OMÜ Mühendislik Bilimleri ve Teknolojisi Dergisi,
OMU Journal of Engineering Sciences and Technology
e-ISSN: 2791-8858 OMUJEST, Mart 2024, 4(1): 15-34

Kuantum Sonrası Güvenli Dijital Kripto Cüzdan

Post-Quantum Secure Digital Crypto Wallet

Ecmel ALBAYRAK[1], Sedat AKLEYLEK[2]

[1]Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Ondokuz Mayıs Üniversitesi, Samsun
· **ecmel.kaytazoglu@bil.omu.edu.tr** · ORCİD > 0000-0002-6440-400X

[2]Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Ondokuz Mayıs Üniversitesi, Samsun
· **akleylek@gmail.com** · ORCİD > 0000-0001-7005-6489

Makale Bilgisi/Article Information

Makale Türü/Article Types: Araştırma Makalesi/Research Article
Geliş Tarihi/Received: 08 Eylül/September 2023
Kabul Tarihi/Accepted: 21 Eylül/September 2023
Yıl/Year: 2024 | **Cilt–Volume:** 4 | **Sayı–Issue:** 1 | **Sayfa/Pages:** 15-34

Atıf/Cite as: Albayrak, E., Akleylek, S. "Kuantum Sonrası Güvenli Dijital Kripto Cüzdan" Ondokuz Mayıs Üniversitesi Mühendislik Bilimleri ve Teknolojisi Dergisi 4(1), Mart 2024: 15-34.

Sorumlu Yazar/Corresponding Author: Ecmel ALBAYRAK

KUANTUM SONRASI GÜVENLİ DİJİTAL KRİPTO CÜZDAN

ÖZ

Blokzincir, merkeziyetsiz bir ağ üzerinde çalışan, bir uzlaşma algoritması tarafından yönetilen, değiştirilemez bir dijital kayıt defteridir. Blokzincirde kullanıcıların gerçek kimlik bilgileri yerine kriptografik olarak dijital kripto cüzdanlarda üretilen cüzdan anahtarları ve cüzdan adresleri kişisel tanımlayıcı olarak kullanılır. Dijital kripto cüzdanlar blokzincirlerinden ayrı olarak geliştirilen uygulamalardır. Ancak onlar olmadan, blokzincirinde kullanıcıyı temsil eden herhangi birşey olmadığı için transfer işlemlerinin gerçekleştirilmesi, akıllı sözleşme uygulamalarının çalıştırılması gibi blokzincir ile etkileşime girilebilecek herhangi bir işlemin yapılması mümkün değildir. Günümüzde dijital kripto cüzdan uygulamalarında anahtar üretim sürecinde açık anahtarlı eliptik eğri dijital imzalama algoritması (ECDSA) kullanılmaktadır. Bu algoritmanın güvenliği eliptik eğri üzerindeki ayrık logaritma probleminin zorluğuna dayanmaktadır. 1994 yılında Shor'un önerdiği algoritma ile açık anahtarlı kriptosistemlerin dayandığı zor problemlerin kuantum bilgisayarlar varlığında polinom zamanda çözülebileceği belirtilmiştir. Bu durum, kuantum bilgisayarlar varlığında açık anahtar kriptografisi kullanılarak oluşturulan bütün sistemler gibi ECDSA kullanılarak oluşturulan kripto cüzdan uygulamalarının da güvenliğinin sağlanamayacağı anlamına gelmektedir. Kuantum sonrası kriptosistemlerin standartlaştırılması ihtiyacından dolayı NIST 2016 yılında bir çağrıda bulunmuştur. Bu çağrı kapsamında belirli aşamalardan geçerek dijital imzalama standardı olarak kafes tabanlı Crystals-Dilithium ve Falcon algoritmaları seçilmiştir. Bu çalışmada Crystals-Dilithium dijital imzalama algoritmasının kripto cüzdan anahtar üretim aşamasında kullanımı sağlanılarak Bitcoin ve Ethereum blokzincirleri için kuantum sonrası güvenli dijital kripto cüzdan önerisinde bulunulmuş ve bu uygulamalar Rust programlama dili ile gerçeklenmiştir. Klasik ve kuantum sonrası için geliştirilen cüzdan uygulamalarının cüzdan bilgilerinin ortalama oluşturulma süresi belirtilmiştir. Ayrıca, klasik ve kuantum sonrası blokzincir prototipi oluşturularak, çalışma kapsamında geliştirilen dijital kripto cüzdan uygulamalarının bu prototipler üzerinde işlem imzalama ve doğrulama süreçlerinin ortalama gerçekleşme süreleri belirtilmiştir.

Anahtar Kelimeler: Blokzincir, Dijital Kripto Cüzdan, Kuantum Sonrası Kriptografi.

❋❋❋

POST-QUANTUM SECURE DIGITAL CRYPTO WALLET

ABSTRACT

Blockchain is an immutable digital ledger that runs on a decentralized network, managed by a consensus algorithm. In blockchain, wallet keys and wallet addresses generated cryptographically in digital crypto wallets are used as personal identifiers instead of users' real identity information. Digital crypto wallets are applications developed separately from blockchains. However, without them, it is not possible to carry out any transaction that can interact with the blockchain, such as transfer transactions, running smart contract applications, since there is nothing in the blockchain that represents the user. Today, public key elliptic curve digital signature algorithm (ECDSA) is used in digital crypto wallet applications. The security of this algorithm is based on the difficulty of the discrete logarithm problem on the elliptic curve. With the algorithm proposed by Shor in 1994, it was stated that the difficult problems on which public key cryptography systems are based can be solved in polynomial time in the presence of quantum computers. This means that in the presence of quantum computers, crypto wallet applications created using ECDSA, like all systems created using public keys, cannot be secured. Due to the need to standardize post-quantum cryptosystems, NIST made a call in 2016. Within the scope of this call, lattice-based Crystals-Dilithium and Falcon algorithms were chosen as the digital signature standard by going through certain stages. In this study, a post-quantum secure digital crypto wallet is proposed for Bitcoin and Ethereum blockchains by using the Crystals-Dilithium digital signature algorithm in the crypto wallet key generation stage, and these applications are implemented with the Rust programming language. The average creation time of wallet information for classical and post-quantum wallet applications is indicated. In addition, classical and post-quantum blockchain prototypes are created and the average execution times of transaction signature and verification processes on these prototypes of digital crypto wallet applications developed within the scope of the study are stated.

Keywords: Blockchain, Digital Crypto Wallet, Post-Quantum Cryptography.

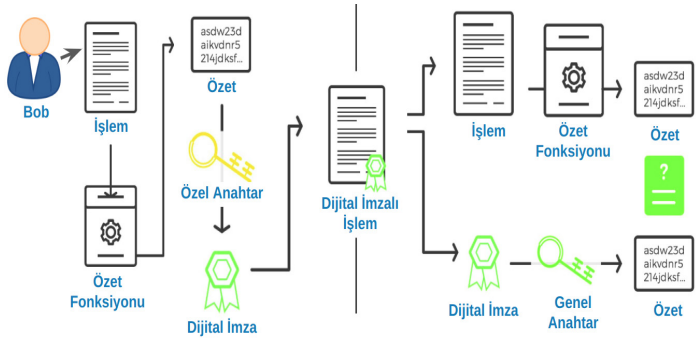
❋❋❋

1. GİRİŞ

Bitcoin tanıtıldığı günden beri blokzincir teknolojisinin gelişimine ve uygulamalarına olan ilgi büyük ölçüde artarak devam etmektedir. 2013 yılı itibariyle Ethereum blokzincirinde akıllı sözleşmelerin kullanılmaya başlanmasıyla, blokzincirler artık sadece bir değer transfer aracı olarak değil, bunun çok daha ötesinde programlanabilir bir yapıya kavuşmuştur [1]. Akıllı sözleşmeler sayesinde blok-

zincir üzerinde merkeziyetsiz uygulamaların (decentralized application - DApp) oluşturulup kullanılmasının önü açılmıştır.

