

$$\begin{aligned}
&\mathbf{a}_L = \xleftarrow{0,1} v\ \in\ \{0,1\}^n \\
&\langle \mathbf{a}_L, \mathbf{2}^n \rangle = v\ \in\ \left[0,\ 2^{n-1}\right] \\
&\mathbf{a}_R = \mathbf{a}_L - \mathbf{1}^n\ \in\ \mathbb{Z}_p^n \\
&\alpha = \xleftarrow{r} \in \mathbb{Z}_p \\
&A = h^\alpha\ \mathbf{g}^{\mathbf{a_L}}\ \mathbf{h}^{\mathbf{a_R}}\ \in\ \mathbb{G} \\
&\mathbf{s}_L = \xleftarrow{r} \in \mathbb{Z}_p^n \\
&\mathbf{s}_R = \xleftarrow{r} \in \mathbb{Z}_p^n \\
&\rho = \xleftarrow{r} \in \mathbb{Z}_p \\
&S = h^\alpha\ \mathbf{g}^{\mathbf{s_L}}\ \mathbf{h}^{\mathbf{s_R}}\ \in\ \mathbb{G} \\
&y = H(\ A \parallel S \parallel C\ )\ \in \mathbb{Z}_p \\
&z = H(\ y\ )\ \in \mathbb{Z}_p
\end{aligned}$$

Elde edilen y ve z değerleri ile sıfır bilgi ispatını aşağıdaki şekilde sunabiliriz;

$$\langle \mathbf{a}_L - z\cdot\mathbf{1}^n,\ \mathbf{y}^n \circ (\mathbf{a}_R + z\cdot\mathbf{1}^n) + z^2\cdot\mathbf{2}^n \rangle = z^2\cdot v + \left(\left(z - z^2\right)\cdot\langle \mathbf{1}^n,\ \mathbf{y}^n\rangle - z^3\cdot\langle \mathbf{1}^n, \mathbf{2}^n\rangle\right) \in \mathbb{Z}_p$$

Değerleri gizlemek için kullandığımız $\mathbf{s}_L$ ve $\mathbf{s}_R$ vektörleri için iki adet vektör polinomu hazırlıyoruz;

$$\begin{aligned}
&l(X) = (\ \mathbf{a}_L - z\cdot\mathbf{1}^n\ )\ + (\ \mathbf{s}_L\cdot X\ ) \\
&r(X) = (\ \mathbf{y}^n(\mathbf{a}_R + 1^n) + (z^2\cdot\mathbf{2}^n)\ )\ + (\ \mathbf{y}^n\cdot\mathbf{s}_R\cdot X\ )
\end{aligned}$$

Vektör polinomlarının çarpımından iki dereceli bir polinom hazırlıyoruz;

$$t(X) = \langle l(X), r(X)\rangle$$

Her iki derecesi için rastsal bir sayı üreterek Pedersen taahhütünü hazırlıyoruz. Bu taahhütleri kullanarak random oracle'dan x değerini alıyoruz ve

rastsal değerler ile maskelenmiş $\mathbf{a}_L, \mathbf{a}_R$ değerleri için taahhütler hazırlıyoruz.

$$
\begin{aligned}
&\tau_1 =\xleftarrow{r}\in \mathbb{Z}_p \\
&\tau_2 =\xleftarrow{r}\in \mathbb{Z}_p \\
&T_1 = g^{t_1}h^{\tau_1} \\
&T_2 = g^{t2}h^{\tau_2} \\
&x = H(\ T_1 \parallel T_2\ ) \ \in \mathbb{Z}_p \\
&\mathbf{l} = l(x) \ = \ (\ \mathbf{a}_L - z \cdot \mathbf{1}^n\ ) \ + (\ \mathbf{s}_L \cdot x\ ) \in \mathbb{Z}_p \\
&\mathbf{r} = r(x) \ = \ (\ \mathbf{y}^n(\mathbf{a}_R + 1^n) + (z^2 \cdot \mathbf{2}^n)\ ) \ + (\ \mathbf{y}^n \cdot \mathbf{s}_R \cdot x\ ) \in \mathbb{Z}_p \\
&\hat{t} = \langle l, r\rangle \in \mathbb{Z}_p \\
&\tau_x = \tau_2 \cdot x^2 + \tau_1 \cdot x + z^2 \cdot \gamma \in \mathbb{Z}_p \\
&\mu = \alpha + \rho \cdot x \in \mathbb{Z}_p
\end{aligned}
$$

