ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

TLS PROTOKOLÜNDE POST-KUANTUM KRİPTOGRAFİ-PROTOKOL GÜVENLİĞİNİ ARTIRMA STRATEJİLERİ

Tuğba ÖZCAN

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2025

ÖZET

Yüksek Lisans Tezi

TLS PROTOKOLÜNDE POST-KUANTUM KRİPTOGRAFİ-
PROTOKOL GÜVENLİĞİNİ ARTIRMA STRATEJİLERİ

Tuğba ÖZCAN

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Mehmet Serdar GÜZEL

Kuantum bilgisayarların gelişmesi, günümüzde kullanılan açık anahtarlı kriptografik sistemlerin güvenliğini ciddi biçimde tehdit etmektedir. Bu nedenle, güvenli iletişim protokollerinin kuantum sonrası algoritmalarla güncellenmesi gerekmektedir. Bu tez, post-kuantum kriptografik (PQC) algoritmaların Transport Layer Security (TLS) 1.3 protokolüne entegrasyonunu ve bu entegrasyonun performans üzerindeki etkilerini incelemektedir.

Çalışma kapsamında, TLS 1.3 protokolünde hem anahtar değişimi hem de sertifika tabanlı kimlik doğrulama senaryoları test edilmiştir. Klasik, hibrit ve saf PQC algoritmalarıyla yürütülen deneylerde farklı TLS 1.3 yapılandırmaları değerlendirilmiş ve her senaryo 1000 tekrar üzerinden ölçülmüştür. Ölçümler, el sıkışma süreleri, sertifika boyutları ve standart sapmalar dikkate alınarak karşılaştırmalı bir şekilde analiz edilmiştir.

Elde edilen bulgular, PQC algoritmalarının TLS 1.3'e entegrasyonunun teknik açıdan mümkün ve uygulanabilir olduğunu göstermektedir. Ayrıca performans kayıplarının büyük ölçüde yönetilebilir seviyede kaldığı görülmüştür. Özellikle anahtar değişiminde ML-KEM ailesi, kimlik doğrulamada ise ML-DSA ve Falcon düşük gecikme süreleri ve istikrarlı performanslarıyla öne çıkmış; SPHINCS+ ise görece büyük imza boyutlarıyla ek gecikme süresine sahiptir. Hibrit yaklaşımlar ise klasik sistemlerle uyumluluğu koruyarak güvenli geçiş için alternatif sunmaktadır.

Eylül 2025, 70 sayfa

Anahtar Kelimeler: Post-Kuantum Kriptografi (PQC), Transport Layer Security (TLS) 1.3, Hibrit Kriptografi, Açık Anahtar Altyapısı (PKI), Elektronik İmza, Anahtar Değişim Mekanizmaları, Kimlik Doğrulama, Kriptografik Çeviklik, Performans Değerlendirmesi, Kuantum-Dirençli Güvenlik, ML-KEM, ML-DSA, Falcon, SPHINCS+

ABSTRACT

Master's Thesis

POST-QUANTUM CRYPTOGRAPHY IN TLS PROTOCOL:
STRATEGIES FOR ENHANCED PROTOCOL SECURITY

Tuğba ÖZCAN

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Prof. Dr. Mehmet Serdar GÜZEL

The advent of quantum computers has the potential to compromise the security of contemporary public-key cryptographic systems. Consequently, the necessity to update secure communication protocols with post-quantum algorithms has become inevitable. The present thesis investigates the integration of post-quantum cryptographic (PQC) algorithms into the Transport Layer Security (TLS) 1.3 protocol, and examines the performance implications of this integration.

In the context of this study, both key exchange and certificate-based authentication scenarios in TLS 1.3 were examined. Experiments were conducted utilising classical, hybrid, and pure PQC algorithms under various TLS 1.3 configurations, with each scenario evaluated across 1000 iterations. The analysis considered handshake latency, certificate sizes, and standard deviations in a comparative manner.

The findings demonstrate that the integration of PQC algorithms into TLS 1.3 is both technically feasible and practically applicable. Furthermore, the performance overhead remains largely manageable. The ML-KEM family is particularly noteworthy in terms of key exchange, while ML-DSA and Falcon demonstrate low latency and stable performance in authentication. Conversely, SPHINCS+ introduces additional delay due to its relatively large signature sizes. Hybrid approaches offer an alternative for a secure transition by maintaining compatibility with classical systems.

September 2025, 70 page

Key Words: Post-Quantum Cryptography (PQC), Transport Layer Security (TLS) 1.3, Hybrid Cryptography, Public Key Infrastructure (PKI), Digital Signature, Key Exchange Mechanisms, Authentication, Cryptographic Agility, Performance Evaluation, Quantum-Resistant Security, ML-KEM, ML-DSA, Falcon, SPHINCS+

TEŞEKKÜR

Tez çalışmam hem akademik ilgim hem de mesleki deneyimim doğrultusunda şekillenmiştir. Amacım post-kuantum kriptografi konusunda derinleşmektir. Uzun yıllar boyunca açık anahtar altyapısı, elektronik imza teknolojileri ve SSL sertifikaları ekosistemi üzerine çalıştım ve bilgi birikimi edindim. Tez çalışma konumun belirlenmesinde bu birikim belirleyici olmuştur. Tez çalışmam ile Post-kuantum algoritmaların değerlendirilmesi açısından literatüre özgün bir katkı sunmak da öncelikli hedefimdir.

Bu süreçte bilgi ve tecrübeleriyle yol gösteren, akademik rehberliği ile çalışmanın yönünü belirlemeye yardımcı olan danışmanım Prof. Dr. Mehmet Serdar GÜZEL'e en içten teşekkürlerimi sunuyorum. Ayrıca meslek hayatımın tüm zorlu süreçlerinde olduğu gibi bu süreçte de varlıklarıyla bana güç veren oğullarım Eymen ve Ahmet Aras' a, her zaman desteğini hissettiren eşim Eray ÖZCAN'a ve manevi desteği ile daima yanımda olan aileme yürekten teşekkür ederim.

Tuğba ÖZCAN
Ankara, Eylül 2025

İÇİNDEKİLER

KISALTMALAR DİZİNİ

AES	:	Advanced Encryption Standard
AEAD	:	Authenticated Encryption with Associated Data
GCM	:	Galois/Counter Mode (AEAD kiplerinden)
BIKE	:	Bit Flipping Key Encapsulation
CA	:	Certificate Authority
CRYSTALS	:	Cryptographic Suite for Algebraic Lattices
CSR	:	Certificate Signing Request
CVP	:	Closest Vector Problem
DER	:	Distinguished Encoding Rules
DHE	:	Diffie–Hellman Ephemeral
ECDH	:	Elliptic Curve Diffie–Hellman
ECDHE	:	Elliptic Curve Diffie–Hellman Ephemeral
ECDSA	:	Elliptic Curve Digital Signature Algorithm
FALCON	:	Fast-Fourier Lattice-based Compact Signatures over NTRU
FIPS	:	Federal Information Processing Standard
HKDF	:	HMAC-based Extract-and-Expand Key Derivation Function
HQC	:	Hamming Quasi-Cyclic
IETF	:	Internet Engineering Task Force
KEM	:	Key Encapsulation Mechanism
LWE	:	Learning With Errors
ML-DSA	:	Module-Lattice–based Digital Signature Algorithm
ML-KEM	:	Module-Lattice–based Key Encapsulation Mechanism
NIST	:	National Institute of Standards and Technology
OCSP	:	Online Certificate Status Protocol
OQS	:	Open Quantum Safe
PEM	:	Privacy-Enhanced Mail
PKI	:	Public Key Infrastructure
PQC	:	Post-Quantum Cryptography
RA	:	Registration Authority
RFC	:	Request for Comments

RTT	:	Round-Trip Time
SIS	:	Short Integer Solution
SNI	:	Server Name Indication
SPHINCS+	:	Stateless Hash-Based Signature Scheme
SVP	:	Shortest Vector Problem
TCP	:	Transmission Control Protocol
TLS	:	Transport Layer Security
WOTS+	:	Winternitz One-Time Signature Plus
X.509	:	Açık Anahtar Altyapısı (PKI) dijital sertifika standardı
X25519	:	Curve25519 üzerinde ECDH (TLS'te yaygın KEX)
P-256 / P-384 / P-521	:	NIST Eğrileri

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

1.1 Genel Arka Plan

Dijital iletişimde verilerimizin güvenli bir şekilde iletilmesi oldukça önemlidir. Bankacılıktan, e-devlet hizmetlerine kadar birçok alanda dijital iletişimin olduğu düşünüldüğünde veri güvenliği de üzerinde düşünülmesi ve teknik açıdan da irdelenmesi gereken bir husus haline gelmektedir. Dijital iletişimde verilerin güvenliği kullanılan kriptografik protokoller ile sağlanır. Bu protokoller, bilgiye yalnızca yetkili kişilerin erişmesini garanti ederken, veri bütünlüğününün korunmasını ve verinin gizli kalmasını da sağlar. Kriptografik protokollerin güvenliği ise protokolü oluşturan kriptografik ilkellere (dijital imza, şifreleme, anahtar değişimi) ve bu ilkellerde kullanılan algoritmaların dayanıklı ve güvenilir olması ile doğrudan ilişkilidir. Taşıma Katmanı Güvenliği (Transport Layer Security, TLS), güvenli iletişim için en sık kullanılan protokollerden biridir.

TLS protokolü şifreleme yöntemleri ve kimlik doğrulamayı kullanır. İletilen verinin doğru yere gizli şekilde gönderilmesi sağlanır. TLS protokolünde RSA (Rivest-Shamir-Adleman) ve Eliptik Eğri Kriptografisi (Elliptic Curve Cryptography, ECC) gibi asimetrik algoritmalar kullanılır. Bu algoritmaların güvenliği matematiksel olarak büyük sayıların asal çarpanlarına ayrılması ve eliptik eğriler üzerinde ayrık logaritma problemi gibi bilinen ve çözülmesi zor problemlere dayanır. Bu problemleri çözmek klasik bilgisayarlar için oldukça zordur. Bu yüzden algoritmalar günümüzde güvenli kabul edilirler. Ancak kuantum bilgisayarların gelişimi bu güvenilirliği tehdit etmektedir. Peter Shor tarafından 1994 yılında geliştirilen kuantum algoritması, hem RSA gibi çarpanlara ayırmaya dayalı sistemleri hem de ECC gibi eliptik eğri kriptografileri güvensiz hale getirebilecektir. Shor algoritması, klasik bilgisayarların üstel zamanda çözebildiği tam sayı çarpanlarına ayırma ve eliptik eğri ayrık logaritma problemlerini kuantum bilgisayarlar üzerinde polinomsal zamanda çözebilme yeteneğine sahiptir (Shor, 1994). Böylece günümüzde yaygın olarak kullanılan asimetrik kriptografi yöntemleri, yeterli sayıda kuantum biti (qubit) içeren bir kuantum bilgisayar üzerinde teorik olarak kırılabilir durumdadır. Shor algoritması, bu hususta geliştirilen tek yöntem değildir. Grover

algoritması, simetrik şifreleme algoritmalarına karşı kuantum hızlandırmasına sahiptir. Bu hızlanma AES gibi blok şifreleme sistemlerinin çözülmesinde anahtar uzunluğu üzerinde karekök düzeyinde bir azaltma sağlar. Bunlar düşünüldüğünde hem asimetrik hem de simetrik kriptografi yöntemleri kuantum çağında yeniden değerlendirilmek zorundadır.

Yukarıda bahsettiğim riskler sadece gelecekte kuantum bilgisayarlarının yaygınlaştığı bir dönemde ortaya çıkacak bir tehdit olarak görülmemelidir. Halihazırda güvenli olduğunu düşündüğümüz kanallar aracılığıyla iletilip şifrelendiği düşünülen verilerin saldırganlar tarafından şifreli şekilde keydedilmesi muhtemeldir. İleride ise bir kuantum bilgisayarı kullanılarak çözülebilecek bir duruma gelecektir. Özellikle savunma sanayi düşünüldüğünde bu ciddi bir güvenlik problemidir. Bu yöntem genellikle "şimdi sakla sonra aç" (Store now, decrypt later) saldırı modeli olarak adlandırılmaktadır. Bu durumda özellikle gizli verilerin uzun süre korunması gereken durumlarda önlemlerin şimdiden alınması önemlidir.

Kuantum bilgisayarların mevcut açık anahtarlı kriptografik sistemleri kırabilme potansiyeline karşı Post-Kuantum Kriptografi (Post Quantum Cryptography, PQC) olarak bilinen yeni algoritmalar geliştirilmektedir. Bu algoritmalar, klasik bilgisayarlarda verimli çalışmak üzere tasarlanırken, Shor ve Grover gibi kuantum algoritmalara karşı da direnç göstermesi beklenmektedir.

2016 yılında Amerika Birleşik Devletleri Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standards and Technology, NIST), PQC algoritmaların belirlendiği bir standartlaştırma sürecini başlatmıştır. (NIST 2024). Bu sürecin yürütülmesinde matematikten, kriptolojiye ve bilgisayar bilimlerine kadar çeşitli disiplinlerden araştırmacılar katılım sağlamıştır. Bu alanda çalışan ve çalışma sonuçlarını ispatlamak isteyen herkesin katılıma açık olan süreç başlangıçta onlarca algoritma önerisiyle başlamıştır. Bu süreçte önerilen algoritmalar, çoğunlukla kafes (lattice) tabanlı, kod tabanlı (code-based), çok değişkenli (multivariate) ve özet tabanlı (hash-based) kriptografik yapılar üzerine kurulmuştur. Bu yapılar, çözülmesi zor matematiksel problemler kullanarak hem kuantum hem de klasik saldırılara karşı direnç sağlayacak

biçimde tasarlanmıştır. NIST'in PQC standardizasyon süreci halen devam etmekte olup, bazı algoritmalar için standartlaşma süreçleri başlatılmıştır.

NIST'in bu şeffaf değerlendirme süreci yalnızca algoritma seçimi değil, dünya genelinde bilgi teknolojilerinde dönüşüm stratejisini belirlemek açısından da oldukça önemlidir. NIST ve benzeri çalışmalara gerçekte uyumluluk kazandırmayı amaçlayan girişimler de bulunmaktadır. PQC algoritmaların gerçek dünyada test edilerek mevcut protokollere entegre edilmesini sağlayan açık kaynak projeler geliştirilmektedir. Bunlardan Open Quantum Safe (OQS) projesi PQC algoritmaların yazılımsal geliştirmesini, test edilmesini ve entegrasyonunu sağlayan bir açık kaynak yazılımıdır. OQS sayesinde PQC algoritmaları OpenSSL gibi TLS protokolünü uygulayan ve geniş çapta kullanılan şifreleme kütüphanelerinde test edilebilmektedir.

Bu tezin konusu ise; PQC algoritmaların TLS protokolüne entegrasyonunun nasıl sağlanacağı ve TLS 1.3 protokolü güvenliğinin artırılması için gerekli stratejilerdir. Yöntem olarak, NIST tarafından seçilen PQC algoritmalar üzerinde yoğunlaşılmıştır. Bu algoritmaların uygulanabilirliğinin görülmesi için bir test ortamı kurulmuştur. Test ortamında OQS alt yapısı temel alınmış ve TLS 1.3 protokolünde seçilen PQC algoritmalarının performans açısından incelenmesi sağlanmıştır.