Blokzincir ağında transfer edilecek verilere işlem (transaction) denilmektedir. İşlemlerin içerisinde genellikle gönderenin ve alıcının cüzdan adresi, gönderilecek miktar ve işlem ücreti bilgileri bulunmaktadır. Bu işlemler ağa gönderilmeden önce işlemin gönderen kişinin kendisine ait olduğunun ispatının yapılabilmesi için gönderici tarafından imzalanması gerekmektedir [2, 3]. Şekil 1'de işlem imzalama ve doğrulama süreçleri gösterilmektedir.



Şekil 1. İşlem imzalama ve doğrulama süreçleri

İmzalama sürecinde, oluşturulan işlem ilk önce özet fonksiyonundan geçirilir ve elde edilen özet değer kullanıcının özel anahtarı kullanılarak dijital imza oluşturulur. Doğrulama sürecinde ise doğrulayıcı düğümler tarafından işlemin orjinal hali özet fonksiyonundan geçirilerek bir özet değer elde edilir. Ardından işlem için oluşturulan dijital imza gönderenin açık anahtarı kullanılarak çözümlenir ve yine bir özet değer elde edilir. Elde edilen bu iki özet değer karşılaştırıldığında işlemler aynı ise işlem doğrulanır ve bekleyen işlemler havuzuna aktarılır [3].

İşlemlerin imzalanmasında, doğrulanmasında dijital kripto cüzdan uygulamalarında oluşturulan açık ve özel anahtarlar kullanılmaktadır. Dolayısıyla dijital kripto cüzdan uygulamaları blokzincirinden ayrı olarak geliştirilebilen uygulamalar olsalar da onlar olmadan blokzincirinde herhangi bir işlem yapılması mümkün değildir.

Günümüzde dijital kripto cüzdanların anahtar ve adres oluşturma süreçleri farklı olabilse de temelde hepsi asimetrik (açık anahtarlı) kriptografi kullanılarak oluşturulmaktadır. Günümüzde kullanılan açık anahtarlı kriptosistemler bazı matematiksel problemlerin zorluğu temel alınarak oluşturulmuştur. Örneğin, Rivest - Shamir - Adleman (RSA) şifreleme sistemi çarpanlara ayırmanın zorluğuna, dijital imzalama algoritması (Digital Signature Algorithm - DSA) ve Diffie-Hellman

(DH) anahtar değişimi, sonlu cisimlerde ayrık logaritma probleminin zorluğuna, eliptik eğri Diffie-Hellman (Elliptic Curve Diffie–Hellman - ECDH) anahtar değişimi ve eliptik eğri dijital imzalama algoritması (Elliptic Curve Digital Signature Algorithm - ECDSA) ise eliptik eğri üzerindeki ayrık logaritma probleminin zorluğuna dayanmaktadır [4, 5].

Peter Shor tarafından 1994 yılında önerilen bir algoritma ile açık anahtarlı kriptosistemlerin güvenliğini sağlayan çözülmesi zor ayrık logaritma ve çarpanlarına ayırma problemlerinin kuantum bilgisayarlarda polinom zamanda çözülebileceği belirtilmiştir [6]. Bu nedenle günümüzde kullanılmakta olan açık anahtarlı RSA, DSA, ECDSA gibi mevcut kriptosistemlerin kuantum bilgisayarlar varlığında güvensiz sistemler olacağı belirtilmektedir [7]. Çizelge 1'de gösterildiği üzere kuantum bilgisayarlar varlığında açık anahtarlı kriptosistemlerin güvenliğinin sağlanamayacağı görülmektedir.

Çizelge 1. Asimetrik kriptosistemler için klasik ve kuantum bilgisayarlarda güvenlik seviyelerinin karşılaştırılması [8]

Kriptosistem	Tip	Güvenlik Seviyesi	Kuantum Bilgisayar ile Güvenlik Seviyesi
RSA-3072	Asimetrik	128-bit	0-bit
DSA-3072	Asimetrik	128-bit	0-bit
ECC-256	Asimetrik	128-bit	0-bit
ECC-384	Asimetrik	128-bit	0-bit

Kuantum bilgisayarlar varlığında açık anahtarlı kriptosistemlerin güvenliğinin sağlanamayacak olması kuantum sonrası için güvenli kriptosistemlerin oluşturulma ihtiyacını ortaya çıkarmıştır. Bu sebeple, NIST 2016 yılında kuantum sonrası kriptosistemlerin standartlaştırılması için bir çağrıda bulunmuştur. Şifreleme, anahtar paketleme ve dijital imzalama kategorilerinde ele alınan algoritmalar; kafes tabanlı, özet tabanlı, kod tabanlı, çok değişkenli polinomlar tabanlı ve izojeni tabanlı gibi matematiksel problemlere dayalı kriptosistemler altında incelenmiştir [9]. Kafes tabanlı algoritmalardan ilk turda toplam 28 algoritma, ikinci turda toplam 12 algoritma, üçüncü turda beş adet finalist ve iki adet yedek aday olmak üzere toplam yedi algoritma seçilmiştir [10]. Üçüncü turun sonunda seçilen algoritmalar listesinde yer alan algoritmalar Çizelge 2'de gösterilmektedir.

Çizelge 2. NIST kuantum sonrası kriptosistemleri standartlaştırma sürecinde üçüncü turda seçilen algoritmalar listesi [11]

Kriptosistem	Şifreleme ve Anahtar Paketleme	Dijital İmzalama
Kafes	Crystals-Kyber	Crystals-Dilithium
Tabanlı		Falcon
Özet Tabanlı		Sphincs+

Dijital imzalama kategorisinde kafes tabanlı Crystals-Dilithium, Falcon algoritmaları ve özet tabanlı Sphincs+ algoritması; şifreleme ve anahtar paketleme kategorisinde ise kafes tabanlı Crystals-Kyber algoritması standart olarak seçilmiştir [11].

Bu çalışmada, dijital kripto cüzdanların ne olduğu ve blokzincir ile olan ilişkisi anlatılarak Ethereum ve Bitcoin blokzincirleri için dijital kripto cüzdan uygulamalarında cüzdan anahtarları ve adresleri oluşturma adımlarının detayı verilmiştir. Çizelge 2'de gösterilen kafes tabanlı Crystals-Dilithium ve Falcon dijital imzalama algoritmalarının farklı güvenlik seviyelerindeki anahtar ve imza boyutları ile imzalama ve doğrulama süreçlerinin performans olarak karşılaştırılmasına değinilmiştir. Ayrıca, Crystals-Dilithium algoritmasının detayına ve güvenliğinin dayandığı zor problemlerin tanımına yer verilmiştir. Kuantum bilgisayarlar varlığında kripto cüzdan uygulamalarının güvensiz hale gelecek olmasından dolayı Crystals-Dilithium algoritmasının dijital kripto cüzdan anahtar üretim sürecinde kullanılması sağlanarak Bitcoin ve Ethereum blokzincirleri için kuantum sonrası güvenli dijital kripto cüzdan uygulaması önerisinde bulunulmuştur.