Elde ettiğimiz değerleri vektör çarpımının sıfır bilgi ispatını hazırlamak için kullanıyoruz;

$$
\begin{aligned}
&u = H(\ \tau_x \parallel \mu \parallel \hat{t}\ ) \ \in \mathbb{Z}_p \\
&p_{ipp} = IPP\left(\mathbf{g}, \mathbf{h}^{-y^n},\ \mathbf{l},\ \mathbf{r},\ g^u\right)
\end{aligned}
$$

Transfer işlemini doğrulayacak olan katılımcıların onaylayabilmesi için aralık kanıtına dair aşağıdaki bilgileri iletiyoruz;

$$\mathcal{P} \rightarrow \mathcal{V} : A, S, T_1, T_2, \tau_x,\ \mu, \hat{t}, p_{ipp}$$

Aşağıda belirtilen yöntem ile bu ispatın geçerli olduğunu etkileşimsiz olarak doğrulayabiliyoruz;

$$
\begin{aligned}
&x = H(\ T_1 \parallel T_2\ ) \ \in \mathbb{Z}_p \\
&y = H(\ A \parallel S \parallel C\ ) \ \in \mathbb{Z}_p \\
&z = H(\ y\ ) \ \in \mathbb{Z}_p \\
&l = g^{\hat{t}}h^{\tau_x} \\
&r = T1^x + T2^{x^2} + C^{z^2} + g^{(\mathbf{y}^n \cdot z - z^2) - (z^3 \cdot 2^n)}
\end{aligned}
$$

Hesaplamış olduğumuz l ve r değerleri ile ispatın ilgili taahhüt için yapılıp yapılmadığını doğruluyoruz;

$$
\begin{aligned}
&l \stackrel{?}{=} r \\
&u = H(\ \tau_x \parallel \mu \parallel \hat{t}\ ) \ \in \mathbb{Z}_p \\
&P = A + S^x + \mathbf{g}^{n^{-z}} + \mathbf{h}^{-y^n \mathbf{y}^n \cdot z + \mathbf{2}^n} + g^{u^{\hat{t}}} - h^{\mu}
\end{aligned}
$$

Son aşamada vektör çarpım ispatını test ediyoruz;

$$Valid \ \stackrel{?}{=} IPP_{verify}\left(\mathbf{g}, \mathbf{h}^{-y^n},\ u,\ P,\ p_{ipp}\right)$$

Vektör çarpımı için sıfır bilgi ispatında ve doğrulmasında aşağıdaki algoritmayı kullanıyoruz ($IPP(..)$);

$input$:

$\boldsymbol{g}, \boldsymbol{h} \in \mathbb{G}^n$

$\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}_p^n$

$P,\ u \in \mathbb{G}$

$n' = sizeof(\ \boldsymbol{g} \mid \boldsymbol{h} \mid \boldsymbol{a} \mid \boldsymbol{b}\ )$

$loop\ (\ n' \neq 1\ )$:

$$n' = \frac{n'}{2}$$

$$L = \sum_{i=0}^{n'} \boldsymbol{g}_{n'+i}^{\boldsymbol{a}_i}\ \boldsymbol{h}_i^{\boldsymbol{b}_{n'+i}}\ u^{\sum_{i=0}^{n'} \boldsymbol{a}_{n'+i}\ \boldsymbol{b}_i \in \mathbb{Z}_p} \in \mathbb{G} \rightarrow \boldsymbol{L}^*$$

$$R = \sum_{i=0}^{n'} \boldsymbol{g}_i^{\boldsymbol{a}_{n'+i}}\ \boldsymbol{h}_{n'+i}^{\boldsymbol{b}_i}\ u^{\sum_{i=0}^{n'} \boldsymbol{a}_i\ \boldsymbol{b}_{n'+i} \in \mathbb{Z}_p} \in \mathbb{G} \rightarrow \boldsymbol{R}^*$$