Ek olarak, bu çalışma TLS 1.3 altında post-kuantum algoritmaların performansını sayısal olarak ortaya koyarken, sertifika ekosistemine geçişin yalnızca teknik hazırlıkla sınırlı olamayacağını da vurgular. Çok paydaşlı yapısı nedeniyle birlikte çalışabilirliğin sağlanması oldukça önemlidir. X.509 sertifika profillerinin güncellenmesi, tarayıcı/işletim sistemi desteğinin sağlanması ve CA/Browser Forum yönlendirmeleri doğrultusunda sertifika otoritelerinin PQC tabanlı sertifikalar yayımlamaya teşvik edilmesi dönüşüm için gereklidir.

1.2 Tezin Amacı ve Kapsamı

Bu tez çalışmasının temel amacı, kuantum bilgisayarların oluşturduğu tehdit karşısında TLS protokolünün nasıl daha güvenli hâle getirilebileceğine dair araştırmalar yapmak ve post-kuantum kriptografi (PQC) algoritmalarına geçiş hakkında çözüm önerileri sunmaktır. Bu kapsamda bir PQC test ortamı oluşturulmuş ve PQC algoritmalar TLS protokolünde test edilerek değerlendirilmiştir. Özellikle PQC geçiş sürecinde NIST tarafından belirlenen ve standartlaştırma sürecinde olan PQC algoritmalarının performans açısından incelenmesi hedeflenmiştir.

Çalışmada, OpenSSL kriptografik kütüphanesinde post-kuantum algoritmaların uygulanmasını mümkün kılan Open Quantum Safe (OQS) projesinin bir parçası olan liboqs kütüphanesi kullanılmıştır. OQS-Provider aracılığıyla Kyber, Dilithium, Falcon ve SPHINCS+ gibi algoritmaların TLS 1.3 protokolünde kullanılarak, bu algoritmalar hem tekil hem de hibrit kullanım senaryolarında test edilmiştir. Testler sırasında istemci-sunucu yapılandırmasına dayanan bağlantılar oluşturulmuştur. Farklı algoritmalarda bağlantının başarı durumu, el sıkışma süresi gibi parametreler üzerindeki değişiklikler de incelenmiştir.

Bu tezin bir diğer amacı da belirli algoritmaları işaret etmekten ziyade yapılan analizler ve performans ölçümleri aracılığıyla PQC geçiş sürecine katkı sağlamak ve kriptografik değişim sürecinde kullanılabilecek geçiş stratejilerini de ele almaktır. Elde edilen bulguların post-kuantum dönemde TLS protokolüne geçişlerde izlenmesine yardımcı olması hedeflenmektedir.

1.3 Bölüm Yapısı

Bu tez toplam beş bölümden oluşmaktadır:
Birinci bölümde, çalışmanın genel arka planı, kuantum hesaplamanın klasik kriptografi üzerindeki etkisi ve post-kuantum kriptografiye duyulan ihtiyaç detaylı şekilde

açıklanmıştır. Ayrıca, NIST'in PQC algoritmaları ile ilgili standardizasyon süreci ile Open Quantum Safe gibi açık kaynak girişimlere de yer verilmiştir.

İkinci bölümde, Transport Layer Security (TLS) protokolünün yapısı, el sıkışma (handshake) mekanizması ve günümüzde kullanılan kriptografik algoritmalar detaylı biçimde incelenmiştir. TLS 1.2 ve TLS 1.3 sürümleri karşılaştırmalı olarak ele alınmıştır. Post-Kuantum Kriptografi Algoritmaları teorik olarak anlatılmış ve bunların standardizasyon süreçlerine değinilmiştir. Son olarak bu bilgilerden yola çıkarak TLS protokolünde PQC kullanımına teorik olarak değinilmiştir.

Üçüncü bölümde, deneysel test ortamının kurulumu ve test yöntemi detaylı şekilde açıklanmıştır. Test senaryoları detaylı şekilde sunulmuştur. Kullanılan PQC algoritmalar, bunların seçim nedenleri, kullanılan sertifikaların özellikleri ve el sıkışma parametreleri ayrıntılı olarak bu bölümde verilmiştir.

Dördüncü bölüm, gerçekleştirilen testlerin bulgularını ve analizlerini içermektedir. Farklı PQC algoritmaları için el sıkışma süreleri, sertifika boyutları ve bağlantı başarı oranları grafiklerle desteklenerek değerlendirilmiştir. Bu bölümde PQC algoritma varyantları kendi içinde değerlendirildiği gibi klasik, hibrit ve saf PQC anahtar değişim senaryoları arasında karşılaştırmalı analizlere de yer verilmiştir.

Beşinci ve son bölümde ise tez kapsamında elde edilen bulgulara göre post-kuantum TLS geçişine ilişkin genel değerlendirmelere yer verilmiştir. Ayrıca, çalışmanın sınırları belirtilmiş ve PQC geçişinde TLS sertifika ekosistemine düşen aksiyonlara da değinilmiştir.

1.4 Literatür Özeti

PQC alanında dünya genelinde oldukça çok sayıda araştırma çalışmaları mevcuttur. Bu araştırmalar güvenlik analizlerini içermekle beraber pratikte kullanılan gerçek dünya uygulamaları ile entegrasyonlara da yoğunlaşmaktadır. Literatürde hem teorik hem de

pratik deneyleri içeren çalışmalar da mevcuttur. PQC algoritmalarının performans, anahtar ve imza boyutları ve hesaplama süreleri gibi metrikler açısından analiz edildiği çeşitli akademik yayınlar araştırılmıştır. Bu yayınlarda algoritmaların anahtar boyutları, imza uzunlukları ve hesaplama süresine odaklanan metrikler analiz edilmektedir. Bu tez kapsamında gerçekleştirilen deneysel performans analizleri de literatürdeki güncel çalışmalarla karşılaştırıldığında benzer eğilimler göstermektedir. Özellikle TLS 1.3 protokolü altında post-kuantum algoritmalarla yapılan el sıkışma ve kimlik doğrulama testleri genel çıkarımlarla örtüşmektedir.

Sosnowski ve arkadaşlarının çalışma sonuçları, Kyber ve HQC gibi post-kuantum anahtar kapsülleme algoritmalarının performans açısından klasik sistemlerle kıyaslanabilir düzeyde seyrettiğini göstermektedir. Öte yandan, Dilithium ve Falcon gibi dijital imza algoritmaları yalnızca kullanılabilir değil, aynı zamanda birçok kullanım senaryosunda daha hızlı sonuçlar üretebilmektedir. Hibrit algoritmaların uygulanmasında gözle görülür bir performans kaybı yaşanmamış, hatta yüksek güvenlik seviyelerinde bu algoritmaların klasik çözümlere kıyasla daha başarılı olduğu tespit edilmiştir. Bu bulgular, post-kuantum algoritmaların TLS gibi zaman açısından kritik uygulamalara teknik olarak entegre edilebileceğini ortaya koymakta ve kuantum sonrası TLS'nin günümüzdeki sistemlere uyarlanmasında pratik bir engel bulunmadığını göstermektedir (Sosnowski et al., 2022).

Alkim ve arkadaşlarının (2020) gerçekleştirdiği deneysel çalışma da bu bulguları destekler niteliktedir. Araştırmada, PQ dijital imza algoritmalarının TLS 1.3 protokolüne entegrasyonu değerlendirilmiş ve gerçekçi ağ koşullarında yapılan testlerde, istemci tarafından gözlemlenen el sıkışma süreleri ile kimlik doğrulamalı sunucuların genel verimliliği analiz edilmiştir. Özellikle imza ve sertifika zinciri boyutlarının, hızlı doğrulama sunan algoritmalar için el sıkışma süresine doğrudan etki ettiği gösterilmiştir. Çalışma, Dilithium ve Falcon'un zaman duyarlı uygulamalar için en uygun PQC algoritmalarından olduğunu ortaya koyarken, Falcon'un kayan nokta donanım desteği ile web uygulamaları için öne çıkabileceği vurgulanmıştır. Ayrıca, farklı imza algoritmalarının bir arada kullanıldığı sertifika zincirlerinin performansa olumlu katkı sağlayabileceği gösterilmiştir.

Paquin, Stebila ve Tamvada (2020) ise PQC algoritmalarının performansını değerlendirmek amacıyla ağ gecikmesi ve paket kaybı gibi değişkenleri kontrol eden bir deneysel çerçeve geliştirmiştir. Hibrit anahtar değişim yapılarının ve post-kuantum imzaların TLS bağlantı sürecine etkisi incelenmiş, özellikle %3'ün üzerindeki paket kayıplarının belirli algoritmalarda performans sorunlarına yol açabileceği gösterilmiştir. Ayrıca, bazı algoritmaların yavaş hesaplama sürelerinin ağ gecikmesi tarafından maskelendiği, dolayısıyla test ortamının gerçekçilik düzeyinin sonuçlar üzerinde belirleyici olabileceği tespit edilmiştir. Bu durum, algoritma seçiminde sadece işlem sürelerinin değil, ağ koşullarının da dikkate alınması gerektiğini ortaya koymaktadır.

Son olarak, Boehme ve arkadaşları (2023) tarafından önerilen karma (mixed) sertifika zinciri yaklaşımı, klasik ve PQC algoritmalarının birlikte kullanılabileceği bir geçiş modeli sunmaktadır. Çalışma, TLS 1.3 içerisinde klasik RSA/ECDSA kök sertifikaları ile PQC imzalı ara ve uç sertifikaların uyum içinde çalışabileceğini göstermiştir. Farklı platformlar üzerinde yapılan doğrulama testleri, bu yapının pratikte uygulanabilir olduğunu ve geçiş sürecine esneklik kazandırabileceğini ortaya koymaktadır. Ayrıca, sertifika zincirindeki algoritma çeşitliliğinin sistem uyumluluğu ve güvenlik açısından nasıl yönetileceği konusu da detaylı şekilde analiz edilmiştir.

Bu tez çalışmasında post-kuantum kriptografik algoritmaların TLS 1,.3 protokolü içinde ne kadar başarılı ve uygulanabilir olduğunu değerlendirmek için test senaryoları geliştirilmiş ve bu senaryoların PQC test ortamında gerçekleştirilmiştir. Başlangıçta PQC entegrasyonu için kimlik doğruma ve anahtar değişimi olmak üzere iki temel çatıda senaryolar ele alınmış ve her bir senaryoda diğer bağlantı parametreleri sabit tutularak algoritmanın etkisi ortaya çıkarılmıştır. Çalışma teorik yaklaşımlar ile pratik deneyleri birleştirerek özgün bir uygulama örneği sunar ve mevcut standartların ötesine geçerek geleceğin güvenli iletişim altyapısına katkı sağlar.

2. KAYNAK ÖZETLERİ ve KURAMSAL TEMELLER

2.1 TLS Protokolü

Transport Layer Security (TLS), internet üzerinde gerçekleşen veri iletişimini güvenli hale getirmek için kullanılan kriptografik protokoldür. TLS protokolü ilk olarak SSL (Secure Sockets Layer) adıyla Netscape tarafından geliştirilmiş, daha sonra Internet Engineering Task Force (IETF) tarafından standardize edilerek TLS ismini almıştır. Protokolün en güncel ve yaygın sürümü olan TLS 1.3, RFC 8446 (The Transport Layer Security (TLS) Protocol Version 1.3) kapsamında tanımlanmıştır (Dierks ve Rescorla, 2018). Özellikle web trafiği (HTTPS), e-posta trafiği (IMAP, SMTP) ve sanal güvenli ağ (VPN) uygulamaları gibi çeşitli alanlarda kullanılan TLS, güvenli iletişim kanalı oluşturmada yaygın olarak kullanılır. TLS, sunucu ve istemci arasındaki iletişimi korur. TLS hem sunucu ve istemci kimliğini doğrulamak hem de iletilen verilerin bütünlüğünü ve gizliliğini sağlamak için simetrik ve asimetrik kriptografiyi birlikte kullanır. Veriler şifrelenerek iletim sırasında üçüncü tarafların içeriği okuması engellenir ve böylece gizlilik sağlanmış olur. İletilen uygulama verileri üzerinde bir değişiklik yapılmadığını garantilemek için dijital imzalar ve MAC kullanılır, böylece bütünlük sağlanır. Sunucunun ve ihtiyaç halinde istemcinin kimliği ise güvenilir TLS sertifikaları yolu ile doğrulanabilir. TLS sertifikaları, sunucu açık anahtarını, sahibi ile ilgili bilgileri ve sertifikayı imzalayan sertifika otoritesinin (Certificate Authority, CA) dijital imzasını içerir. İstemci, sertifikaya dayalı kimlik doğrulamada sertifikayı doğrular.

TLS ile güvenliği sağlanacak olan sunucu ve istemci arasında oluşturulan bağlantıya TLS oturumu (TLS session) denir. Bir oturumunun başlatılması, başlangıçta el sıkışması (handshake) olarak adlandırılan süreçle başlar. Bu süreçte taraflar kullanılacak TLS oturumunun nasıl kurulacağını belirler. Örneğin, TLS sürümü, veri alışverişinde hangi anahtar kullanılacak ve anahtarın nasıl kullanılacağı, kimlik doğrulama sertifikaları gibi hususlar üzerinde anlaşır. Sunucu ve istemci örneğinde; el sıkışmasını istemci başlatır. Sunucu ve istemci TLS oturumunun güvenliğini garanti altına alacak kriptografik işlemlerde kullanılacak algoritmalar üzerinde bir anlaşmaya varır. Anahtar değişimi (key exchange), kimlik doğrulama (authentication) ve kullanılacak dijital imzalar (digital

signatures), tüm verilerin şifrelenmesi (bulk encryption) ve özetleme (hashing) için anlaşma sağlanmalıdır. Sunucu kimliği sertifika aracılığı ile doğrulanır ve bu sertifikanın doğrulanması istemci tarafından yapılır. TLS çift taraflı kimlik doğrulamaya da izin vermektedir. İstemcinin sertifika sunduğu durumda sunucu tarafından sertifika doğrulanmalıdır (mutual TLS). El sıkışması sürecinin ardından uygulama süreci başlar. TLS oturumu başlatılmıştır, uygulamalar istemci ve sunucu arasında veri iletişimini güvenli bir tünel olarak görülebilecek TLS oturumu aracılığı ile yaparlar. (Anonymous, 2021).

2.2 TLS 1.3 Protokolü ve El Sıkışma Süreçleri

TLS protokolü ilk kullanılmaya başladığından beri yıllar içerisinde değişen güvenlik tehditlerine ve performans ihtiyaçlarına yanıt verebilmek amacıyla değişikliklere uğramıştır. TLS 1.2 sürümü, uzun süre kabul gören standart olarak kullanılmış ancak zamanla ortaya çıkan saldırı türleri ve karmaşık yapıda olmasının neden olduğu güvenlik açıkları nedeniyle, daha sade ve güvenli bir sürüm ihtiyacı doğmuştur (Rescorla, 2008). Bu doğrultuda geliştirilen TLS 1.3, 2018 yılında IETF tarafından RFC 8446 ile standartlaştırılmıştır (Rescorla, 2018).