Bu çalışmanın geri kalan kısmı şu şekilde organize edilmiştir: İkinci bölümde, bileşenleriyle birlikte dijital kripto cüzdan tanımı yapılmış ve blokzinciriyle olan ilişkisine değinilmiştir. Ayrıca, Ethereum ve Bitcoin kripto cüzdan uygulamalarının anahtar ve adres oluşturma süreçlerinin detayı verilmiştir. Üçüncü bölümde, kuantum sonrası kriptografi, çalışma kapsamında kullanılan algoritmalar çerçevesinde ele alınmış ve Crystals-Dilithium algoritması ve güvenliğinin dayandığı zor problemler anlatılmıştır. Dördüncü bölümde, Ethereum ve Bitcoin için kuantum sonrası dijital kripto cüzdan önerisinde bulunulmuş ve uygulama geliştirme detayları verilmiştir. Ayrıca, dijital imzalama algoritması olarak kullanılan algoritma hariç tamamen aynı özelliklere sahip klasik ve kuantum sonrası blokzincir prototipi oluşturulup klasik ve kuantum sonrası cüzdan uygulamalarının bu prototipler üzerinden transfer işlemlerini gerçekleştirmesi sağlanmış ve cüzdan bilgileri üretim sürelerinin, işlem imzalama ve doğrulama sürelerinin ortalama değerleri üzerinden performans karşılaştırılması yapılmıştır. Beşinci bölümde, sonuç ve önerilere yer verilmiştir.

2. DİJİTAL KRİPTO CÜZDANLAR

Varlıklar, dijital kripto cüzdanlarda değil blokzincirinde tutulmaktadır. Dijital kripto cüzdanlarda ise sahip olunan varlıkların ispatının yapılmasını sağlayan veriler bulunmaktadır. Bu veriler, kriptografik olarak oluşturulan açık anahtar, özel anahtar ve cüzdan adresidir. Bu veriler aynı zamanda blokzincirinde kişilerin tanımlayıcısı olarak kullanılmaktadır [12]. Blokzincir ile etkileşimde bulunabilmek için kullanıcıyı temsil eden ve tanımlayan bu bilgilere dolayısıyla dijital kripto cüzdanlara ihtiyaç vardır.

Dijital kripto cüzdanlar blokzincirinden ayrı olarak geliştirilen uygulamalardır [13]. Ancak, dijital kripto cüzdanlarda anahtar üretim sürecinde kullanılan dijital imzalama algoritması ile blokzincirinde işlem imzalama ve doğrulama sürecinde kullanılan dijital imzalama algoritması aynı olmalıdır. Aksi halde cüzdanda oluşturulan anahtarlarla işlemin imzalanması ve doğrulanması yapılamayacağı için transfer işlemleri de gerçekleştirilemeyecektir.

2.1. Dijital Cüzdan Anahtarları ve Adresi

Günümüz dijital kripto cüzdanlarda anahtar oluşturma sürecinde açık anahtarlı eliptik eğri kriptografisinden yararlanılmaktadır. Açık anahtarlı kriptografide açık ve özel olmak üzere iki anahtar bulunmaktadır ve bu anahtarlar birbiriyle matematiksel bir ilişki içerisindedir. Açık anahtarlı kriptosistemlerde mesaj açık anahtar ile şifrelenirken, şifreli mesaj ise özel anahtar ile çözümlenmektedir [5]. Bitcoin ve Ethereum blokzincirlerinde ve dijital kripto cüzdanlarda ECDSA ve secp256k1 eğrisi (y^2=x^3+7) kullanılır [3].

2.1.1. Özel (Private) Anahtar

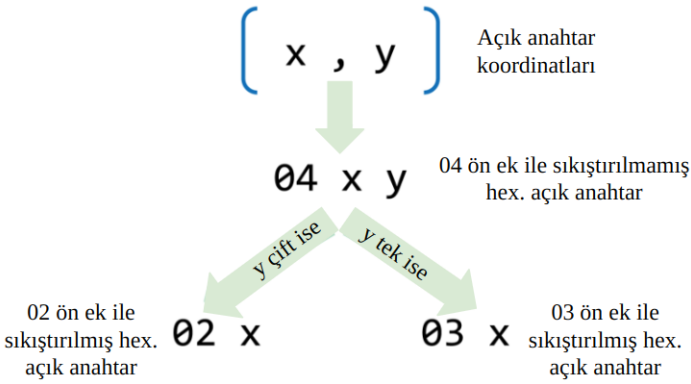
Ethereum ve Bitcoin için özel anahtar, tercihen rastgele bir tohum kullanılarak rastgele oluşturulmuş 256 bitlik bir sayıdır. Bu haliyle özel anahtar ham (raw) formatındadır ve onaltılık (hexadecimal) formata çevrilerek 64 karakterli onaltılık karakter dizisi olarak da gösterilebilir. Üretilen özel anahtarlar işlemlerin imzalanmasında ve açık anahtarların üretiminde kullanılır [2]. Özel anahtarlar kimseyle paylaşılmamalı ve saklı tutulmalıdır. Aksi halde blokzincirde geri dönüşü olmayacak şekilde varlık kayıplarının yaşanması söz konusu olabilmektedir.

2.1.2. Açık (Public) Anahtar

Açık anahtar, matematiksel olarak özel anahtarla ilişkilidir ve Bitcoin, Ethereum blokzincirlerinde eliptik eğri kriptografisi kullanılarak özel anahtardan üretilirler. Açık anahtarlar cüzdan adreslerinin üretiminde ve özel anahtarlarla oluşturulan imzaların doğrulanmasında kullanılır [12].

Rastgele üretilen 256 bitlik bir özel anahtar (k) ile secp256k1 eliptik eğrisi üzerindeki G noktasının skalar çarpımı sonucunda açık anahtar (Q) üretilmektedir. Q = k × G eliptik eğri çarpımındaki G noktasının değeri ve açık anahtarın değeri bilinse dahi özel anahtarın bulunması zor bir problemdir. Bu problem eliptik eğrilerin ayrık logaritma problemi olarak adlandırılır ve eliptik eğrilere dayalı açık anahtarlı kriptosistemlerin temeli ve güvenliği bu denklemin çözümünün zorluğuna dayanmaktadır [3].

Eliptik eğri çarpımıyla üretilen açık anahtar, yine eliptik eğri üzerinde bulunan bir noktaya denk gelmektedir. Bu nedenle açık anahtar, x ve y koordinatları şeklinde ifade edilmektedir. Sıkıştırılmış ve sıkıştırılmamış formatlarda açık anahtar türleri bulunmaktadır ve bu x ve y koordinatları açık anahtar türlerinin oluşturulmasında kullanılmaktadır. Sıkıştırılmamış formatta açık anahtar üretiminde, 04 ön eki ile açık anahtarın x ve y koordinatı birlikte kullanılırken; sıkıştırılmış formatta açık anahtar üretiminde, y değeri çift ise 02, tek ise 03 ön eki ile açık anahtarın sadece x koordinatı kullanılmaktadır [2]. Şekil 2'de açık anahtar formatlarının ön ekleriyle birlikte oluşum aşamaları gösterilmektedir.



Şekil 2. Açık anahtar türlerinin oluşturulması [2]

2.1.3. Cüzdan Adresi

Cüzdan adresleri açık anahtardan üretilmektedir ve transfer işlemlerinde cüzdan adresi baz alınarak gönderim işlemi gerçekleştirilmektedir. Cüzdan adres oluşum süreçleri blokzincir platformuna göre değişiklik gösterebilmektedir [12]. Örneğin Bitcoin ve Ethereum blokzincirlerinde cüzdan anahtarları oluşturma süreci aynı olsa da adres oluşturma süreçleri ve kullanılan kriptografik bileşenler farklılık göstermektedir.