$$x = H(\ L \parallel R\ ) \in \mathbb{Z}_p$$

$$\boldsymbol{g}' = \overleftarrow{\underset{i=0}{n'}}\ \boldsymbol{g}_i{}^{x^{-1}}\ \boldsymbol{g}_{n'+i}^{x} \in \mathbb{G}^{n'}$$

$$\boldsymbol{h}' = \overleftarrow{\underset{i=0}{n'}}\ \boldsymbol{h}_i{}^{x}\ \boldsymbol{h}_{n'+i}^{x^{-1}} \in \mathbb{G}^{n'}$$

$$P' = L^{x^2}\ R^{x^{-2}}\ P \in \mathbb{G}$$

$$\boldsymbol{a}' = \overleftarrow{\underset{i=0}{n'}}\ \boldsymbol{a}_i\ x + \boldsymbol{a}_{n'+1}\ x^{-1} \in \mathbb{Z}_p^{n'}$$

$$\boldsymbol{b}' = \overleftarrow{\underset{i=0}{n'}}\ \boldsymbol{b}_i\ x + \boldsymbol{b}_{n'+1}\ x^{-1} \in \mathbb{Z}_p^{n'}$$

$output$:

$\boldsymbol{a}'_0, \boldsymbol{b}'_0,\ \boldsymbol{L},\ \boldsymbol{R} \in \mathbb{G}$

3.4 Denetçi Erişimi

Pedersen taahhütleri ile temsil edilen varlıkların denetçi rolüne sahip katılımcılar tarafından görülebilmesi için denetçi katılımcının açık anahtarı T'nin kullanıldığı bir kanıtlanabilir giz paylaşım (*publicly verifiable secret sharing*) şeması (*PVSS[18]*) uygulanmıştır. Bir yada daha çok denetçinin bulunabilmesi, belirli bir eşik değerinin tanımlanması sureti ile en az tanımlanmış eşik değeri sayısınca denetçinin bir araya gelerek gizli anahtara ulaşması gibi senaryoların uygulanabilmesi bu algoritma ile mümkün hale gelmiştir.

Taşma kanıtları hazırlanırken aynı zamanda kanıt için kullanılan varlık miktarı bilgisi denetçinin açabileceği şekilde şifrelenir ve bu şema üzerinden şifrelemede kullanılan anahtar paylaşılır. Ayrık logaritma eşitliği ispatı ile anahtara denetçinin erişebileceği de garanti altına alınmış olur. Bu noktada

kullanılan simetrik şifreleme algoritmasının bir aritmatik devre üzerinden çalıştırılması ve bu işlem içinde kanıt üretilmesi planlanmaktadır. r değerini transferde kullanılan *nonce* olarak kabul ettiğimizde;

$$
\begin{aligned}
&X = g^r \\
&P_h = h^r \\
&P_T = T^r \\
&k = H(\ X\ ) \\
&\alpha = Enc(k,\ v) \\
&\mathcal{P} \rightarrow \mathcal{V} : \alpha, P_h, P_T
\end{aligned}
$$

P_h ve P_T'nin eş ayrık logaritmaya sahip olduğunu etkileşimsiz sıfır bilgi ispatı ile kanıtlamak için $g = T, x = r$ olarak;

$$
\begin{aligned}
&xG = g^x \\
&xH = h^x \\
&v =\xleftarrow{r} \in \mathbb{Z}_p^n \\
&vG = g^v \\
&vH = h^v \\
&c = H(\ xG \parallel xH \parallel vG \parallel vH) \\
&r = v - c \cdot x \\
&\mathcal{P} \rightarrow \mathcal{V} : vG,\ vH,\ c,\ r
\end{aligned}
$$

Logaritma eşitliğini doğrulamak için $xG = P_T, xH = P_h$ olarak;

$$g^r + xG^c \stackrel{?}{=} h^r + xH^c$$

3.5 Adreslerin Gizliliği

Katılımcıların varlık transferi yaparken kullanmış oldukları tüm adresler rastsal olarak üretilmektedir. Her bir katılımcı kendisi için bir harcama anahtar çifti üretir ve açık harcama anahtarını denetçi ile paylaşır. Denetçi bu açık anahtara karşılık bir gizli tarama anahtarı üreterek katılımcıya teslim eder. Katılımcı kendisine varlık transfer etmek isteyen partilere açık tarama anahtarı ve açık harcama anahtarından oluşan adresini verir. Gönderici kimse rastsal bir sayı r_A seçer ve kendisine saklar. $R_A = g^r$ ve $T_A = g^{H(r_A \| V_B)}$ değerlerini hesaplayarak transferi T_A adresine gönderir. Bu hesapta kullanılan V_B değeri alıcının tarama anahtarıdır. Gönderici transferine aynı zamanda R_A değerini de iletir.