TLS 1.3'te güvenliği ve performansı artırmaya yönelik TLS 1.2'ye göre yapılan büyük değişikliklerden bazıları aşağıda özetlenmiştir:

İleri gizliliğin zorunlu hale getirilmesi: Statik RSA ve Diffie-Hellman (DH) yöntemleri, uzun dönemli anahtarların gizli kalmasına dayandığından ileri gizlilik (forward secrecy) sağlayamamaktadır. Bir saldırgan gelecekte bu uzun dönemli anahtarları ele geçirirse, geçmiş trafikte iletilen verileri çözebilir. Bu nedenle statik RSA ve Diffie-Hellman şifreleme süitleri protokolden çıkarılmış, yerlerine ECDHE (Elliptic Curve Diffie-Hellman Ephemeral) ve DHE (Diffie-Hellman Ephemeral) gibi geçici (ephemeral) anahtar değişim yöntemleri eklenmiştir (Anonymous, 2024). Zayıf algoritmaların kaldırılması: RC4, MD5, SHA-1 gibi kriptografik olarak zayıf veya kırılması mümkün hale gelmiş algoritmalar kullanımdan tamamen kaldırılmıştır. Bu adım, protokolün modern saldırı türlerine karşı dayanıklılığını artırmıştır.

El sıkışma sürecinin sadeleştirilmesi: Handshake süreci sadeleştirilerek mesaj sayısı azaltılmış ve böylece gecikme süresi azaltılarak oturum başlatma hızı artırılmıştır.

AEAD (AEAD- Authenticated Encryption with Associated Data) algoritmalarının zorunlu hale getirilmesi: Simetrik şifreleme algoritmaları listesi, yalnızca İlişkili Verilerle Kimlik Doğrulanmış Şifreleme (AEAD- Authenticated Encryption with Associated Data) algoritmalarını içerecek şekilde sadeleştirilmiştir. AEAD algoritmaları, kriptografide gizlilik ve bütünlüğü tek bir işlemde eşzamanlı olarak sağlamak amacıyla kullanılmaktadır. Geleneksel simetrik şifreleme yöntemlerinden farklı olarak AEAD, veriyi şifrelerken aynı zamanda mesajın kimliğini de doğrulayan bir bütünlük denetimi gerçekleştirir.

Cipher suite yapısının yeniden düzenlenmesi: Cipher suite kavramı yeniden yapılandırılmış, kimlik doğrulama ve anahtar değişim mekanizmaları, kayıt şifreleme algoritmasından ve özetleme algoritmasından ayrıştırılarak daha modüler bir yapı benimsenmiştir (AppViewX, 2025).

Bu dönüşümler, TLS protokolünün yalnızca geçmişteki zayıflıkları gidermesiyle kalmayıp, aynı zamanda kriptografik çevikliğin sağlanması amacıyla gelecekte karşılaşılabilecek tehditlere karşı daha esnek, sade ve güvenli bir yapıya dönüştüğünü göstermektedir. Böylece TLS, modern şifreleme algoritmalarının entegrasyonu ve değişen güvenlik ihtiyaçlarına hızlıca uyum sağlayabilecek bir altyapı kazanmıştır (Anonymous, 2021). Uyumlanabilir yapının en önemli faydası, TLS'i esnek hale getirmesi ve gelişen kriptografi ihtiyaçlarına cevap verebilecek şekilde tasarlanmasıdır.

2.3 TLS 1.3'te Güvenli İletişimin Temelleri

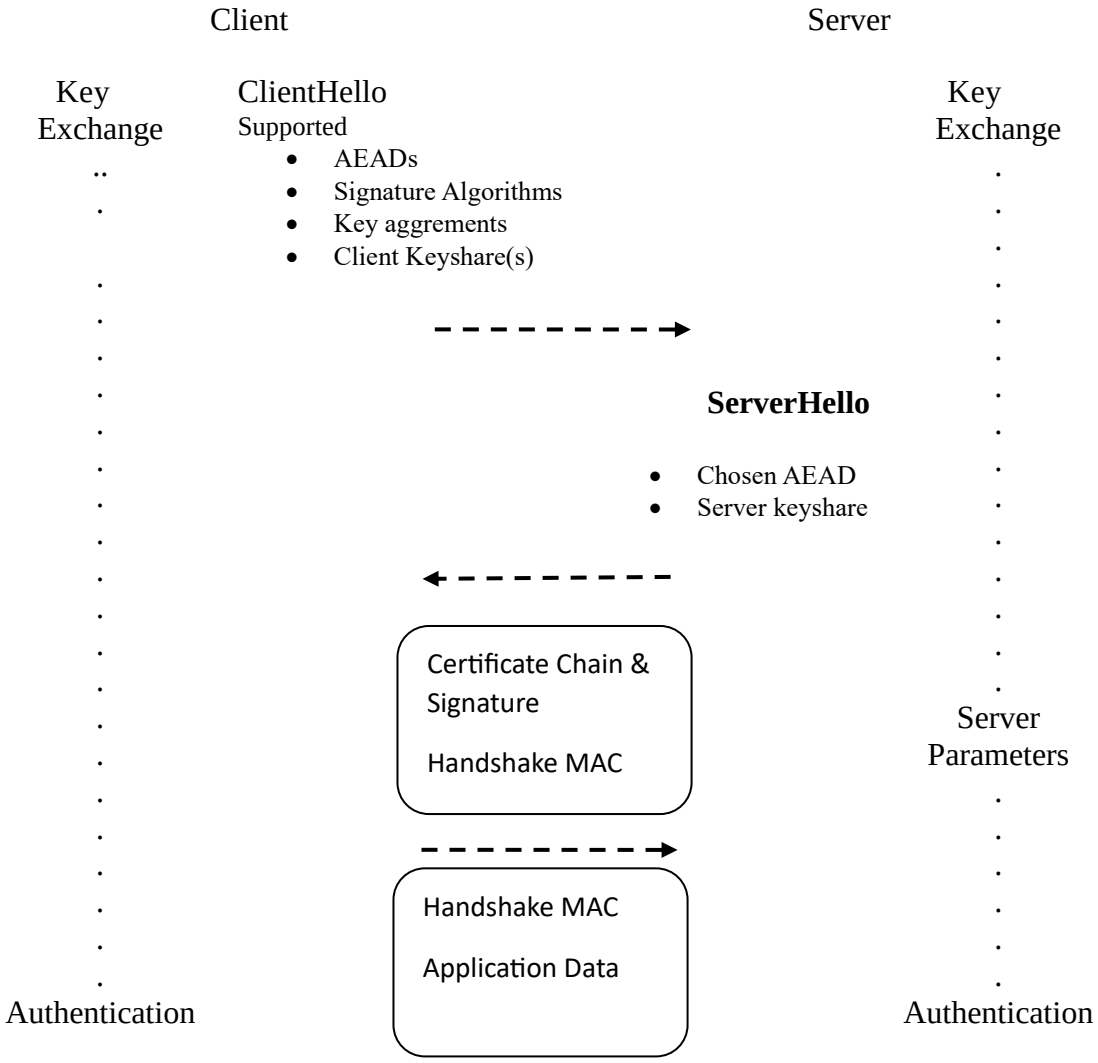
TLS protokolünde, kullanılacak kriptografik parametreler el sıkışma (handshake) süreci ile belirlenir. Bu alt protokol, istemci ve sunucunun ilk bağlantı kurduğu anda devreye girer. El sıkışma protokolü; tarafların kullanılacak TLS sürümü üzerinde uzlaşmasını, kriptografik algoritmaların seçilmesini, gerekirse kimlik doğrulamasının yapılmasını ve

ortak bir gizli anahtar materyalinin oluşturulmasını sağlar. El sıkışma süreci başarıyla tamamlandıktan sonra, uygulama katmanındaki veriler bu süreçte elde edilen anahtarlar ile şifrelenerek korunur. El sıkışma aşamasında meydana gelebilecek herhangi bir hata veya protokol ihlali durumunda bağlantı sonlandırılır; bazı durumlarda bu sonlandırma, bir uyarı mesajı (alert) ile gerçekleşir (Rescorla, 2018). RFC 8446'da belirtildiği ve Şekil 2.1 TLS 1.3 El sıkışma süreci'nde görüldüğü üzere el sıkışma üç aşamalı olarak gerçekleştirilmektedir.

1- Anahtar Değişimi (Key Echange): Anahtar anlaşması, anahtar değişimi veya anahtar dağıtımı olarak da adlandırılabilmektedir. Amacı iki tarafın da bir dinleyici hiçbir şey öğrenemeden paylaşımlı bir anahtar üzerinde anlaşabilmesini sağlamak olan kriptografik protokoldür. Paylaşılan anahtarlama materyalinin oluşturulması ve kriptografik parametrelerin seçilmesi bu aşamadadır. Bu aşamada istemci rasgele bir nonce (ClientHello.random) değeri, teklif edilen protokol versiyonları bilgileri, şifreleme süitleri listesi, key_share uzantısı içinde bir dizi Diffie-Hellman anahtar paylaşımı veya önceden paylaşılan anahtar etiketleri (pre_shared_key) ve ek alanlardan oluşan bir 'ClientHello' mesajı gönderir. Sunucu ClientHello mesajını işler ve bağlantı için uygun kriptografik parametreleri belirler. Daha sonra anlaşmaya varılan bağlantı parametrelerini gösteren ServerHello mesajını oluşturur ve bununla yanıt verir. TLS 1.3'te ClientHello ve ServerHello mesajlarının ardından gerçekleşen anahtar değişimi işlemi, güvenli iletişimin temelidir. Bu süreçte kullanılan temel mekanizmalar:
   - ECDHE (Elliptic Curve Diffie-Hellman Ephemeral): Her oturum için yeni bir anahtar çifti üretir. Eliptik Eğri tabanlı olduğu için hem güvenli hem de performanslıdır. Hem istemci hem sunucu kendi ECDHE anahtarlarını ClientHello ve ServerHello mesajlarında key_share uzantısı ile paylaşır. Ortak bir anahtar (shared secret) bu bilgiden hesaplanır ve HMAC-tabanlı Extract-and-Expand Key Derivation Function (HKDF) kullanılarak oturum anahtarları elde edilir. (Krawczyk, Eronen ve Arkko, 2010).
   - PSK (Pre-Shared Key) önceden paylaşılmış bir anahtarın (örneğin yapılandırma dosyası vb.) istemci ve sunucu tarafından bilindiği durumlar içindir. Bu durumda key_share uzantısı gerekli olmayabilir. ClientHello mesajında istemci, PSK ile

bağlantı kurmak istediğini belirtir. Sunucu da ServerHello mesajıyla bunu onaylarsa, her iki taraf aynı PSK üzerinden türetilmiş anahtarlarla el sıkışma yapar.

- TLS 1.3 ayrıca PSK + ECDHE kombinasyonunu da destekler. Buna 'PSK with (EC)DHE' denir. PSK'nın istemci kimliğinin ispatı ve hızlı bağlantı için tercih edildiği bu kombinasyonda ECDHE ile ileri gizlilik sağlanmış olur.



Şekil 2.1 TLS 1.3 El sıkışma süreci

2- Sunucu Parametreleri (Server Parameters): Sunucu tarafından gönderilen 'EncryptedExtensions' ve 'CertificateRequest' mesajlarını kapsar. Kalancinin kimliğinin doğrulanıp doğrulanmayacağı, uygulama katmanı protokol desteği gibi el sıkışmanın geri

kalan bilgilerini içermektedir. Bu mesajlar da ‘server_handshake_traffic_secret’ tan türetilen anahtarlar ile şifreli olarak gönderilir.

3- Kimlik Doğrulama (Authentication): Kimlik doğrulama aşamasında ‘Certificate’, ‘CertificateVerify’ ve ‘Finished’ mesajları ile kimlik doğrulama(authentication), anahtar onayı (key confirmation) ve el sıkışma bütünlüğü (handshake integrity) sağlanır. Bu süreç belirli girdilere dayanarak gerçekleştirilen standart bir süreçtir. Bu süreçte kullanılan başlıca girdiler şunlardır:

- Doğrulama için kullanılacak sertifika,
- El sıkışma sürecinde daha önce iletilmiş mesajlardan oluşan Handshake Context (ileti geçmişinin özeti),
- Anahtar türetimi amacıyla kullanılan Base Key (temel anahtar).

TLS, bu girdilerden hareketle TLS 1.3 el sıkışmasında üç temel doğrulama mesajı oluşturur. Sertifika tabanlı kimlik doğrulamanın gerekli olduğu her seferinde aynı mesaj seti kullanılır. Bunlar:

- ‘Certificate’ mesajı uç noktanın sertifikası ve sertifika başına uzantıları içerir. Sunucuya veya istemciye ait dijital sertifika ve gerekli destekleyici sertifikalar iletilir. Ancak PSK tabanlı el sıkışmalarda, özellikle 0-RTT gibi durumlarda, sertifikaya dayalı istemci kimlik doğrulaması yapılmaz. Eğer sertifika ile kimlik doğrulaması yapılmıyorsa ve sunucu tarafından ve sunucu ‘CertificateRequest’ mesajı göndermediyse istemci tarafından atlanır. ‘Certificate’ mesajı ham açık anahtarların veya önbelleğe alınmış bilgi uzantılarının kullanılmadığı durumda sertifika değil sunucunun uzun vadeli anahtarına karşılık gelecek başka değerler de içerebilir. ((Wouters vd., 2014), (Kee vd., 2016)).
- ‘CertificateVerify’, ‘Certificate’ mesajındaki sertifikanın geçerliliğinin doğrulanması amacıyla bu sertifika açık anahtarına karşılık gelen özel anahtar kullanılarak ‘Handshake Context’ ve ‘Certificate’ mesajının özetinin imzalanması ile oluşturulur. Ek olarak bu noktaya kadar handshake bütünlüğünün korunduğunu da garanti eder. Uç nokta bir sertifika aracılığıyla kimlik doğrulaması yapmıyorsa bu mesaj da atlanır.
- ‘Finished’ mesajı Authentication bloğunun son mesajıdır. El sıkışma sürecinde iletilen tüm mesajların özet değeri üzerine HMAC (Hash-based MAC) uygulanarak mesaj

bütünlüğü sağlanır. Bu işlemde kullanılan MAC anahtarı, ilgili temel anahtardan (Base Key) türetilir.

Bu noktada, el sıkışma tamamlanmış ve istemci ile sunucu arasında kimlik doğrulamalı şifreleme yoluyla korunan uygulama katmanı verilerini değiş tokuş etmek için kayıt katmanının gerektirdiği anahtarlama malzemesi türetilmiştir. 'Finished' mesajından sonraki her kayıt artık şifreli şekilde (application traffic key kullanılarak) gönderilmelidir.

2.4 Post-Kuantum Kriptografi Algoritmaları

Kuantum bilgisayarlar klasik kriptografik sistemleri tehdit etmektedir. Bu durum günümüzde kullanılan bilgi güvenliği prensiplerinin yeniden gözden geçirilmesi gerektiği sonucunu doğurmaktadır. Halihazırda yaygın olarak kullanılan RSA, DSA, ECDSA, ECDH gibi algoritmaların güvenliği, asal çarpanlara ayırma veya eliptik eğri ayrık logaritma problemlerinin zor matematiksel problemler olmasına dayanır. Bunların klasik bilgisayarlarca çözülmesinin zor olduğu bilinmektedir. Ancak bu durum kuantum bilgisayarlar için geçerli değildir.