2.2. Ethereum için Dijital Kripto Cüzdan Adres Oluşturma Aşamaları

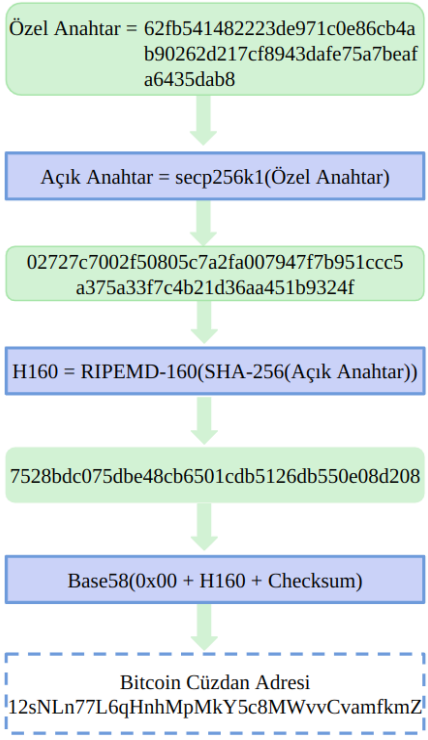
Ethereum blokzincirinde anahtar oluşturma sürecinde ECDSA kullanılmaktadır. Özel anahtar rastgele 256 bitlik bir değer olarak üretilir ardından eliptik eğri çarpımıyla açık anahtar üretilir. Elde edilen açık anahtar Keccak-256 özet fonksiyonundan geçirilir ve 256 bitlik bir özet değer elde edilir. Bu özet değerin son 20 baytlık kısmının önüne 0x ön eki getirilerek Ethereum cüzdan adresi oluşturulur [14]. Şekil 3'te Ethereum için cüzdan adres oluşum aşamaları örnek çıktılarla gösterilmektedir.

Şekil 3. Ethereum cüzdan adresi üretme aşamaları [20]

2.3. Bitcoin için Dijital Kripto Cüzdan Adres Oluşturma Aşamaları

Bitcoin blokzincirinde anahtar oluşturma sürecinde Ethereum'da olduğu gibi ECDSA kullanılmaktadır. Özel anahtar rastgele 256 bitlik bir değer olarak üretilir ardından eliptik eğri çarpımıyla açık anahtar üretilir. Elde edilen açık anahtar Hash160 olarak da bilinen çift özet alma işleminden geçirilir. Bu çift özet alma işleminde açık anahtar ilk önce SHA-256 özet fonksiyonundan ardından RIPEMD-160 özet fonksiyonundan geçirilerek 160 bitlik bir özet değer elde edilir.

Son olarak Satoshi Nakamoto tarafından önerilen Base58 kodlama türünde çıktı üreten Base58Check kodlaması aşamaları bu elde edilen özet değer üzerinde uygulanır ve bitcoin cüzdan adresi üretilir [2]. Şekil 4'te Bitcoin için cüzdan adres üretim aşamaları örnek çıktılarla gösterilmektedir.



Şekil 4. Bitcoin cüzdan adres üretim aşamaları [20]

Base58, büyük veri dizelerini daha kullanıcı dostu bir biçimde göstermek için kullanılan 58 adet karakterden oluşan ve 0 (sıfır), O (büyük o), l (küçük L), I (büyük i) gibi karıştırılabilecek harfleri içermeyen bir kodlama türüdür. Base58Check kodlaması ise yazım hatalarına karşı ekstra güvenlik eklemek için yerleşik bir hata kontrol koduna (checksum) sahip olan bir Base58 kodlama formatıdır [2].

Base58(prefix + veri + checksum) işlemi Base58Check kodlaması olarak adlandırılmaktadır. Buradaki ön ek (prefix) değeri üretilecek cüzdan adres formatına göre değişmektedir. Bitcoin mainnet ağında Açık Anahtar Özetine Ödeme (Pay to Public Key Hash - P2PKH) formatında cüzdan adresi üretilecek ise 00 ön eki, Betik Özetine Ödeme (Pay to Script Hash - P2SH) formatında cüzdan adresi üretilecek ise 05 ön eki kullanılmaktadır. Checksum ise (1) formülünde olduğu gibi ön ek ve

Hash160 ile elde edilen özet değerin iki kez SHA-256 özet fonksiyonundan geçirilmesiyle üretilen özet değerin ilk dört baytının alınmasıyla elde edilir [2].

İlk 4 bayt(SHA-256(SHA-256(prefix + H160))) (1)

Bitcoin blokzincirinde P2PKH formatında adres üretmek için açık anahtarı çift özet alma işleminden geçirdikten sonra elde edilen özet değerin Base58Check kodlamasından geçirilmesi gerekmektedir. P2PKH formatındaki cüzdan adresleri 1 ile başlamaktadır. P2SH adres formatları ise 3 ile başlar ve P2PKH'de olduğu gibi çift özet alma işleminde açık anahtar değil belirli harcama koşullarını içeren bir komut dosyasının (redeem script) özeti kullanılır. Ardından elde edilen özet değer Base58Check kodlamasından geçirilir.

Açık anahtarın ve koordinatı olmak üzere Bitcoin mainnet ağında;

Sıkıştırılmamış formatta açık anahtar kullanılarak P2PKH formatında cüzdan adresi (2) formülü ile elde edilir.

Base58(0×00 + RIPEMD-160(SHA-256(04 + +)) + checksum) (2)

Sıkıştırılmış formatta açık anahtar kullanılarak P2PKH formatında cüzdan adresi (3) formülü ile elde edilir.

Base58(0×00 + RIPEMD-160(SHA-256(02 || 03 +)) + checksum) (3)

P2SH formatta cüzdan adresi ise (4) formülü ile elde edilir.

Base58(0×05 + RIPEMD-160(SHA-256(redeem script)) + checksum) (4)

Çizelge 3'te Bitcoin mainnet ve testnet ağları için P2PKH ve P2SH formatlarında adres üretmek için kullanılan ön ek bilgileri yer almaktadır.

Çizelge 3. Bitcoin mainnet ve testnet için cüzdan adres ön ekleri [15, 16]

Mainnet			
Cüzdan Adres Formatı	**Ön Ek (Hex)**	**Ön Ek (Base58)**	**Açıklama**
P2PKH	0x00	1	25 Bayt
P2SH	0x05	3	25 Bayt
Testnet			
Cüzdan Adres Formatı	**Ön Ek (Hex)**	**Ön Ek (Base58)**	**Açıklama**
P2PKH	0x6F	m veya n	25 Bayt
P2SH	0xC4	2	25 Bayt

3. KUANTUM SONRASI KRİPTOGRAFİ

Kuantum sonrası güvenli açık anahtarlı kriptosistemlerin standartlaştırılması için NIST'in yaptığı çağrıda eleme yoluyla belirli aşamalardan geçerek üçüncü turun sonunda 2022 yılı seçilen algoritmalar listesinde yer alan algoritmalar Çizelge 2'de gösterilmektedir. Kafes tabanlı dijital imzalama algoritması olarak seçilen Crystals-Dilithium ve Falcon algoritmalarının NIST'in farklı güvenlik seviyelerinde algoritmaları vardır. NIST'in sırasıyla 2, 3 ve 5 güvenlik seviyesinde Dilithium2, Dilithium3 ve Dilithium5 algoritmaları, sırasıyla 1 ve 5 güvenlik seviyesinde Falcon-512 ve Falcon-1024 algoritmaları bulunmaktadır. Dilithium'un ayrıca performans olarak SHAKE yerine AES kullanılan Dilithium2-AES, Dilithium3-AES, Dilithium5-AES türleri de bulunmaktadır. Bu algoritmalar dijital imzalama algoritmaları olduğu için bünyelerinde anahtar oluşturma, imzalama ve doğrulama algoritmalarını barındırmaktadır [17].