Alıcı kimse $T'_A = g^{H(v_B \| R_A)}$ değerini hesaplar ve $T'_A \stackrel{?}{=} T_A$ eşitliğini test eder. Burada kullanılan v_B alıcının gizli tarama anahtarıdır. Eşitliğe uyan transferler ilgili kullanıcı için gerçekleşmiş demektir ve bu rastsal üretilen adresin özel anahtarını sahip olduğu özel harcama anahtarı ile şu şekilde hesaplar: $t'_A = H(v_B \| R_A) + s_b$.

Denetçi rolünde bulunan katılımcı, transferlerin hedef adreslerinde açık harcama ve açık tarama anahtarları ile hangi transferin kime yapıldığını tespit eder.

Kaynaklar

[1] Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), American National Standard X9.62.

[2] Andrew M. Odlyzko, The future of integer factorization. http://www.dtc.umn.edu/~odlyzko/doc/future.of.factoring.pdf

[3] ECRYPT, D5.4: Algorithms, Key Size and Protocols Report (2018). http://www.ecrypt.eu.org/csa/documents/D5.4-FinalAlgKeySizeProt.pdf

[4] Bristlecone Quantum Processor. https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html

[5] T. Ruffing, Switch Commitments: A Safety Switch for Confidential Transactions https://eprint.iacr.org/2017/237.pdf

[6] C. Dwork, N. Lynch, and L. Stockmeyer: Consensus in the presence of partial synchrony https://groups.csail.mit.edu/tds/papers/Lynch/jacm88.pdf

[7] E. Buchman: Byzantine Fault Tolerance in the Age of Blockchains https://atrium.lib.uoguelph.ca/xmlui/bitstream/handle/10214/9769/Buchman_Ethan_201606_MAsc.pdf

[8] Daniel J. Bernstein: Curve25519: new Diffie-Hellman speed records. https://cr.yp.to/ecdh/curve25519-20060209.pdf

[9] Mike Hamburg: Decaf: Eliminating cofactors through point compression https://www.shiftleft.org/papers/decaf/decaf.pdf

[10] Ristretto: A technique for constructing prime order elliptic curve groups with non-malleable encodings. https://ristretto.group/

[11] C.P. Schnorr: Efficient identification and signatures for smart cards. https://pdfs.semanticscholar.org/8d69/c06d48b618a090dd19185aea7a13def894a5.pdf

[12] Amos Fiat and Adi Shamir: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. CRYPTO 1986: pp. 186-194

[13] Daniel J. Bernstein, Mike Hamburg, Anna Krasnova, Tanja Lange. "Elligator: Elliptic-curve points indistinguishable from uniform random strings." https://elligator.cr.yp.to/elligator-20130828.pdf

[14] Daniel J. Bernstein. "ChaCha, a variant of Salsa20." Workshop Record of SASC 2008: The State of the Art of Stream Ciphers. https://cr.yp.to/chacha/chacha-20080128.pdf

[15] Torben Pryds Pedersen: Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing https://link.springer.com/content/pdf/10.1007%2F3-540-46766-1_9.pdf

[16] E. Ben-Sasson, I. Bentov, Y. Horesh, M. Riabzev: Scalable, transparent, and post-quantum secure computational integrity https://eprint.iacr.org/2018/046.pdf

[17] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, G. Maxwell: Bulletproofs: Short Proofs for Confidential Transactions and More https://eprint.iacr.org/2017/1066.pdf

[18] Berry Schoenmakers: A Simple Publicly Verifiable Secret Sharing Scheme and its Application to Electronic Voting https://www.win.tue.nl/~berry/papers/crypto99.pdf