Shor Algoritması, kuantum bilgisayarların klasik kriptografik sistemler üzerindeki en bilinen tehdittir. Shor, asal çarpanlara ayırma ve ayrık logaritma problemlerini polinomsal sürede çözebilen bir kuantum algoritması geliştirmiştir. Bu algoritmanın etkisi, RSA, DSA, ElGamal, DH, ECDSA ve ECDH gibi mevcut açık anahtarlı sistemlerin çoğunun temel varsayımlarını geçersiz kılmasıyla ortaya çıkar. Yeterli sayıda qubit'e sahip kuantum bilgisayarın gerçekleşmesi durumunda, bu algoritmaların güvenliği tamamen ortadan kalkacaktır [(Shor, 1994); BSI, 2022].

Grover Algoritması, simetrik anahtar şifreleme algoritmalarına karşı daha sınırlı fakat yine de önemli bir kuantum hızlandırma sağlamaktadır. Grover algoritması, bir arama problemini klasik yöntemlere kıyasla karekök sürede çözebilir. Bu, örneğin 2^n olasılığı olan bir anahtar uzayında, çözümün klasik olarak $O(2^n)$ yerine $O(2^{n/2})$ sürede bulunabilmesini sağlar. Dolayısıyla AES-128 gibi simetrik algoritmaların anahtar boyutu

yükseltilerek kuantum saldırılarına karşı daha güvenli varyantların (ör. AES-256) kullanılması önerilir. [(Grover, 1996); BSI, 2022]

Harrow–Hassidim–Lloyd (HHL) Algoritması, doğrusal denklem sistemlerinin çözümünü kuantum ortamında hızlandırmak üzere geliştirilen bir diğer kuantum algoritmadır. Her ne kadar doğrudan kriptografiye karşı bir tehdit oluşturmasa da bazı kriptanaliz yöntemlerinde potansiyel hızlandırıcı etki yaratabileceği değerlendirilmektedir. HHL algoritması, özellikle kuantum makine öğrenmesi ve optimizasyon problemlerinde öne çıkmaktadır. [(BSI, 2022)]

Bu nedenlerle hem açık anahtarlı hem de bazı simetrik kriptografik sistemler için farklı düzeylerde tehditler bulunmaktadır. Kuantum bilgisayarların pratikte kullanılabilme ihtimalinde, mevcut açık anahtarlı kripto sistemler büyük ölçüde güvenliğini kaybedecektir. Post-kuantum kriptografisi (PQC), klasik asimetrik şifrelemeye karşı gelebilecek potansiyel tehditleri ele almak için geliştirilen yeni kriptografik algoritmaları kapsamaktadır. PQC temelleri de yine çözülmesi zor matematiksel problemlere dayanır. Bu problemler; Lattice tabanlı problemler (LWE, SIS), Kod tabanlı problemler (hata düzeltme kodları), Hash tabanlı problemler (Merkle imzaları), Multivariate polinomial problemler ve Isogeny tabanlı problemler gibi temel kategorilerde toplanır.

PQC alanında birçok ülkede ve kuruluşta kuantum bilgisayarlarının oluşturabileceği tehdide karşı kriptografik sistemleri güçlendirmek için çalışmalar yapılmaktadır. Bu çalışmalar PQC algoritmaların değerlendirme ve standartlaştırma süreçlerini içerir. Avrupa Birliği bünyesinde ENISA (European Union Agency for Cybersecurity), Fransa'daki ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information) ve Almanya'daki BSI (Bundesamt für Sicherheit in der Informationstechnik) gibi ulusal siber güvenlik kurumları, PQC konusunda kendi değerlendirme raporlarını yayımlamakta ve stratejik görüşlerini paylaşmaktadır (ANSSI, 2022; BSI, 2018). Bu raporlar algoritma güvenliği, uygulama açısından zorluklar ve geçiş sürecine dair teknik bilgilendirmelerin yanında geçiş sürecine dair stratejik önerileri de içermektedir. Aynı şekilde Japonya, Güney Kore, Çin ve Kanada gibi ülkeler de PQC algoritmalarını test etmektedir. Bu alandaki en kapsamlı ve etkili girişim, Amerika Birleşik Devletleri Ulusal Standartlar ve

Teknoloji Enstitüsü (NIST) tarafından Kasım 2016'da başlatılan ve Aralık 2016'da resmi çağrısı yayımlanan Post-Quantum Cryptography Standardization Project'tir (NIST, 2016). Bu tez kapsamında incelenen algoritmalar, bu proje çerçevesinde NIST tarafından seçilmiş ve standartlaştırma süreci tamamlanmış ya da hâlen devam etmekte olan PQC algoritmalarından derlenmiştir. NIST PQC çalışmasında dünya genelinden gelen onlar PSK ca algoritma için birden çok adımlı bir değerlendirme süreci işletmiştir. Algoritmaların seçiminde güvenlik ana kriter olmasının yanı sıra performans, esneklik, mesaj ve anahtar boyutları, uygulamalar açısından uygulanabilirlik gibi diğer kriterler de dikkate alınmıştır. Süreç aşamalı olarak yürütülmüş ve her aşamada güvenlik analizi, performans testleri yapılmış ve geri bildirimleri değerlendirilmiştir.

NIST'in standardizasyon süreci, açık anahtarlı kriptografi alanında iki temel işlev üzerine odaklanmıştır:

- Anahtar Kapsülleme Mekanizmaları (KEM): Güvenli anahtar paylaşımı için
- Dijital İmza Algoritmaları: Kimlik doğrulama ve veri bütünlüğü için

Bu işlevlerin her ikisi de TLS gibi güvenli iletişim protokollerinde kritik bir rol oynamaktadır. 2024 yılında aşağıdaki algoritmalar Federal Bilgi İşlem Standardı (FIPS) olarak resmî biçimde yayımlanmıştır:

- FIPS 203: ML-KEM (Module-Lattice-based Key Encapsulation Mechanism, önceki adıyla CRYSTALS-Kyber)
- FIPS 204: ML-DSA (Module-Lattice-based Digital Signature Algorithm, önceki adıyla CRYSTALS-Dilithium)
- FIPS 205: SLH-DSA (Stateless Hash-based Digital Signature Algorithm, önceki adıyla SPHINCS+)

Her algoritma, farklı güvenlik seviyelerine karşılık gelen parametrik varyantlar sunmakta olup hem yazılım hem de donanım uygulamaları için optimize edilmiş tasarımlara sahiptir. Sonuç olarak bu standartlaştırma süreci hem kuantum güvenli anahtar paylaşımı hem de kuantum dirençli dijital imza çözümleri ile post-kuantum geçiş süreci için temel bir çalışmadır.

2.4.1 Kafes tabanlı Kriptografi (Lattice-Based Cryptography)

Kafes tabanlı kriptografi (lattice-based cryptography), post-kuantum kriptografinin önde gelen alanı olup hem anahtar değişim hem de dijital imza algoritmaları için PQC algoritmalar sunmaktadır. Bu algoritmalar, matematiksel kafesler üzerinde tanımlı çözülmesi zor kabul edilen, "NP-zor" (Nondeterministic Polynomial time-hard) sınıfına ait olan problemlere dayanır, çözümleri hem klasik hem de kuantum bilgisayarlar için zordur.

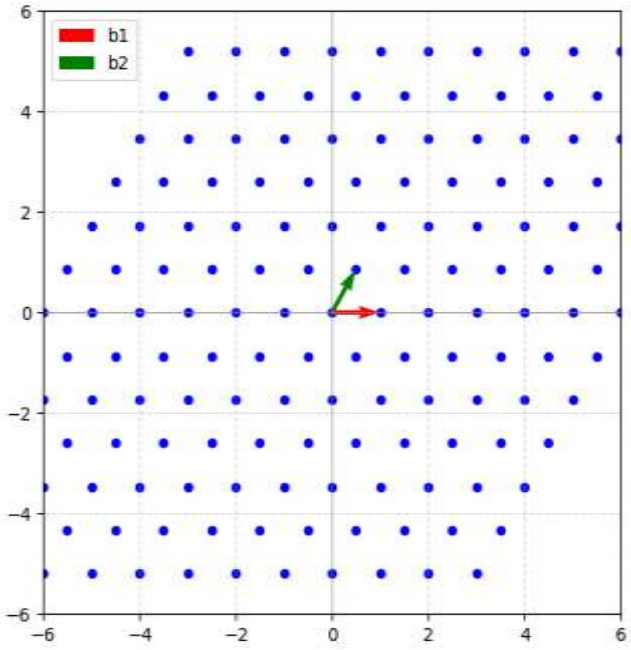
Matematiksel olarak bir kafes (lattice), $\mathbb{R}^n$ uzayında doğrusal bağımsız vektörlerden oluşan bir bazis yardımıyla, bu vektörlerin tam sayı katsayılarıyla alınan tüm lineer kombinasyonlarının oluşturduğu noktaların kümesidir. Başka bir deyişle, bir kafes Λ, bir bazis matrisi B ve tam sayı katsayılardan oluşan bir vektör x ile aşağıdaki şekilde tanımlanır:

$$\Lambda = \{\sum_{i=1}^{n} x_i b_i \vee x_i \in Z\}$$

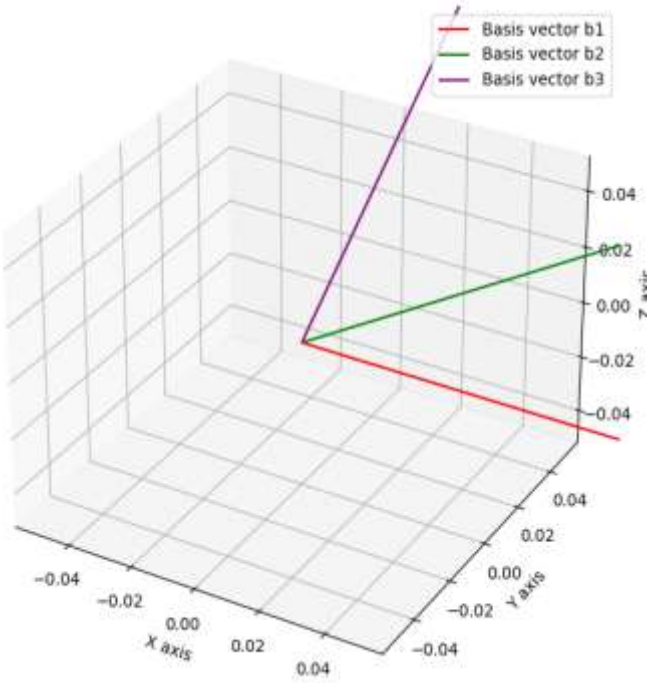
Burada:

- $xi \in \mathbb{Z}^n$: Tam sayı katsayılarından oluşan bir vektör
- $B = \{b_1, b_2, \ldots, b_n\}$: $n\ x\ n$ boyutlarında doğrusal bağımsız vektörlerden oluşan bir bazis matrisidir.

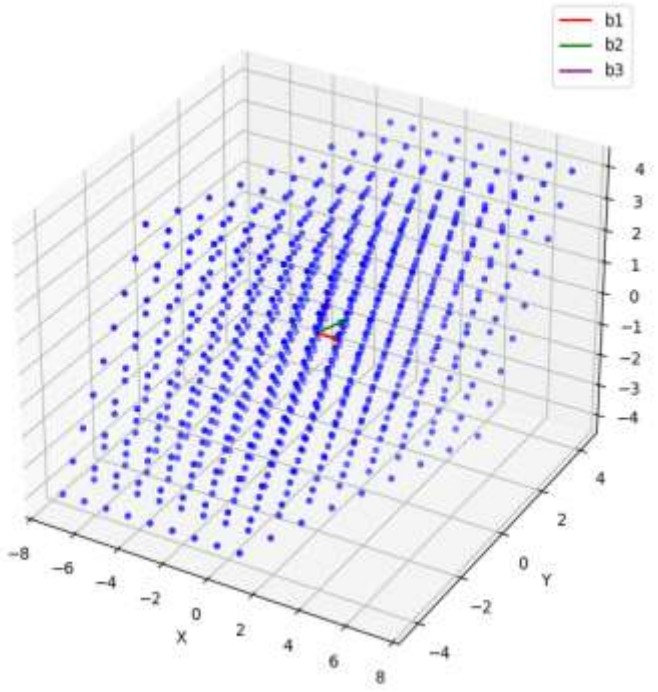
Şekil 2.2'de 2 boyutlu bir kafes gösterimi ve bazis vektörleri, Şekil 2.3 ve Şekil 2.4'te ise üç boyutlu bazis vektörleri ve bu vektörlerden elde edilebilecek bir kafes yapısı görülebilir. Burada, mavi noktalar kafes noktalarını, kırmızı ve yeşil oklar ise bazis vektörlerini temsil etmektedir.



Şekil 2.2 İki boyutlu kafes gösterimi ve bazis vektörleri



Şekil 2.3 Üç boyutlu kafes yapısı bazis vektörleri

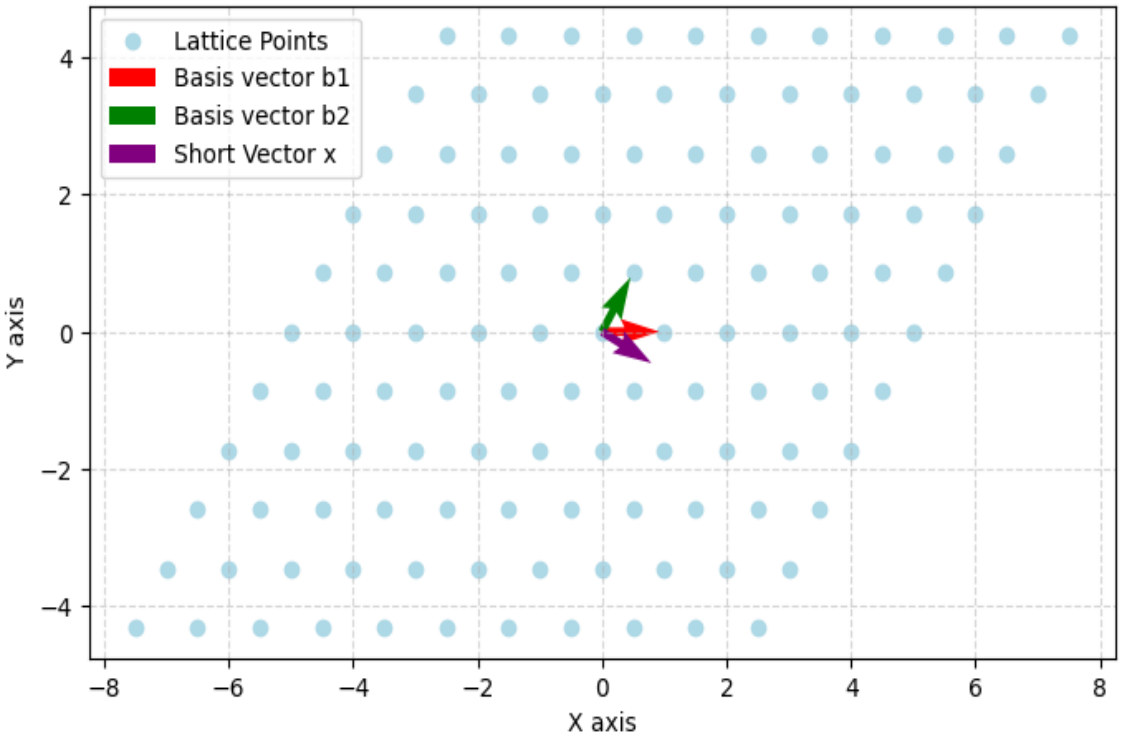


Şekil 2.4 Üç boyutlu kafes yapısı

Kafesler, kriptografide zor kabul edilen kısa vektör bulma (SVP – Shortest Vector Problem) ve en yakın vektör bulma (CVP – Closest Vector Problem) problemleriyle yakından ilişkilidir. Learning With Errors (LWE) problemi, temel olarak bir kafes noktasına rastgele bir hata eklendiğinde, orijinal noktanın (gizli vektörün) geri kazanılmasının zor olmasına dayanır. Matematiksel olarak bu durum şu şekilde ifade edilir:

$$b = A.s + e$$

Burada: A genellikle modüler bir matris, s gizli vektör (secret), e hata vektörü (noise) b ise gözlemlenen gürültülü vektördür. Şekil 2.5'te, $A.s$ ile gösterilen gerçek kafes noktası ve ona eklenen hata vektörü sonucu oluşan $b = A.s + e$ noktası arasındaki fark görselleştirilmiştir. Bu görselde, LWE probleminin temel mantığı görülebilir: gürültülü sinyalden hangi "temiz" kafes noktasının türetildiğini bulmak, matematiksel olarak oldukça zordur.