NIST'in güvenlik 1 seviyesinde önerilen algoritmaların kırılmasının en az AES-128 kadar zor olduğu, güvenlik 2 seviyesinde önerilen algoritmaların en az SHA-256 kadar zor olduğu, güvenlik 3 seviyesinde önerilen algoritmaların en az AES-192 kadar zor olduğu, güvenlik 4 seviyesinde önerilen algoritmaların en az SHA-384 kadar zor olduğu ve güvenlik 5 seviyesinde önerilen algoritmaların en az AES-256 kadar zor olduğu belirtilmektedir [18].

Çizelge 4. Kuantum sonrası algoritmaların anahtar ve imza boyutları [19, 20]

Algoritma	Açık Anahtar Boyutu (Bayt)	Özel Anahtar Boyutu (Bayt)	İmza Boyutu (Bayt)
Dilithium2	1312	2528	2420
Dilithium2-AES	1312	2528	2420
Falcon-512	897	1281	690

Çizelge 4'te kuantum sonrası kafes tabanlı dijital imzalama algoritmaları olan Dilithium2, Dilithium2-AES ve Falcon-512 algoritmalarının açık anahtar, özel anahtar ve imza boyutları gösterilmektedir. Bu çizelgeye göre aynı güvenlik seviyesindeki Dilithium2 ve Dilithium2-AES algoritmalarının aynı anahtar ve imza boyutlarına sahip olduğu ancak Falcon-512 algoritmasının anahtar ve imza boyutunun diğer iki algoritmaya göre daha küçük olduğu görülmektedir.

Dilithium2-AES algoritması, Dilithium2 algoritmasına göre anahtar oluşturma sürecinde 1,7 kat, imzalama sürecinde 1,3 kat ve doğrulama sürecinde 1,6 kat daha iyi performansa sahiptir [19]. Dilithium2 algoritması ise imzalama sürecinde Falcon-512 algoritmasından 6,3 kat daha iyi performansa sahiptir [21]. Falcon-512 algoritması düşük anahtar ve imza boyutuna sahip olsa da imzalama sürecinde

Dilithium2 algoritmasına göre daha düşük performansa sahiptir ve uygulanması daha kompleks ve zordur.

Crystals-Dilithium algoritması Fiat-Shamir dönüşümü baz alınarak oluşturulan kafes tabanlı bir dijital imzalama algoritmasıdır. Güvenliği Modüller Üzerinde Tanımlı Hatalarla Öğrenme (Module Learning with Errors - MLWE) ve Modüller Üzerinde Tanımlı Kısa Tam Sayı Çözüm (Module Short Integer Solution - MSIS) problemlerinin zorluğuna dayanmaktadır. MLWE problemi Crystals-Dilithium algoritmasında anahtar oluşturma sürecinde, MSIS problemi ise imzalama ve doğrulama sürecinde kullanılmaktadır [19].

Crystals-Dilithium'un üç güvenlik seviyesi için n = 256 olacak şekilde $q = 8380417 = 2^{23} - 2^{13} + 1$ asal değeri ve $\mathfrak{R}_a = \mathbb{Z}_q[x]/(x^n + 1)$ polinom halkası kullanılır. n değeri $\mathfrak{R}_q$ polinom halkasının boyutu, q ise $\mathfrak{R}_q$ polinom halkasının modülüdür. $\mathbb{Z}_q$, mod q olarak ifade edilen $\{0,\ldots,q-1\}$ aralığında tanımlı tam sayılardır. $\mathbb{Z}_q^n$, katsayıları $\mathbb{Z}_q$'dan seçilerek oluşturulan n boyutlu vektördür. $\mathfrak{R}_q^d$ ise katsayıları $\mathbb{Z}_q^n$'den seçilerek oluşturulan d boyutlu modüldür [19, 22].

MLWE Problemi

$s \in \mathfrak{R}_q^d$ olacak şekilde bir vektör ve $\chi \in \mathfrak{R}_q$ olacak şekilde bir hata dağılımı varsayımında bulunulduğunda rastgele $a \in \mathfrak{R}_q^d$ özel ve $e \leftarrow \chi$ hata vektör değerlerinin bulunması problemidir [23].

MSIS Problemi

$a_1,\ldots,a_m \in \mathfrak{R}_q^d$ modülünde a değerleri rastgele seçilmek ve $0 < ||z|| < \beta$ olmak üzere $z_1,\ldots,z_m \in \mathfrak{R}$ değerlerini bulma problemidir [24].

4. KUANTUM SONRASI DİJİTAL KRİPTO CÜZDANLAR

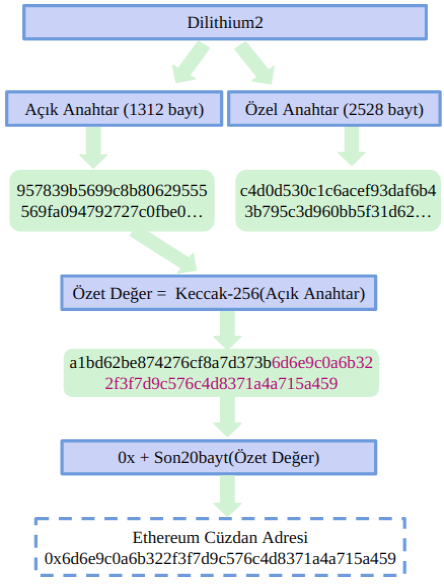
Kuantum sonrası dijital kripto cüzdanların oluşturulabilmesi, günümüzde bu uygulamalar için kullanılan ECDSA yerine kuantum sonrası için önerilen bir dijital imzalama algoritmasının kullanılmasıyla mümkün hale gelecektir. Bu uygulamalarda sadece kuantum sonrası güvenli anahtarların üretiminin sağlanması blokzincirinde transfer işleminin gerçekleştirilmesi için yeterli değildir. Transfer işleminin gerçekleştirilebilmesi için oluşturulan işlemin imzalanması da gerekmektedir. Günümüz Bitcoin ve Ethereum blokzincirlerinde imzalama ve doğrulama süreçlerinde ECDSA kullanıldığı için kuantum sonrası cüzdan uygulamasıyla üretilen kuantum güvenli cüzdan anahtarları ile bu blokzincirlerde imzalama ve doğrulama süreçleri gerçekleştirilemeyecektir. Dolayısıyla kripto cüzdan uygulamalarında kullanılan dijital imzalama algoritması ile blokzincir uygulamasında kullanılan dijital imzalama algoritması aynı olmalıdır. Diğer bir ifadeyle kuantum

sonrası dijital kripto cüzdanlar ile transfer işlemlerinin gerçekleştirilebilmesi kuantum sonrası bir blokzincir platformu varlığında mümkün hale gelecektir.

Çalışma kapsamında önerilen cüzdan uygulamalarının anahtar üretim aşamasında NIST güvenlik seviyesi 2 olan Dilithium2 algoritması kullanılmıştır. Üretilen açık anahtar boyutu 1312 bayt ve özel anahtar boyutu 2528 bayt olduğu için kullanıcıya üretilen anahtarların son 32 baytlık kısmı (256 bit) gösterilmektedir. Ancak, cüzdan adresi oluşturma, işlem imzalama ve doğrulama süreçlerinde Dilithium2 algoritması ile üretilen anahtar çiftleri kullanılmaktadır.