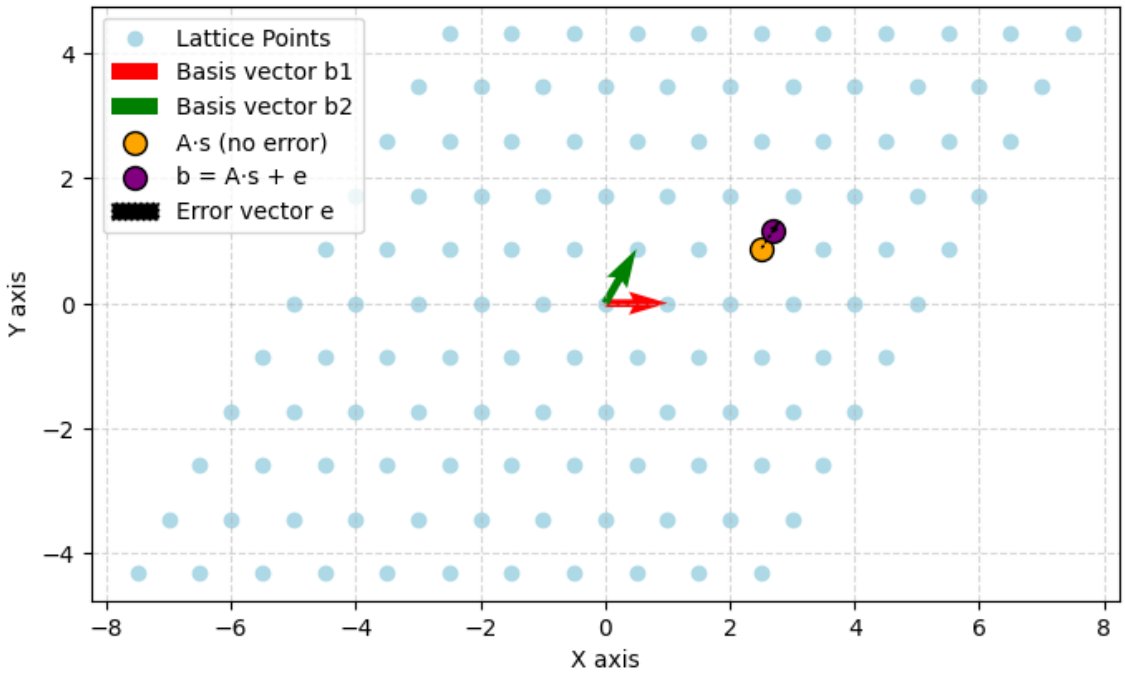
Şekil 2.5 LWE problemi temeli

LWE probleminin doğrudan uygulanması, büyük matris ve vektör işlemleri nedeniyle yüksek bellek ve işlem maliyeti getirir. Bu nedenle daha verimli varyantlar geliştirilmiştir. Bunlar:

Ring-LWE (R-LWE): Klasik LWE probleminin cebirsel halka ortamında yeniden tanımlanmış halidir. Tüm işlemler polinom halkası içinde gerçekleştirilir ve dairesel (circular) matris yapıları sayesinde matris çarpımı, Hızlı Fourier Dönüşümleri (NTT) gibi yöntemlerle çok daha verimli hesaplanabilir

Module-LWE (Mod-LWE), Ring-LWE probleminin daha genel bir biçimi olarak tanımlanır ve post-kuantum kriptografi alanındaki birçok modern algoritmanın, özellikle ML-KEM ve ML-DSA gibi NIST tarafından seçilen standartların, temelini oluşturan matematiksel zorluk problemidir. Ring-LWE, tek değişkenli polinom halkaları üzerinde tanımlanırken, Module-LWE, bu yapıyı birden fazla bileşenli modüllere genişleterek hem esneklik hem de daha iyi parametrik kontrol sunar. Bu genelleme performans iyileştirmesi sağlarken aynı zamanda güvenlik açısından da optimize edilebilir olmuştur.

Short Integer Solution (SIS): Bir kafesin içinde $A \cdot x \equiv 0 mod q$ denklemine uyan ve normu küçük olan bir x vektörünü temsil eder. Şekil 2.6'da ve yeşil oklar temel vektörleri, mor ok ise küçük normlu bir çözüme karşılık gelen x vektörünü göstermektedir. Burada, SIS probleminin temel zorluğunu görsel olarak ifade edilmiştir. Matematiksel olarak küçük normlu doğrusal kombinasyonların bulunması zordur. Tıpkı LWE' de olduğu gibi, SIS probleminin modül geliştirmeleri de (Mod-SIS) güncel kriptografik protokollerde kullanılmak üzere optimize edilmiştir.



Şekil 2.6 SIS problemi temeli

CRYSTALS-Kyber: CRYSTALS-Kyber algoritması, modül kafes yapısına dayalı ve CCA (Chosen Ciphertext Attack) atağına karşı güvenli bir anahtar kapsülleme mekanizması (KEM) olarak geliştirilmiş olup, post-kuantum kriptografiye yönelik en güçlü adaylardan biri olarak kabul edilmiştir. Algoritma, 2017 yılında NIST tarafından başlatılan Post-Quantum Cryptography Standardization Project kapsamında, "Kyber" adıyla değerlendirilmek üzere sunulmuştur. Kyber, Joint Submission Team adı altında çalışan uluslararası bir araştırma ekibi tarafından tasarlanmıştır. Bu ekipte yer alan isimler arasında Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler ve Damien Stehlé yer

almaktadır (Bos et al., 2017). Ekip, daha önce çeşitli lattice-tabanlı şemalar üzerine çalışmalarıyla bilinen araştırmacılardan oluşmaktadır. (Bos et al., 2017).

Güvenliğini Module-LWE (Mod-LWE) problemine dayandıran Kyber, aynı zamanda Ring-LWE yapılarından türetilen verimlilik avantajlarını da içinde barındıran hibrit bir tasarıma sahiptir. Bu sayede hem teorik güvenlik gereklerini karşılamakta hem de pratik uygulamalarda yüksek performans sunabilmektedir.

NIST tarafından yürütülen değerlendirme süreci sonucunda Kyber algoritması, 2022 yılında resmi olarak standartlaştırılması önerilen algoritmalar arasında yer almış ve nihayetinde 2024 yılında FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard dokümanı kapsamında "ML-KEM" adıyla yayımlanarak standart hâline gelmiştir.

Kyber algoritmasında taraflar arasında ortak bir gizli anahtar paylaşımı, beş temel aşamada gerçekleştirilir**.** 1. aşamada, Alice, anahtar üretimi için halka ortamında tanımlı rastgele bir matris A ve gizli vektör s' oluşturur. Burada açık anahtar olarak A, gizli anahtar olarak ise s' olarak belirlenmiştir. 2. Aşamada**;** Bob, Alice'in açık anahtarını A kullanarak kendi rastgele gizli vektörü s ile çarpma işlemi yapar ve küçük bir rastgele hata vektörü e ekleyerek şu ifadeyi hesaplar:

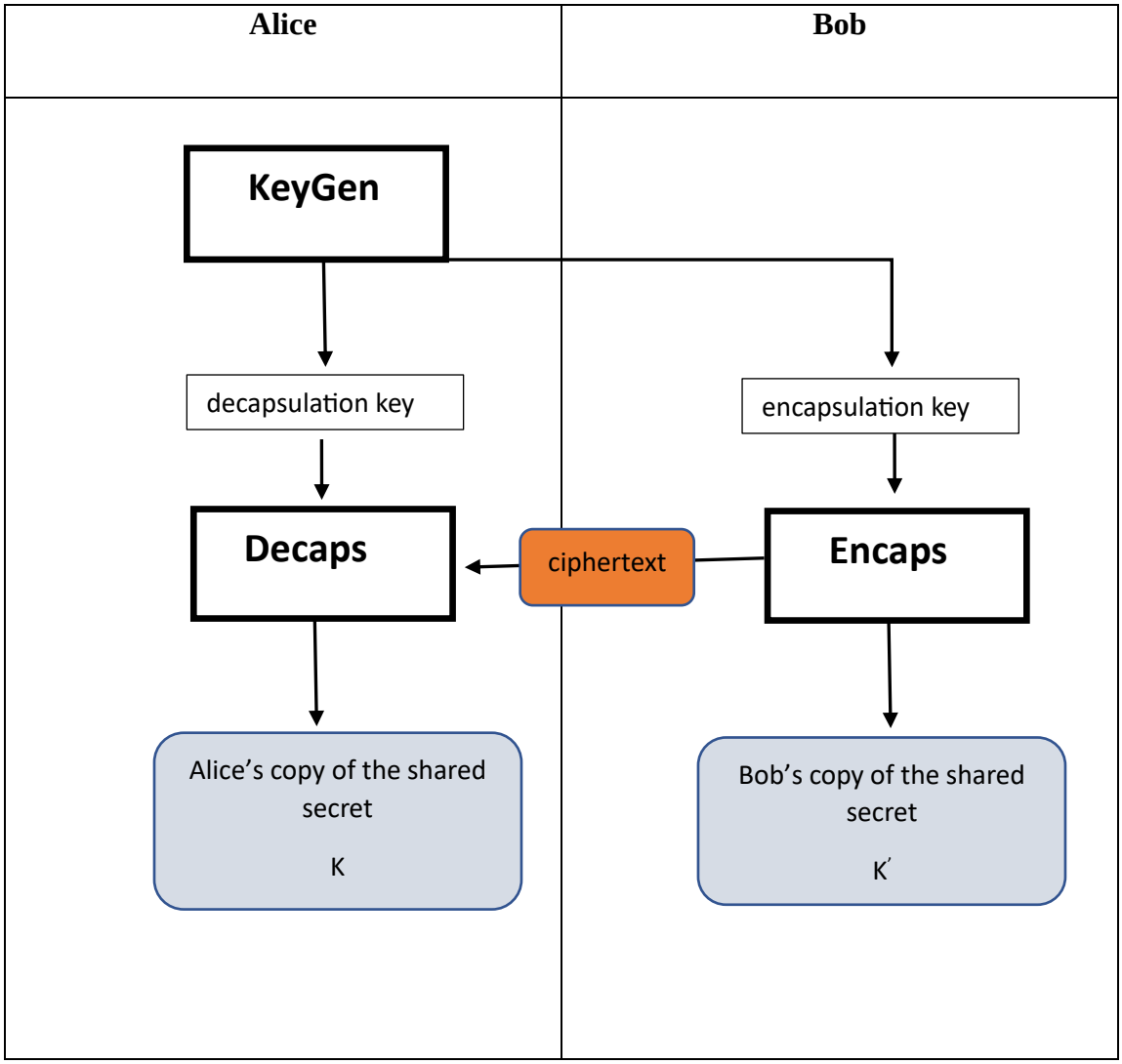
$$b = A \cdot s + e$$

b Değeri şifreli kapsül (ciphertext) olarak adlandırılır. 3. aşamada, Bob, bu kapsülü Alice'e gönderir ve kendi tarafında, bu işlemden türettiği paylaşılan gizli anahtarı K olarak saklar. 4. aşamada, Alice, elindeki s' gizli anahtarını kullanarak gelen b değeri üzerinden benzer bir doğrusal cebir işlemi uygular:

$$v = b \cdot s' = (A \cdot s + e) \cdot s' = A \cdot s \cdot s' + e \cdot s'$$

Bu işlem sonucunda, her iki tarafın hesapladığı sırlar K ve K' belirlenmiştir ancak küçük farklar içerebilir. 5. ve son aşamada, Kyber algoritmasında yer alan özel bir "reconciliation" (uzlaştırma) işlemi mekanizması devreye girer. Bu yöntem sayesinde taraflar, bu küçük farkları tolere ederek aynı ortak sırda uzlaşmış ve sonuç olarak

$K = K'$ İle uzlaşılmış, anahtar paylaşımı başarıyla tamamlanmıştır. (Şekil 2.7)



Şekil 2.7 Anahtar değişim mekanizması (KEM)

CRYSTALS-Dilithium: CRYSTALS-Dilithium, modül kafesleri üzerindeki kafes problemlerinin zorluğuna dayanan seçilmiş mesaj saldırıları (chosen message attack) altında güçlü bir şekilde güvenli olduğu bilinen bir dijital imza şemasıdır. Dilithium, NIST kuantum sonrası kriptografi projesine sunulan aday algoritmalardan biri olmuştur. NIST tarafından 2024 yılında yayımlanan FIPS 204 Module-Lattice-Based Digital Signature Standard kapsamında standartlaştırılmıştır. Bu algoritma, FIPS 204 içinde resmi olarak ML-DSA (Module-Lattice-Based Digital Signature Algorithm) adıyla tanımlanmıştır. Dilithium, güvenliğini Module-LWE ve Module-SIS problemlerine dayandırır. Lattice tabanlı kriptografik yapılar arasında öne çıkan bu imza algoritması, halka ortamında tanımlı polinomlar üzerinde işlem yapar ve yüksek verimlilikle uygulamalar geliştirilmesine imkân tanır. (Ducas et al., 2018).

Çizelge 2.1 ML-DSA anahtar/imza boyutları ve güvenliği'nde görüldüğü üzere: ML-DSA, ECDSA ve RSA gibi klasik dijital imza algoritmalarına kıyasla daha büyük imza ve anahtar boyutlarına sahiptir. Farklı güvenlik seviyelerine uygun şekilde üç farklı parametre seti tasarlanmıştır: ML-DSA-44 (Dilithium2), ML-DSA-65 (Dilithium3) ve ML-DSA-87 (Dilithium5). Bu setler sırasıyla yaklaşık 128-bit, 192-bit ve 256-bit güvenlik seviyeleri sunmak üzere tasarlanmıştır.

Çizelge 2.1 ML-DSA anahtar/imza boyutları ve güvenliği

Parametre Seti	Hedef Güvenlik	Açık Anahtar Boyutu	Özel Anahtar Boyutu	İmza Boyutu
ML-DSA-44	≈128 bit	1312 byte	2560 byte	2420 byte
ML-DSA-65	≈192 bit	1952 byte	4032 byte	3309 byte
ML-DSA-87	≈256 bit	2592 byte	4896 byte	4627 byte

CRYSTALS-Dilithium / ML-DSA imza oluşturma ve doğrulama işlemlerine dair matematiksel temeller ise anahtar üretimi, imzalama ve imza doğrulamada aşağıdaki şekillerde kullanılır:

Anahtar Üretimi (Key Generation): Sistemde rastgele bir matris $A\epsilon R_q^{kxl}$ ve gizli vektörler $s_1\epsilon R_q^l$ ve $s_2\epsilon R_q^k$ seçilir. Açık anahtar $t := A \cdot s_1 + s_2$ hesaplanarak oluşturulur. Açık anahtar çifti (A, t), gizli anahtar ise (A, s_1 , s_2) olarak tanımlanır:

$A\epsilon R_q^{kxl}$ Üzerinde Rastgele bir matris ve $s_1\epsilon R_q^l$ ve $s_2\epsilon R_q^k$ olmak üzere:

$$(s_1, s_2)\epsilon S_\eta^l \times S\eta_\eta^k$$

$$t := A \cdot s_1 + s_2$$

$$t := t_1 + 2^d t_0$$

$$(pk, sk) := \big((A, t_1), (s_1, s_2, t_0)\big)$$

İmzalama (Sign): Mesaj μ için rastgele bir $y\epsilon R_q^l$ vektörü seçilir ve $\omega = A \cdot y$ hesaplanır. ω 'nin üst bitleri ω_1 = HighBits (ω) alınarak $c = H(\omega_1, \mu)$ challenge değeri elde edilir. İmza bileşenleri $z = y + c \cdot s_1$ ve $h = MakeHint(-c \cdot t, w - c \cdot s_2 + c \cdot t)$ hesaplanarak $\sigma = (z, c, h)$ İmzası oluşturulur: (Çizelge 2.2 İmza oluşturma girdi ve çıktı parametreleri)

$$y\epsilon R_q^l$$

$$\omega = A \cdot y$$

$$c = H(\omega_1, \mu)\ (\omega_1 = \omega\text{'nin üst bitleri})$$

$$z = y + c \cdot$$

Çizelge 2.2 İmza oluşturma girdi ve çıktı parametreleri

Input (Girdi)	Output (Çıktı)
Message μ $sk = (s_1, s_2, t_0)$	Signature **σ = (z, c, h)**

İmza Doğrulama (Signature Validation): Doğrulama sürecinde
$w'_1 = UseHint(h, A \cdot z - c \cdot t)$ Hesaplanır ve $c' = H(\mu \parallel w'_1)$ değeri elde edilir. İmzanın geçerliliği $c \stackrel{?}{=} c'$ eşitliği kontrol edilerek belirlenir:

$$\omega' = A \cdot z - c \cdot t_1$$
$$c' = H(\omega_1', \mu).$$
$$c \stackrel{?}{=} c'$$

Falcon: Falcon algoritması, post-kuantum dijital imza standartları arasında öne çıkan bir diğer algoritmadır. NIST tarafından FIPS 206 standardı kapsamında yayımlanması planlanmakta olup, standart ile FN-DSA (FFT over NTRU-Lattice-Based Digital Signature Algorithm) olarak adlandırılacaktır. Falcon'un en önemli avantajlarından biri, benzer güvenlik seviyesindeki diğer algoritmalara kıyasla son derece küçük imza boyutları sunmasıdır (Çizelge 2.3 Falcon anahtar/imza boyutları ve güvenliği). Örneğin, Falcon-512 yaklaşık 666 bayt uzunluğunda imza üretmektedir (Ducas et al., 2019). Bu özellik, bant genişliğinin sınırlı olduğu ağ protokolleri veya belge boyutunun kritik olduğu sistemlerde Falcon'u ön plana çıkarmaktadır. Diğer yandan Falcon'un imzalama süreci, yüksek hassasiyet gerektiren Gaussian örnekleme yöntemine dayanır ve bu yöntem kayan noktalı (floating point) aritmetik içerir. Bu hesaplamalar, sayısal kararlılığın önemli olduğu ortamlarda dikkatle ele alınmalıdır. Özellikle FP (floating point) birimi bulunmayan donanımlar (ör. mikrodenetleyiciler, bazı HSM'ler) üzerinde Falcon implementasyonu zorluk yaratabilir. Ayrıca, hassasiyet eksikliği veya yuvarlama hataları gibi nedenlerle zayıf şekilde gerçekleştirilen örnekleme süreçleri, potansiyel yan kanal saldırılarına açık hâle gelebilir. (NIST PQC Round 3 Report, 2022).

Falcon'un güvenliği, NTRU-tabanlı kafeslerde tanımlı olan SIS problemine dayanmaktadır. Algoritma, trapdoor sampling (tuzak kapı örnekleme) tekniği kullanarak özel anahtarı bilen tarafın kısa çözüm vektörlerini yüksek doğrulukla seçmesini sağlar. Falcon'un temel yapısı, Gentry–Peikert–Vaikuntanathan (GPV) imza çerçevesine dayanır ve Fourier transform temelli halkasal kafeslerde optimize edilmiştir. Bu yapı sayesinde hem 128-bit düzeyinde güvenlik hem de verimli imza boyutları sağlanmaktadır.

Çizelge 2.3 Falcon anahtar/imza boyutları ve güvenliği

Parametre Seti	Hedef Güvenlik	Açık Anahtar Boyutu (byte)	Özel Anahtar Boyutu(byte)	İmza Boyutu(byte)
Falcon-512	128 bit	897	1281	666
Falcon-1024	256	1793	2305	1280

Falcon imza oluşturma ve doğrulama işlemlerine dair matematiksel temeller ise anahtar üretimi, imzalama ve imza doğrulamada aşağıda özetlenmiştir:

Falcon algoritmasında kullanılan sistem parametreleri belirli sabitler üzerinden tanımlanmıştır. Modulus değeri $q = 12289$ olup, hem Falcon-512 hem de Falcon-1024 parametre setlerinde aynıdır. Yani farklı güvenlik seviyelerine karşılık gelen varyantlarda polinom derecesi (n) değişirken, modulus değeri sabit kalır. Falcon-512'de $n = 512$ kullanılırken, Falcon-1024'te $n = 1024$ değeri tercih edilmektedir. Ayrıca, Gaussian örnekleme için kullanılan standart sapma (σ) ve norm sınırları (β) parametre setine göre farklılaşır. Ancak modulus değerinin sabit tutulması hem güvenlik kanıtlarının tutarlılığı hem de algoritmanın verimliliği açısından kritik bir tercih olarak tanımlanmıştır.

Anahtar Üretimi (Key Generation): Falcon algoritmasında aşağıdaki adımlar ile anahtar üretimi yapılır:

1- Polinomların Seçilmesi: Falcon'un temelinde NTRU kafesleri vardır. İlk adımda küçük katsayılı iki polinom seçilir:

$$f, g \in Z_n[x]/(x^n + 1)$$

Burada n parametre setine göre 512 veya 1024'tür.

f ve g, küçük tamsayı katsayılara sahip olacak şekilde rastgele seçilir.

2- NTRU Denklemi*:* Açık anahtar NTRU eşitliği kullanılarak türetilir:

$$h = g \cdot f^{-1} \, (mod\ q)$$

Burada f^{-1}, f polinomunun q modulo tersidir. Bu işlem f'in terslenebilir olması şartı ile yapılır.

3- Trapdoor Matrisi: Gizli anahtarın güvenliğini ve imzalama sürecinde kısa vektör örneklemeyi mümkün kılmak için bir trapdoor yapısı tanımlanır. Bu, dört polinomdan oluşan bir yapı ile sağlanır:

$$B = \begin{matrix} g & -f \\ G & -F \end{matrix}$$

Buradaki f ve g, NTRU denkleminin genişletilmiş çözümüyle elde edilen yardımcı polinomlardır. Bu matris Fourier (FFT) domain'ine aktarılır ve Gram-Schmidt ortogonalizasyonu uygulanarak örnekleme için kullanılacak biçime getirilir.

4- Anahtar Çifti:

Açık Anahtar (pk): h ve
Özel Anahtar (sk): (f, g, B_{FTT})- f ve g polinomları ile Fourier domain'inde saklanan trapdoor bilgisi olarak elde edilmiştir.

İmza Oluşturma (Signature Generation): Mesaj μ in özeti alınır:

$$C = H(\mu)$$

Aşağıdaki eşitliği sağlayan bir kısa vektör bulunur (SIS Problem)

$$h \cdot s_1 + s_2 \equiv c \,(mod\ q)(f, g)$$

GPV çerçevesinde özet bir trapdoor sampling yöntemi ile kullanılır. Bu işlem Fourier domainine çevrilen $fveg$ nin yardımı ile Gaussian örnekleme yapar. Çözüm vektörü (s_1, s_2) hem doğrusal ilişkiyi sağlar hem de kısa normludur.

$$\text{İ}mza\ \sigma\ =\ (s_1, s_2)$$

Her iki vektör de norm kontrolünden geçmelidir: $||s_1||, ||s_2|| \leq \beta\ (\beta: norm\ sınırı)$

Çizelge 2.1 Falcon anahtar üretimi, imza oluşturma ve doğrulama

İmza Doğrulama (Signature Verification): Öncelikle mesajın hash'i yeniden hesaplanır:

$$c' = H(\mu)$$

Aşağıdaki eşitlik kontrol edilir:

$$h \cdot s_1 + s_2 \equiv c' \, (mod \, q)$$

İmza vektörlerinin norm sınırları kontrol edilir:

$$||s_1|| \leq \beta_1 \, ve \, ||s_2|| \leq \beta_2$$

Burada β_1, β_2: Önceden tanımlanmış norm sınırlarıdır. Doğrulama sonucuna göre imzanın geçerliliğine karar verilir. Hem eşitlik hem de norm kontrolleri başarılıysa imza geçerli, değilse geçersizdir.

2.4.2 Hash tabanlı Kriptografi (Hash-Based Cryptography)

Hash tabanlı kriptografi post-kuantum kriptografi alanlarından biridir. Bu yöntemlerde dijital imzalar yalnızca kriptografik hash fonksiyonları kullanılarak oluşturulur. Güvenlik, hash fonksiyonlarının çakışma dirençliliği (collision resistance), ön görüntü dirençliliği (preimage resistance) ve ikinci ön görüntü dirençliliği (second preimage resistance) gibi temel kriptografik özelliklerine dayanmaktadır. Grover algoritması, hash fonksiyonlarının güvenlik seviyesini yarıya indirse de yeterince büyük hash çıktı boyutları kullanılarak kuantum dirençli güvenlik sağlanabilmektedir.

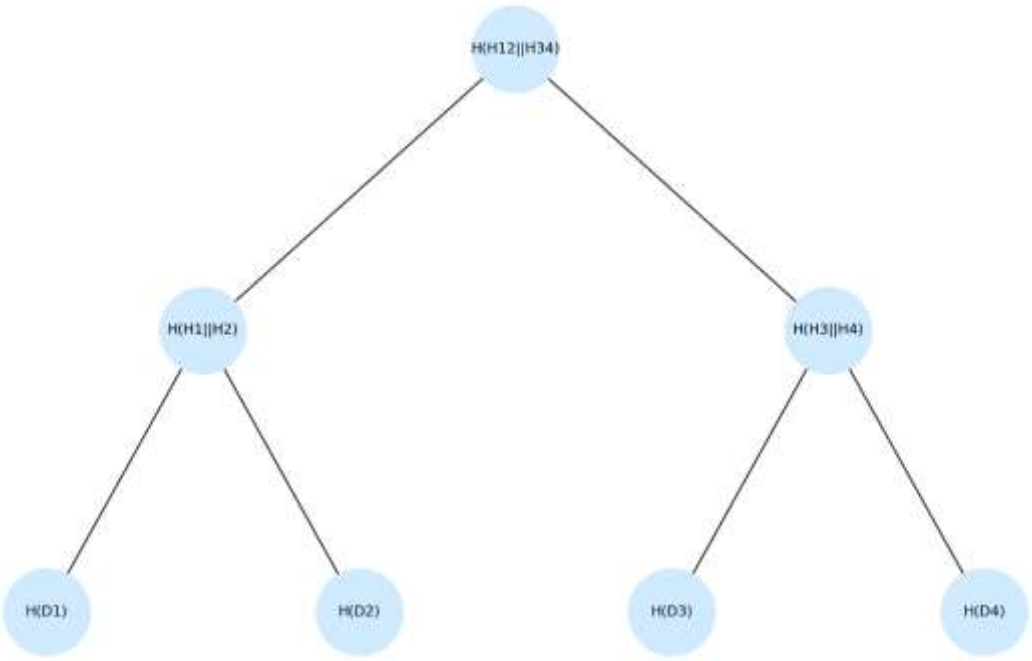
XMSS: Hash tabanlı imza algoritmaları arasında en öne çıkan yapılar Merkle ağaçları tabanlı olanlardır. İlk örneklerden Merkle Signature Scheme (MSS), yalnızca bir mesaj için güvenli imza üretebilirken, zamanla geliştirilen yöntemlerle bu kısıtlama aşılmıştır. XMSS, hash tabanlı ve kuantum dirençli bir dijital imza algoritması olup, RFC 8391 kapsamında IETF tarafından standardize edilmiştir (Hülsing et al., 2018). Ayrıca BSI tarafından da TR-03116-4 teknik raporunda önerilen algoritmalardan biridir (BSI, 2018). XMSS, durum izlemeli (stateful) bir algoritmadır; yani her imza için kullanılan özel anahtar bileşeni yalnızca bir kez kullanılabilir. Bu durum, sistemin imza durumunu sürekli takip etmesini gerektirir ve uygulama tarafında dikkatli anahtar yönetimini zorunlu kılar.

XMSS, temel yapıtaşı olarak Winternitz One-Time Signature Plus (WOTS+) algoritmasını kullanır ve bu tek kullanımlık imzaları, çoklu imza üretimine olanak tanıyan

bir Merkle ağaç yapısı ile düzenler. Merkle ağacı, her yaprak düğümünün veri bloklarının hash değerini, iç düğümlerin ise alt düğümlerinin hash değerini içerdiği ikili bir ağaç yapısıdır. Bu yapı, tüm verinin bütünlüğünü kök düğüm üzerinden doğrulama imkânı sunar. Merkle ağaç yapısı *Şekil 2.8 Merkle ağaç yapısı*'nda görüldüğü üzere aşağıdaki şekilde özet değerler hesaplanarak yapılandırılır:

$$H1 = H(D_1), H_2 = H(D_2), \ldots, H_4 = H(D_4)$$
$$\text{İç } d\text{üğü}mler: H_{1,2} = H(H_1 \parallel H_2), H_{3,4} = H(H_3 \parallel H_4)$$
$$Kök = H\left(H_{1,2} \parallel H_{3,4}\right)$$

Yaprak düğümlerinden herhangi birinde yapılan değişiklik, kök hash değerine kadar yansır. Bu sayede, veri bütünlüğünün bozulup bozulmadığı kolayca tespit edilebilir.