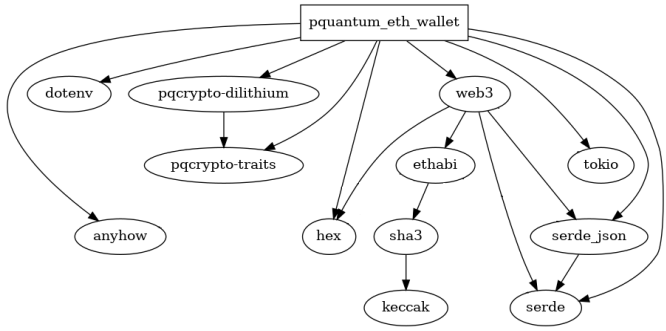
4.1. Ethereum için Kuantum Sonrası Dijital Kripto Cüzdan Önerisi

Ethereum için üretilen kuantum sonrası dijital kripto cüzdan uygulamasında ilk önce Dilithium2 algoritması kullanılarak açık ve özel anahtarların üretilmesi sağlanmıştır. Ardından üretilen açık anahtar Keccak-256 özet fonksiyonundan geçirilerek 256 bitlik özet değer elde edilmiştir. Bu özet değerin son yirmi baytı alınarak ve önüne 0x ön eki getirilerek Ethereum için kuantum sonrası dijital kripto cüzdan adresi üretilmiştir. Şekil 5'te Ethereum için önerilen kuantum sonrası dijital kripto cüzdan uygulamasının adres oluşturma adımları örnek çıktılarla gösterilmektedir. Şekil 3'te gösterilen Ethereum cüzdan adres oluşturma sürecinden farklı olarak anahtar üretim sürecinde ECDSA yerine kuantum sonrası Dilithium2 algoritması kullanılmıştır.



Şekil 5. Ethereum için kuantum sonrası anahtar ve adres üretme aşamaları [25]

Ethereum için kuantum sonrası cüzdan uygulama önerisi Rust programlama dili ile gerçeklenmiştir ve uygulamada kullanılan kütüphaneler Şekil 6'da gösterilmektedir. Cüzdan anahtarlarını üretmekte kullanılan Dilithium2 algoritması için *pquantum-dilithium* ve *pquantum-dilithium-traits* kütüphaneleri [26, 27], hata yönetimi için *anyhow* kütüphanesi [28], çevre değişkenlerinin tutulmasında *dotenv* kütüphanesi [29], hexadecimal formata dönüşüm için *hex* kütüphanesi [30], Keccak-256, adres formatı ve Ethereum düğümüyle etkileşim kurulabilmesi için *web3* kütüphanesi [31], verileri dosyaya kaydederken ve dosyadan okuma yaparken veri serileştirme ve ters serileştirme işlemleri için *serde* ve *serde_json* kütüphaneleri [32, 33], Ethereum ağına bağlanıldığında bakiye sorgulama, blok numarası öğrenme gibi işlemlerin eş-zamansız bir şekilde yapılabilmesi için *tokio* kütüphanesi [34] kullanılmıştır. Uygulama aynı zamanda Github'da https://github.com/ecmelkytz/post-quantum-crypto-wallets/tree/main/pq_eth_wallet adresinde yayınlanmaktadır.



Şekil 6. Ethereum kuantum sonrası dijital cüzdan uygulamasında kullanılan kütüphaneler [25]

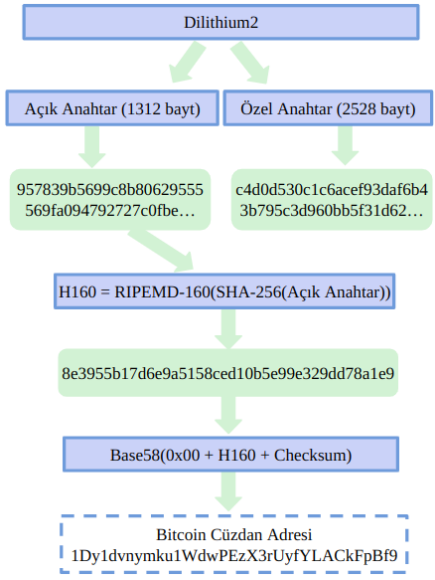
Uygulama çalıştırıldığında üretilen olası açık anahtar, özel anahtar ve cüzdan adresi bilgileri aşağıdaki gibi olacaktır.

```
"secret_key": "6d13***", "public_key":
"572fe19d0575d4f817ac79e335ff7af858647cfe1f26fd66c97
9630fbce10783", "address":
"0x6d6e9c0a6b322f3f7d9c576c4d8371a4a715a459"
```

4.2. Bitcoin için Kuantum Sonrası Dijital Kripto Cüzdan Önerisi

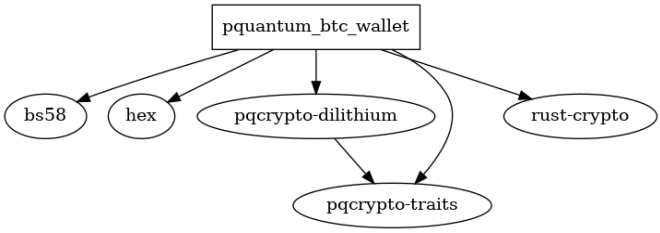
Bitcoin için üretilen kuantum sonrası dijital kripto cüzdan uygulamasında ilk önce Dilithium2 algoritması kullanılarak açık ve özel anahtarların üretilmesi sağlanmıştır. Ardından açık anahtar sırasıyla SHA-256 ve RIPEMD-160 özet fonksiyonlarından geçirilerek 160 bitlik özet değer elde edilmiştir. Bu özet değer Base58Check kodlamasından geçirilerek P2PKH formatında cüzdan adresi üre-

tilmiştir. Şekil 7'de Bitcoin için önerilen kuantum sonrası dijital kripto cüzdan uygulamasının anahtar ve adres oluşturma adımları örnek çıktılarla gösterilmektedir. Şekil 4'te gösterilen Bitcoin cüzdan adres oluşturma sürecinden farklı olarak anahtar üretim sürecinde ECDSA yerine kuantum sonrası Dilithium2 algoritması kullanılmıştır.



Şekil 7. Bitcoin için kuantum sonrası anahtar ve adres üretme aşamaları [25]

Bitcoin için kuantum sonrası cüzdan uygulama önerisi Rust programlama dili ile gerçeklenmiştir ve uygulamada kullanılan kütüphaneler Şekil 8'de gösterilmektedir. Cüzdan anahtarlarını üretmekte kullanılan Dilithium2 algoritması için *pquantum-dilithium* ve *pquantum-dilithium-traits* kütüphaneleri [26, 27], Base58 formatında çıktılar üretebilmek için *bs58* kütüphanesi [35], hexadecimal formata dönüşüm için *hex* kütüphanesi [30], SHA-256 ve RIPEMD-160 özet değerlerini üretebilmek için *rust-crypto* kütüphanesi [36] kullanılmıştır. Uygulama aynı zamanda Github'da https://github.com/ecmelkytz/post-quantum-crypto-wallets/tree/main/pq_btc_wallet adresinde yayınlanmaktadır.



Şekil 8. Bitcoin kuantum sonrası dijital cüzdan uygulamasında kullanılan kütüphaneler [25]

Uygulama çalıştırıldığında üretilen olası açık anahtar, özel anahtar ve cüzdan adresi bilgileri aşağıdaki gibi olacaktır.

```
1    “secret_key”: “da00e***”
2    “public_key”:
3    “1a4c0b7cc5f01f90eacc25438904e511a3a5c068b79f29347a
4    2353cbb44412e0”,
5    “address”: “1Nhr9LK4yoSZtPtriubXxAbmXHYTojPxpu”
```

Bitcoin ve Ethereum için önerilen kuantum sonrası cüzdan uygulamalarında üretilen cüzdan adresleri, Bitcoin ve Ethereum blokzincir adres formatlarına uygun olduğunda herhangi bir Ethereum veya Bitcoin cüzdan adresinden bu cüzdan adreslerine para transferi gerçekleştirilebilir. Ancak, bu cüzdan adreslerinden herhangi bir Bitcoin veya Ethereum cüzdan adresine para transferinin gerçekleşebilmesi için Bitcoin ve Ethereum blokzincir ağlarında imzalama ve doğrulama süreçlerinde Dilithium2 algoritmasının kullanılması gerekmektedir. Bu sebeple önerilen kuantum sonrası kripto cüzdanları kullanarak transfer işlemlerinin kuantum sonrası bir blokzincir uygulamasında gerçekleştirildiğinin gösterilebilmesi adına kuantum sonrası bir blokzincir prototipi oluşturulmuştur.