Şekil 2.8 Merkle ağaç yapısı

WOTS+ Algoritması: WOTS+, hash tabanlı bir tek kullanımlık dijital imza algoritmasıdır. Her bir özel anahtar öğesi, mesajın hash değerine göre belirlenen sayıda hash işlemi uygulanarak imza oluşturulur. Bu zincir uzunluğu, Winternitz parametresi (ω) tarafından belirlenir. İmza oluşturma işlemi, özel anahtarın her bir öğesinin belirli sayıda

hash edilmesiyle gerçekleştirilir. Doğrulama sürecinde, imza kullanılarak WOTS+ genel anahtarı yeniden oluşturulur ve bilinen değer ile karşılaştırılır. Her anahtar çifti yalnızca bir kez kullanılmalıdır; aksi takdirde sistemin güvenliği tehlikeye girecektir.

İmza Oluşturma (Key Generation): XMSS'de açık anahtar, Merkle ağacının kök düğümünden oluşur; özel anahtar ise her biri yalnızca bir kez kullanılabilen WOTS+ anahtar çiftlerinden oluşur. İmzalanacak mesaj, öncelikle kriptografik bir özet fonksiyonundan geçirilerek sabit uzunlukta bir özet değeri elde edilir. Bu özet, kullanılmamış bir WOTS+ özel anahtarı ile imzalanır. Ardından, imzalanan yaprağın Merkle ağacındaki yerini doğrulamak için gerekli kimlik doğrulama yolu (authentication path) hesaplanır ve imzanın bir parçası olarak WOTS+ imzasına eklenir.

İmza Doğrulama: Doğrulayıcı taraf, imzadan WOTS+ genel anahtarını yeniden üretir ve verilen mesaj özeti ile uyumlu olup olmadığını kontrol eder. Sonrasında, eklenmiş kimlik doğrulama yolunu kullanarak Merkle ağacının kök düğümüne ulaşır. Hesaplanan kök düğüm, gönderilen açık anahtardaki kökle karşılaştırılır. Eğer eşleşme sağlanırsa, imza geçerli kabul edilir.

NIST, XMSS algoritmasını hash tabanlı ve kuantum dirençli bir imza mekanizması olarak kabul etmekle birlikte, NIST SP 800-208 (Recommendation for Stateful Hash-Based Signature Schemes) tavsiyelerinde bu algoritmanın yalnızca belirli uygulama senaryoları için uygun olduğunu vurgulamaktadır. Özellikle, XMSS'in durum izlemeli (stateful) yapısı, çoklu işlemci veya dağıtık sistem ortamlarında kullanılmasını zorlaştırmaktadır. Bu nedenle NIST, bu algoritmaların entegre edilmesinde dikkatli olunması ve özel anahtarın yeniden kullanım risklerinin önlenmesi için uygun protokol düzeyinde önlemler alınmasını önermektedir. Bu hali ile TLS protokolü için de her imza için durum takibi gerektirmesi ve anahtarın yeniden kullanılmasının önlenmesi gerekir. Bu durum çoklu işlem ve sunucu ortamlarında oldukça karmaşık hale gelecektir. Ayrıca imza boyutunun büyük olduğu da düşünüldüğünde TLS için uygun bir aday olarak görülmemektedir.

SPHINCS+: SPHINCS+, NIST tarafından 2024 yılında yayımlanan Federal Information Processing Standards Publication 205 (FIPS 205): Stateless Hash-Based Signature

Standard (SPHINCS+) kapsamında resmi olarak standartlaştırılmış durumsuz bir hash tabanlı dijital imza algoritmasıdır. Bu standartta algoritma, SPHINCS+ adıyla tanımlanmış ve kriptografik dayanıklılığı hash fonksiyonlarının güvenliğine dayalı olarak açıklanmıştır. XMSS'in durum takibi gerektiren yapısına alternatif olarak geliştirilmiştir ve imza üretimi için herhangi bir dışsal durum bilgisi tutma ihtiyacını ortadan kaldırır. SPHINCS+ imza yapısı XMSS'den önemli ölçüde farklıdır. XMSS yalnızca WOTS+ ve Merkle ağacı yapısıyla imza üretirken, SPHINCS+ ek olarak FORS (Forest of Random Subsets) adı verilen sınırlı sayıda kullanımlık imza şeması içerir. FORS, mesaj hash'ini küçük parçalara bölerek her parça için ayrı imza oluşturur. SPHINCS+ ayrıca tek bir Merkle ağacı yerine çok katmanlı (hyper-tree) Merkle yapısı kullanır.

SPHINCS+ algoritması, farklı güvenlik seviyeleri ve performans gereksinimleri için çeşitli parametre setlerine sahiptir. FIPS 205 standardında tanımlanan ana parametre setleri şunlardır:

- SPHINCS+-128s/f/simple: 128-bit güvenlik seviyesi sağlar. "s" (small) versiyonu daha küçük imza boyutu, "f" (fast) versiyonu daha hızlı imza üretimi için optimize edildiği işaretini taşır.
- SPHINCS+-192s/f/simple: 192-bit güvenlik seviyesi için tasarlanmıştır.
- SPHINCS+-256s/f/simple: En yüksek güvenlik seviyesi olan 256-bit için kullanılır.

Bu parametre setlerindeki "s" ve "f" ayrımı, imza boyutu ile performans arasındaki dengeyi temsil eder. "Small" versiyonları daha az ağaç yüksekliği kullanarak küçük imzalar üretirken, "fast" versiyonları daha fazla paralelleştirme imkânı sunarak hızlı imza üretimi sağlar. "Simple" versiyonları ise en basit hash fonksiyonu kullanımını tercih ederek uygulama kolaylığını önceliklendirir. Bu esneklik, SPHINCS+'ın farklı uygulama senaryolarına uyarlanabilmesini sağlamaktadır.

2.4.3 Kod tabanlı Kriptografi (Code-Based Cryptography)

Kod tabanlı Kriptografi, 1978'de McEliece tarafından önerilen bir şifreleme sistemine dayanır ve kriptografik güvenliği, hata düzeltici kodların zorluk seviyesinden faydalanarak sağlar. Bu yaklaşımın temelinde, rastgele seçilmiş ve gizli yapısı bilinmeyen bir kodu çözmenin zorluğu yatar. Mevcutta, kod tabanlı sistemlerin kuantum bilgisayarlara karşı dayanıklı olduğu düşünülmektedir. Bunun en önemli nedeni, Shor ve Grover algoritmalarının bu tür kod tabanlı problemleri çözmede üstel hızlanma sağlayamamasıdır. Kod tabanlı kriptografinin güvenliği, NP-zor bir problem olan syndrome decoding'e dayanır. Bu problem, belirli bir sindrom ve hata ağırlığı verildiğinde, belirtilen ağırlıkta bir hata vektörü bulmayı gerektirir ki bu da hesaplama açısından oldukça zordur.

McEliece Şifreleme Sistemi: McEliece şifreleme sistemi, Goppa kodları olarak bilinen özel bir hata düzeltici kod sınıfını kullanır. Sistem, çok büyük açık anahtar boyutlarına sahip olmasına rağmen hem şifreleme hem de şifre çözme işlemleri oldukça hızlıdır. Anahtar üretiminde, kullanıcı, gizli bir Goppa kodu seçer ve buna karşılık gelen genel jeneratör matrisinden açık anahtar üretir. Gönderici, mesajı jeneratör matrisi ile çarparak kod kelimesine dönüştürür ve buna rastgele bir hata vektörü ekler. Alıcı, gizli kod bilgisi sayesinde hatayı düzelterek orijinal mesajı geri elde eder ve şifreyi çözmüş olur. McEliece sistemi, 40 yılı aşkın süredir kriptanalize karşı direnç göstererek ciddi bir kırılma vakası yaşamamıştır. Bu özelliği, onu büyük açık anahtar boyutu dezavantajına rağmen, post-kuantum kriptografi alanında güvenilir bir aday haline getirmektedir (Bernstein et al., 2008).

Classic McEliece: Kod tabanlı kriptografi ailesine mensup olan Classic McEliece, bir anahtar kapsülleme mekanizması (Key Encapsulation Mechanism – KEM) olarak, NIST'in post-kuantum kriptografi (PQC) standardizasyon sürecinde değerlendirilen algoritmalardan biridir. NIST, 4. Tur değerlendirmesinde algoritmayı finalist olarak ilan etmiş, ancak 2025 yılı itibarıyla henüz resmi olarak standartlaştırmamıştır. Teknik açıdan değerlendirildiğinde, Classic McEliece algoritması özellikle çok büyük genel anahtar boyutlarına sahiptir. Önerilen parametre setlerine göre bu boyutlar yaklaşık 261 KB ile 1.3 MB arasında değişmektedir (Bernstein et al., 2019). Bu durum, özellikle bant genişliği

kısıtlı ağ ortamlarında ve sınırlı bellek kapasitesine sahip gömülü sistemlerde algoritmanın entegrasyonunu zorlaştırmaktadır. Ek olarak, Classic McEliece yaklaşık kırk yılı aşkın süredir akademik literatürde yer almakta ve çok çeşitli saldırı senaryolarına karşı analiz edilmiş bir yapıya sahiptir. Özellikle kuantum bilgisayarların tehdidinde hem klasik hem kuantum saldırılara karşı gösterdiği direnç, algoritmayı hâlâ önemli bir post-kuantum alternatif hâline getirmektedir.

BIKE: BIKE (Bit Flipping Key Encapsulation) algoritması, NIST tarafından yürütülen post-kuantum kriptografi (PQC) standardizasyon sürecinin 4. Turunda değerlendirilen algoritmalardan biridir. Özellikle QC-MDPC (Quasi-Cyclic Moderate Density Parity Check) kodlarına dayanan yapısıyla dikkat çeken BIKE, donanım avantajı ve görece daha küçük anahtar boyutlarına sahip bir çözüm sunmaktadır. Ancak NIST, Mart 2025 itibarıyla 4. Tur değerlendirmesini tamamlamıştır. Sonuç olarak BIKE, daha az olgun güvenlik analizi ve dekodlama hatası oranı gibi nedenlerle NIST tarafından tercih edilmemiştir. Bu karar, NIST'in resmî açıklamaları ve PQC Konferansları'nda belirtilen güvenlik analizlerinin karşılaştırması sonucunda alınmıştır.

2.4.4 Çok değişkenli polinomlara dayalı Kriptografi (Multivariate Polynomial Cryptography)

Çok değişkenli polinomlara dayalı kriptografi, post-kuantum kriptografi alanında dijital imza sistemleri için bir diğer seçenektir. Bu yaklaşım, sonlu alanlar üzerinde tanımlı çok değişkenli ikinci dereceden (quadratic) polinom denklemler sistemlerinin çözümünün hesaplama açısından zor olmasına dayanır. Matematiksel olarak bu problem, MQ Problem (Multivariate Quadratic Problem) olarak bilinir ve NP-tam sınıfında yer alır (Garey & Johnson, 1979). Bu yapı, kuantum bilgisayarların çözmekte zorlandığı bir problemi temel aldığı için güçlü bir güvenlik temeli sunmaktadır. Bu alandaki algoritmaların en büyük avantajı, imzalama ve doğrulama işlemlerinin oldukça hızlı olmasıdır. Ancak çoğu zaman anahtar boyutlarının oldukça büyük olması, özellikle düşük kaynaklı cihazlarda kullanımını sınırlar.

Rainbow: Rainbow, çok değişkenli imza sistemleri içerisinde en bilinen ve yaygın algoritmalardan biridir. NIST'in post-kuantum kriptografi standardizasyon sürecinde dijital imza kategorisinde finalist olarak belirlenmiş ancak son turda Beullens tarafından geliştirilen verimli anahtar kurtarma saldırısının ortaya çıkması nedeniyle 2022 yılında yarışmadan elenmiştir (Beullens, 2022).

GeMSS: GeMSS (Great Multivariate Short Signature), kısa imza boyutları ve hızlı doğrulama işlemleri sunan bir diğer çok değişkenli algoritmadır. NIST sürecinde uzun süre önde gelen adaylardan biri olmuş, ancak karmaşık yapısı ve uygulamadaki potansiyel zafiyetler nedeniyle finale kalamamıştır (Courtois et al., 2020).

Güncel Gelişmeler NIST'in post-kuantum standardizasyon sürecinde multivariate algoritmalar genellikle elense de araştırma topluluğu bu alanda çalışmaya devam etmektedir. MAYO, PROV ve diğer yeni multivariate imza şemaları geliştirilmekte ve analiz edilmektedir. Bu algoritmalar, özellikle IoT cihazları gibi hesaplama gücü sınırlı ortamlarda hızlı imzalama gereksinimleri için potansiyel çözümler sunmaktadır. Multivariate kriptografinin temel avantajı olan yüksek performans, gelecekteki özel kullanım alanları için değerli olmaya devam etmektedir.

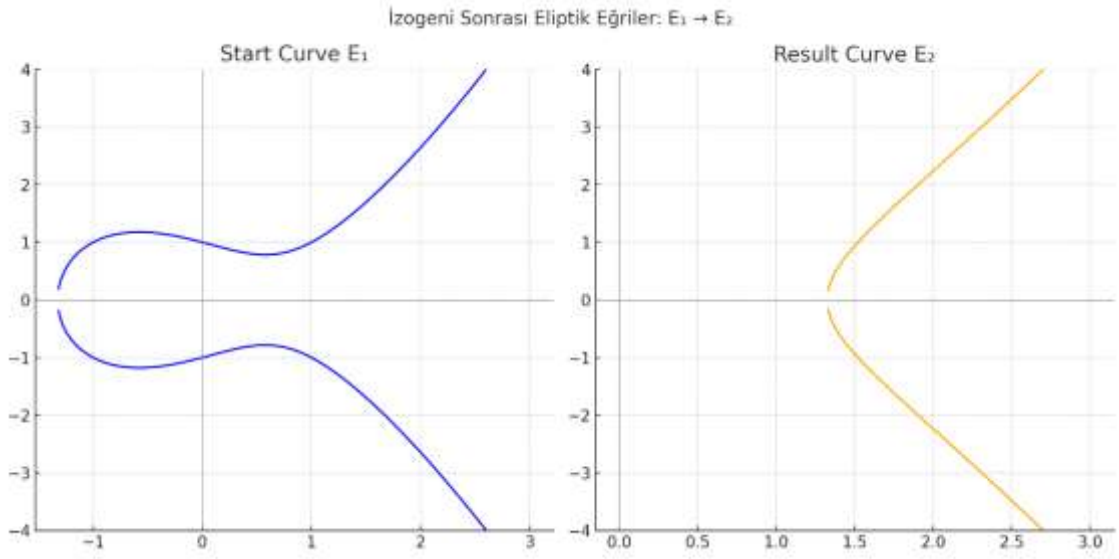
2.4.5 İzogeni tabanlı Kriptografi (Isogeny-Based Cryptography)

İzogeni tabanlı kriptografi, post-kuantum döneme uygun kriptografik algoritmalar arasında, özellikle anahtar boyutunun küçüklüğü açısından dikkat çeken bir yaklaşımdır. Bu yaklaşım, eliptik eğriler arasındaki izogenilerin hesaplanmasının zorluğuna dayanmaktadır. Klasik eliptik eğri kriptografisinde (ECC) kullanılan yapılar ile benzerlik göstermesine rağmen, izogeni tabanlı sistemler farklı matematiksel varsayımlar üzerine kuruludur ve kuantum bilgisayarlara karşı daha dirençlidir.