4.3. Bitcoin ve Ethereum için Klasik ve Kuantum Sonrası Blokzincir Prototipinin Oluşturulması

Çalışma kapsamında, Rust programlama dili ile geliştirilen, imzalama ve doğrulama süreçlerinde Dilithium2 algoritmasının kullanıldığı kuantum sonrası bir blokzincir prototipi oluşturulmuştur. Bitcoin ve Ethereum için oluşturulan kuantum sonrası güvenli kripto cüzdan uygulamalarının bu prototip üzerinde çalıştırılıp transfer işleminin gerçekleştirilmesi https://github.com/ecmelkytz/post-quantum-blockchain adresinde yayınlanmaktadır. Ayrıca, kuantum sonrası için oluşturulan blokzincir prototipi ile aynı özelliklere sahip ancak sadece Dilithium2 algoritması yerine ECDSA kullanıldığı klasik blokzincir prototipi oluşturularak klasik ve kuantum sonrası dijital kripto cüzdanların kullanımı

ile işlemlerin imzalanması ve doğrulanması süreçlerinin karşılaştırılmasının yapılabilmesi sağlanmıştır. Rust programlama dili ile oluşturulan klasik Bitcoin ve Ethereum kripto cüzdan uygulamaları https://github.com/ecmelkytz/crypto-wallets adresinde ve bu cüzdan uygulamalarının oluşturulan klasik blokzincir prototipi üzerinde çalıştırılıp transfer işleminin gerçekleştirilmesi https://github.com/ecmelkytz/blockchain adresinde yayınlanmaktadır.

Klasik ve kuantum sonrası Bitcoin ve Ethereum için oluşturulan blokzincir prototipinde zorluk derecesi iki olarak belirlenmiştir. Zorluk derecesinin iki olması madenci hesap tarafından bulunması gereken özet değerin ilk iki hanesinin sıfır ile başlayan bir değer olması gerektiğini belirtmektedir. Bu değer artırılarak özet değerin bulunması zorlaştırılabilir.

4.4. Klasik ve Kuantum Sonrası Dijital Kripto Cüzdan Uygulamalarının Performans Olarak Karşılaştırılması

Klasik ve kuantum sonrası cüzdan uygulamaları arasında yapılan performans karşılaştırmaları Intel® Core™ i7-7500U CPU @ 2.70 GHz işlemci kullanılarak yapılmıştır. Cüzdan bilgilerinin oluşturulma süresinin belirlenmesi, işlem imzalama ve doğrulama sürelerinin karşılaştırılması, bu işlemlerin 1000 kez çalıştırılması sonucu elde edilen sürelerin ortalamasının alınmasıyla hesaplanmıştır.

Çizelge 5'te Bitcoin ve Ethereum için oluşturulan klasik ve kuantum sonrası dijital kripto cüzdan uygulamalarının, cüzdan bilgilerini (anahtar ve adres) oluşturma süreleri gösterilmektedir. Bu çizelgeye göre kuantum sonrası dijital kripto cüzdan bilgilerinin klasik dijital kripto cüzdan bilgilerine göre daha kısa sürede oluşturulduğu görülmektedir.

Çizelge 5. Bitcoin ve Ethereum için klasik ve kuantum sonrası kripto cüzdan bilgilerinin ortalama oluşturulma süreleri

Cüzdan	Klasik (ms)	Kuantum Sonrası (ms)
Bitcoin	28.175	0.195
Ethereum	29.156	0.159

Klasik ve kuantum sonrası dijital kripto cüzdan uygulamalarının imzalama ve doğrulama süreçlerinin karşılaştırılması Rust programlama dili ile klasik ve kuantum sonrası için geliştirilen blokzincir prototipleri üzerinden yapılmıştır. Çizelge 6'da cüzdan uygulamalarının imzalama ve doğrulama süreleri gösterilmektedir.

Çizelge 6. Bitcoin ve Ethereum için klasik ve kuantum sonrası blokzincir prototipinde imzalama ve doğrulama işlemlerinin ortalama gerçekleşme süreleri

	Klasik Blokzincir Prototipi		Kuantum Sonrası Blokzincir Prototipi	
Cüzdan	**İmzalama (ms)**	**Doğrulama (ms)**	**İmzalama (ms)**	**Doğrulama (ms)**
Bitcoin	28.088	28.107	0.155	0.112
Ethereum	28.425	27.955	0.136	0.114

Kuantum sonrası için geliştirilen blokzincir prototipinde işlem imzalama ve doğrulama süreçlerinin klasik blokzincir prototipine göre daha kısa sürede gerçekleştirildiği görülmektedir. Rust programlama dili ile ECDSA için yazılan ve performans olarak anahtar oluşturma, imzalama ve doğrulama sürelerinin farklı olduğu kütüphaneler bulunmaktadır [37]. Aynı şekilde, Crystals-Dilithium algoritması için Rust programlama dili ile yazılan farklı kütüphanelere https://crates.io adresinden ulaşmak mümkündür. Bu çalışmada, klasik dijital kripto cüzdan ve blokzincir prototipinde ECDSA için *secp256k1* kütüphanesi [38] kullanılmıştır. Kuantum sonrası dijital kripto cüzdan ve blokzincir prototipi için *pquantum-dilithium* ve *pquantum-dilithium-traits* kütüphaneleri [26, 27] kullanılmıştır. Uygulamada dijital imzalama algoritması için kullanılan kütüphaneye göre anahtar oluşturma, imzalama ve doğrulama işlemlerinin gerçekleşme süreleri farklılık gösterecektir.

5. SONUÇ VE ÖNERİLER

Bu çalışmada, blokzincir ile dijital kripto cüzdan uygulamaları arasındaki ilişki belirtilerek Bitcoin ve Ethereum blokzincirleri için dijital kripto cüzdan üretim aşamaları detayı verilmiştir. Günümüz kripto cüzdan uygulamalarında kullanılan açık anahtarlı eliptik eğri kriptografisinin kuantum bilgisayarlar varlığında güvenliğinin sağlanamayacak olmasından dolayı NIST'in kuantum sonrası kriptosistemleri standartlaştırma sürecinde seçilen dijital imzalama kategorisinde kafes tabanlı Crystals-Dilithium algoritmasının cüzdan anahtar üretiminde kullanımı sağlanarak kuantum sonrası Bitcoin ve Ethereum blokzincirleri için güvenli dijital kripto cüzdan uygulaması önerisinde bulunulmuştur. Bitcoin ve Ethereum için klasik ve kuantum sonrası cüzdan bilgilerinin ortalama oluşturulma süresi belirtilmiştir. Ayrıca, bu uygulamaların imzalama ve doğrulama süreçlerinin kıyaslanabilmesi için klasik ve kuantum sonrası blokzincir prototipi oluşturulmuş ve bu prototipler üzerinden cüzdan uygulamalarının çalıştırılıp transfer işleminin gerçekleştirilmesi sağlanarak imzalama ve doğrulama süreçlerinin ortalama süreleri belirtilmiştir.

Bu çalışma ayrıca kuantum sonrası için önerilen farklı algoritmaların dijital kripto cüzdan uygulamalarında kullanılması adına referans kaynak olarak kullanılabilir ve üretilen kuantum sonrası dijital kripto cüzdan uygulamalarının performans olarak karşılaştırılması üzerinde çalışmalar yapılabilir.

Teşekkür

2. yazar TÜBİTAK tarafından 121R006 numaralı proje kapsamında kısmi olarak desteklenmiştir.

Yazar Katkı Oranları

Çalışmanın Tasarlanması (Design of Study): EA(% 50), SA(% 50)

Veri Toplanması (Data Acquisition): EA(% 50), SA(% 50)

Veri Analizi (Data Analysis): EA(% 50), SA(% 50)

Makalenin Yazımı (Writing Up): EA(% 50), SA(% 50)

Makale Gönderimi ve Revizyonu (Submission and Revision): EA(% 50), SA(% 50)

KAYNAKLAR

[1] Ayvaz, S., Çetin, S. C. ve Aydar, M. Kimlik Sistemlerinde Blokzincir Kullanımı, Sağıroğlu, Ş. ve Akleylek, S. (ed.). Siber Güvenlik ve Savunma: Blokzincir ve Kriptoloji, Ankara: Nobel, 5, ss. 117-164, 2021

[2] Antonopoulos, A. M. Mastering Bitcoin Programming the Open Blockchain, Second Edition, Sebastopol, O'Reilly, 2017

[3] Nakov, S. Practical Cryptography for Developers, Software University, 2018.

[4] Akleylek, S., Yıldırım, H. M. ve Tok, Z. Y. Kriptoloji ve Uygulama Alanları: Açık Anahtar Altyapısı ve Kayıtlı Elektronik Posta, Akademik Bilişim 11, Bildiriler Kitabı, ss. 723-728, 2013

[5] Paar, C. ve Pelzl, J. Understanding Cryptography: A Textbook for Students and Practitioners, Springer, 2010. https://doi.org/10.1007/978-3-642-04101-3

[6] Shor, P. W. Algorithms for quantum computation: discrete logarithms and factoring, In Proceedings 35th Annual Symposium on Foundations of Computer Science (SFCS '94), 20-22 November, Santa Fe, NM, USA, 1994, ss. 124-134

[7] Bernstein, D. J., Buchmann, J. ve Dahmen, E. Post-Quantum Cryptography, Berlin: Springer, 2009

[8] Bernstein, D. J. ve Lange, T. Post-Quantum Cryptography. Nature. 549, ss. 188-194, 2017

[9] NIST, Post Quantum Cryptography, https://csrc.nist.gov/Projects/Post-Quantum-Cryptography

[10] Alagic, G., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y., Miller, C. , Moody, D., Peralta, R., Perlner, R., Robinson, A., Smith-Tone, D. and Apon, D. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process, NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, 2022. https://doi.org/10.6028/NIST.IR.8413

[11] NIST, Post Quantum Cryptography Selected Algorithms https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022

[12] Yaga, D., Mell, P., Roby, N., ve Scarfone, K. Blockchain Technology Overview—National Institute of Standards and Technology Internal Report 8202. Gaithersburg, MD: National Institute of Standards and Technology, 2018

[13] Mirza, M. M., Ozer, A. ve Karabiyik, U. (2022). Mobile Cyber Forensic Investigations of Web3 Wallets on Android and iOS. Applied Sciences. 12(21), 11180. https://doi.org/10.3390/app122111180

[14] Antonopoulos, A. M., Wood, G. Mastering Ethereum: Building Smart Contracts and Dapps. Sebastopol, CA, USA: O'Reilly Media, Inc., 2018

[15] Walker, G. Address. Learn me a Bitcoin, 2021

[16] Seguias, B. E. K. Bitcoin – Private key, Public key, and Addresses, 2018

[17] NIST. PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates

[18] NIST, Post Quantum Cryptography Security (Evaluation Criteria), https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography standardization/Evaluation-Criteria/Security-(Evaluation-Criteria)

[19] Bai, S., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G. ve Stehl´e, D. CRYSTALS-Dilithium algorithm specifications and supporting documentation, 2021
[20] Fouque, P. A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Ricosset, T., Seiler, G., Whyte, W. ve Zhang, Z. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU, 2017.
[21] Sikeridis, D., Kampanakis, P. & Devetsikiotis, M. Post-Quantum Authentication in TLS 1.3: A Performance Study, 27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, 2020
[22] ETSI (2021). CYBER; Quantum-Safe Signatures. Technical Report, TR 103 616 V1.1.1. https://www.etsi.org/deliver/etsi_tr/103600_103699/103616/01.01.01_60/tr_103616v010101p.pdf
[23] Akleylek, S. ve Seyhan, K. Kuantum Bilgisayarlar Sonrası Güvenilir Kafes Tabanlı Kriptosistem Temelleri, Sağıroğlu, Ş. ve Akleylek, S. (ed.). Siber Güvenlik ve Savunma Problem ve Çözümler, Ankara: Grafiker Yayınları, 2, 169-209, 2019
[24] Langlois, A. ve Stehlé, D. Worst-case to average-case reductions for module lattices, Des. Codes Cryptogr, 2015, 75, 565–599. https://doi.org/10.1007/s10623-014-9938-4
[25] Albayrak, E. Blokzincirde Akıllı Sözleşmeler ve Güvenli E-Cüzdan Uygulaması, Yüksek Lisans Tezi, 2023.
[26] Pqcrypto-dilithium. Erişim tarihi: 29 Ağustos, 2023, Erişim adresi: https://docs.rs/pqcrypto-dilithium/0.4.6/pqcrypto_dilithium
[27] Pqcrypto-traits. Erişim tarihi: 29 Ağustos, 2023, Erişim adresi: https://docs.rs/pqcrypto-traits/0.3.4/pqcrypto_trait
[28] Anyhow. Erişim tarihi: 29 Ağustos, 2023, Erişim adresi: https://docs.rs/anyhow/latest/anyhow
[29] Dotenv. Erişim tarihi: 29 Ağustos, 2023, Erişim adresi: https://docs.rs/dotenv/0.15.0/dotenv
[30] Hex. Erişim tarihi: 29 Ağustos, 2023, Erişim adresi: https://docs.rs/hex/0.4.3/hex
[31] Web3. Erişim tarihi: 29 Ağustos, 2023, Erişim adresi: https://docs.rs/web3/0.17.0/web3/
[32] Serde. Erişim tarihi: 29 Ağustos, 2023, Erişim adresi: https://docs.rs/serde/1.0.136/serde
[33] Serde_json. Erişim tarihi: 29 Ağustos, 2023, Erişim adresi: https://docs.rs/serde_json/1.0.40/serde_json
[34] Tokio. Erişim tarihi: 29 Ağustos, 2023, Erişim adresi: https://docs.rs/tokio/1.29.1/tokio
[35] Bs58. Erişim tarihi: 29 Ağustos, 2023, Erişim adresi: https://docs.rs/bs58/latest/bs58
[36] Rust-crypto. Erişim tarihi: 29 Ağustos, 2023, Erişim adresi: https://docs.rs/rust-crypto/0.2.36/crypto
[37] Arcieri, T. *Rust secp256k1 ECDSA benchmarks*. Erişim tarihi: 5 Eylül, 2023, Erişim adresi: https://github.com/tarcieri/rust-secp256k1-ecdsa-bench
[38] Secp256k1. Erişim tarihi:5Eylül,2023, Erişim adresi: https://docs.rs/secp256k1/0.20.3/secp256k1/index.html