İzogeni tabanlı sistemler, iki eliptik eğri arasındaki izogeni bulmanın hesaplama açısından zorluğunu kullanır. Eliptik eğriler üzerinde tanımlanan izogeniler, eliptik eğriler arasında özel bir homomorfizma türüdür ve belirli bir dereceye sahip

fonksiyonlardır. Bu fonksiyonlar, iki eğri arasındaki yapısal ilişkiyi tanımlar ve simetrik olmayan (asimetrik) bir problem sunar: Bir izogeni ve sonuç eğri biliniyorsa, başlangıç eğrisini tahmin etmek zor olabilir. Bu asimetri, kriptografik olarak avantaj sağlar.

Şekil 2.9 Elliptik Eğri Isojeni sonrası temsili gösterimi'nde, izomorfizm tabanlı sistemlerin dayandığı temel matematiksel yapı olan eliptik eğriler üzerindeki izogeni ilişkisi görselleştirilmiştir. Şemanın sol tarafında yer alan E_1 eğrisi, bir eliptik eğriyi; sağ tarafta yer alan E_2 eğrisi ise bu eğriye uygulanan belirli dereceye sahip bir izogeni (ϕ) sonucu elde edilen yeni eğriyi temsil eder. Her iki eğri de $y^2 = x^3 - ax + by^2$ formundadır ve izogeni, bu eğrilerin parametrelerini dönüştürerek yeni bir eğri elde edilmesini sağlar.



Şekil 2.9 Elliptik Eğri Isojeni sonrası temsili gösterimi

İzogeniler, eğriler arasında homomorfik bir ilişki kurar; yani gruplar arası yapıyı koruyarak bir dönüşüm gerçekleştirir. Ancak bu dönüşüm asimetriktir:

$E_1 \rightarrow E_2$dönüşümü kolayca hesaplanabilirken, ters yönde $E_2 \rightarrow E_1$dönüşümünü elde etmek (özellikle izogeniyi belirlemek) hesaplama açısından zordur. Bu asimetri, izogenitabanlı kriptografik protokollerin temel güvenlik varsayımını oluşturur.

Bu yaklaşım ilk olarak Couveignes (1997) ve Rostovtsev–Stolbunov (2006) tarafından teorik olarak önerilmiş, daha sonra Jao ve De Feo (2011) tarafından SIDH (Supersingular Isogeny Diffie-Hellman) algoritması ile pratiğe dökülmüştür.

SIDH ve SIKE: SIDH (Supersingular Isogeny Diffie-Hellman) ve bu algoritmanın daha güvenli versiyonu olan SIKE (Supersingular Isogeny Key Encapsulation), izomorfizm tabanlı kriptografi ailesinin en bilinen algoritmalarıdır. SIKE, NIST PQC standardizasyon sürecinin üçüncü turuna kadar ilerlemiş ve en küçük anahtar boyutlarına sahip algoritmalardan biri olarak dikkat çekmiştir. Ancak Temmuz 2022'de SIDH algoritması ciddi bir kriptanalitik saldırıya uğramış ve SIKE'nin temelini oluşturan yapının güvenliği zedelenmiştir. Castryck ve Decru tarafından önerilen bu saldırı, supersingular eliptik eğri izogenilerini kullanarak SIKE algoritmasını etkili biçimde kırmıştır. Bu gelişme sonrasında NIST, SIKE algoritmasını yarışmadan çıkardığını duyurmuştur (NIST, 2022).

İzogeni tabanlı kriptografi, anahtar boyutunun küçüklüğü açısından önemli bir avantaj sunsa da SIKE gibi önde gelen algoritmaların güvenlik açıkları nedeniyle bu alandaki beklentiler önemli ölçüde düşmüştür. Bununla birlikte, matematiksel olarak hâlen aktif bir araştırma alanı olmaya devam etmekte, alternatif yapıların güvenlik açısından incelenmesine yönelik çalışmalar sürmektedir.

2.4.6 Diğer PQC algoritma sınıfları

Post-kuantum kriptografi alanında, ana kategoriler dışında kalan ve nispeten daha özgün matematiksel yapılara dayanan bazı algoritma sınıfları da mevcuttur. Bu sınıflar, genellikle daha az bilinen ya da daha özel uygulamalara yönelik olarak geliştirilmiştir.

NTRU (N-th Degree Truncated Polynomial Ring Units): NTRU, halka tabanlı bir şifreleme algoritmasıdır ve teknik olarak lattice-based kriptografinin bir alt dalı olarak sınıflandırılır. 1996 yılında Hoffstein, Pipher ve Silverman tarafından önerilen bu algoritma, polinomsal halka yapısı içinde gerçekleştirilen çarpımlara dayanır. NTRU, ML-KEM'e kıyasla daha basit bir yapıya sahiptir ve özellikle gömülü sistemlerde düşük hesaplama maliyeti sunmaktadır. Orijinal NTRUEncrypt, NIST PQC sürecinin ilk turunda

değerlendirilmiş ve ilerleyen aşamalara geçememiştir. Ancak NTRU ailesi tamamen süreçten çıkmamış; NTRU Prime varyantları (Streamlined NTRU Prime ve NTRU LPRime) 3. tura kadar ilerlemiş, fakat final standardizasyona dahil edilmemiştir. Algoritmanın sadeliği ve performansı nedeniyle özel uygulamalarda alternatif bir çözüm olarak değerlendirilmeye devam etmektedir.

Kuantum Teknolojileri ve Post-kuantum Kriptografi İlişkisi: Post-kuantum algoritmalarının yanı sıra, kuantum teknolojisini doğrudan kullanan yaklaşımlar da mevcuttur. Kuantum Anahtar Dağıtımı (QKD), kuantum mekaniği prensiplerini doğrudan kullanan ve teorik olarak mükemmel güvenlik sağlayan bir yöntemdir. QKD, post-kuantum kriptografi algoritmalarından farklı bir paradigmayı temsil eder çünkü kuantum teknolojisi kullanır. Pratik kısıtlamaları (mesafe, hız, altyapı maliyeti) nedeniyle yaygın kullanım için henüz uygun değildir, ancak özel güvenlik gereksinimlerinde alternatif çözüm sunmaktadır.

Sonuç olarak, bu alternatif yaklaşımlar, ana post-kuantum kategorilerinin yanında çeşitlilik sunarak, farklı güvenlik modellerine ve uygulama gereksinimlerine yönelik çözüm arayışında önemli rol oynamaktadır. Ancak her yaklaşımın güncel güvenlik durumu ve pratik uygulanabilirliği sürekli olarak araştırılmaya ve değerlendirilmeye devam etmektedir.

2.5 TLS Protokolünde Post-Kuantum Kriptografi Kullanımı

TLS protokolü gibi yaygın kullanılan iletişim protokolünün PQC algoritmalarla uyumlu hale getirilmesi, uzun vadeli veri gizliliği ve sistem güvenliği açısından önemlidir. Ancak Post-kuantum kriptografiye geçiş süreci, yalnızca yeni algoritmaların geliştirilmesiyle sınırlı olamaz. Aynı zamanda bu algoritmaların protokole entegre edilmesi ve bunun pratikte uygulanabilirliği de üzerinde durulması gereken önemli bir noktadır. Post-kuantum algoritmaların pratikte kullanılabilirliğini test etmek ve TLS benzeri mevcut protokollere entegrasyonunu kolaylaştırmak amacıyla çeşitli açık kaynak projeleri geliştirilmiştir. Bu çalışmalardan öne çıkan proje Open Quantum Safe (OQS) projesidir. Open Quantum Safe, post-kuantum kriptografi algoritmalarını deneysel olarak mevcut

kriptografik protokollere entegre etmek için hem kriptografik kütüphaneler (liboqs) hem de OpenSSL ve OpenSSH gibi yaygın kütüphanelerin PQC algoritmaları içeren sürümlerini sağlamaktadır. Bu sayede örneğin TLS protokolünde Kyber veya Dilithium gibi algoritmalarla hibrit el sıkışmalarını deneyimleyebilmek için bir ortam sağlanmaktadır. (OQS, 2023).

Bu tür açık kaynak girişimlerin yanı sıra, NIST de hibrit mekanizmalar konusundaki tutumunu belirten bir rapor yayımlamıştır. NIST, post-kuantum algoritmalara geçiş sürecinde, özellikle kritik altyapılarda doğrudan geçişin mümkün olmayabileceğini, bu nedenle hibrit yaklaşımların uygulanabilir ve güvenli bir geçiş stratejisi sunduğunu belirtmiştir (NIST, 2024). Hibrit yapılar, klasik güvenliğe olan bağımlılığı aşamalı olarak azaltırken, aynı zamanda kuantum sonrası tehditlere karşı önlem alınmasına imkân tanır. Bunun nasıl yapılacağına dair ise teknik öneriler, Douglas Stebila ve çalışma arkadaşlarının IETF çatısı altında sunduğu taslak dokümanlarda belirtilmektedir. Stebila'nın önerdiği hibrit model, TLS 1.3 protokolü içerisine hem klasik ECDHE hem de post-kuantum KEM mekanizmalarını dahil ederek çift yönlü güvenlik sağlamayı amaçlamaktadır (Stebila vd., 2021).

2.5.1 TLS 1.3'te hibrit anahtar değişimi

Güvenliğin kriptografi temelli olarak sağlandığı kritik altyapılarda kullanılan kriptografik algoritmaların değiştirilmesi, birçok paydaşa sahip bir ekosistemin de dönüşüme hazır olmasını gerektirir. Hali hazırda aktif olarak kullanılan TLS ve benzeri protokollerde uygun algoritma seçimi, uygulamaların buna hazır oluşu, donanımsal yeterlilikler ve standartlaşma süreçleri önemli hususlardır. Oldukça karmaşık bağımlılıkları olan sistemlerde bu tür değişikliklerin bir anda yapılması, mevcut istemci ve sunucularla uyumsuzluklara ve kritik sistemlerde iletişim kesintilerine yol açabilir. Bu nedenle post-kuantum algoritmalara geçiş kademeli ve dikkatli bir şekilde planlanmalıdır. Topyekûn bir geçişin söz konusu olamayacağı göz önünde bulundurulduğunda kademeli olarak geçişte; eski sistemlerle yeni sistemlerin bir süre birlikte çalışabilmesini sağlamak (interoperability) bu geçişin sorunsuz gerçekleşmesi açısından bir zorunluluktur.

Kuantum bilgisayarlara karşı dirençli kriptografik algoritmaların geliştirilmesi devam ederken, mevcut istemci ve sunucuların büyük bir kısmı yalnızca klasik algoritmaları desteklemeye devam edecektir. Bu nedenle eksi ve yeni sistem arasında güvenli bağlantıya devam edilmesini sağlayacak hibrit bir yapının kurulması ihtiyacı doğmuştur. TLS 1.3 protokolünde hibrit anahtar değişimi (hybrid key exchange) yöntemi, geçiş sürecinde hem mevcut altyapılarla uyumluluğu hem de kuantum sonrası güvenliği aynı anda sağlamayı hedeflemektedir. Bu yaklaşım, post-quantum algoritmalara geçişte iki temel prensibin aynı anda gerçekleştirmeyi amaçlar: Geriye Dönük Uyumluluk (backward compatibility) ve Kuantum Sonrası Güvenlik (post-quantum security). Geriye Dönük Uyumluluk (Backward compatibility), sistemin mevcut klasik kriptografik altyapılarla uyumlu çalışmaya devam edebilmesini ifade eder; yani, kuantum dirençli algoritmaları desteklemeyen istemci veya sunucularla da bağlantı kurulabilir. Böylece yeni sistemler eski sistemlerle iletişim kurarken iletişim kesintiye uğramaz. Öte yandan, kuantum sonrası güvenlik gelecekte ortaya çıkabilecek kuantum bilgisayar tehditlerine karşı şimdiden önlem alınmasını sağlar. Bu prensipler doğrultusunda hibrit yapıda kullanılan post-kuantum algoritmalar, el sıkışma sırasında kullanılan klasik algoritmalara ilave bir güvenlik katmanı oluşturacaktır. Hibrit yöntemde el sıkışma süreci sırasında üretilen oturum anahtarı, yalnızca klasik algoritmalarla değil, aynı zamanda kuantuma karşı dirençli algoritmalar tarafından da korunmuş olur.

Hibrit yaklaşımla ilgili kuantuma dayanıklı algoritmaların TLS gibi yaygın protokollere entegrasyonu konusunda araştırmalar literatürde yer almaktadır. Stebila ve ekibinin "Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH" başlıklı bir çalışması bulunmaktadır. Bu çalışmada NIST'in ikinci tur post-kuantum algoritma adayları arasından seçilen bazı algoritmalarla, TLS ve SSH protokolleri üzerinde prototip uygulamaları geliştirilmiştir. Bu çalışmalar hibrit anahtar değişim yapılarının pratikte uygulanabilirliği üzerine deneyler yapmakta ve kuantum sonrası güvenliğe geçiş sürecinde hibrit çözümlerin önemli bir ara adım olabileceğine işaret etmektedir (Stebila ve ark., 2019).

Hibrit el sıkışma mekanizmaları üzerinde standartlaşma çalışmaları da mevcuttur. IETF tarafından RFC 9180 (Hybrid Public Key Encryption- HPKE) genel hibrit şifreleme için

yayımlanmışken, TLS 1.3'te hibrit anahtar değişimi için özel taslak belgeler hazırlanmaktadır. Son olarak yayımlanan, Hybrid Key Exchange in TLS 1.3 (Internet-Draft: draft-ietf-tls-hybrid-design, Stebila et al., 2025), hibrit yapının TLS 1.3 protokolüyle nasıl entegre edilebileceğine dair teknik ayrıntıları ortaya koymaktadır. Bu öneriye göre el sıkışması şu şekilde gerçekleştirilir: İstemci, key_share uzantısı içerisinde hem klasik (örneğin ECDHE) hem de post-kuantum (örneğin Kyber) anahtarlarını sunucuya iletir. Sunucu da benzer biçimde her iki algoritmaya karşılık gelen geçici anahtar bileşenlerini istemciye gönderir. Bu süreç sonucunda elde edilen iki ayrı gizli anahtar bileşeni birleştirilerek tek bir ortak gizli bilgi (shared secret) hâline getirilir. Burada istemci ve sunucu birbirlerine hem klasik hem de kuantuma dayanıklı algoritmayı kullanarak anahtar paylaşmıştır. Bu "combined shared secret" yapısı hem mevcut sistemlerle uyumluluğu sürdürmeyi hem de kuantum sonrası dönemde kriptografik dayanıklılığı artırmayı hedeflemektedir. Bu yapı sayesinde, klasik ECDHE algoritmasıyla bugünkü istemcilerle uyumluluk korunurken, kuantuma dayanıklı algoritma da ilave güvenlik sağlar. Eğer bir taraf PQC desteklemiyorsa, klasik algoritma üzerinden bağlantı kurulabilir; ancak her iki taraf da destekliyorsa, hibrit yapı aktif olacaktır.

3. MATERYAL ve YÖNTEM

3.1 Kullanılan Donanım ve Yazılım Araçları

Bu çalışma kapsamında gerçekleştirilen testler için post kuantum algoritmaların TLS 1.3 protokolünde kullanılabilirliğini test etmek amaçlı bir test ortamı oluşturuldu. Tüm denemelerde, Intel(R) Core (TM) i5-11400H 2,70 GHz hızındaki 6 çekirdekli işlemcili ve 16 GB RAM'e sahip Windows işletim sistemli bir bilgisayar kullanıldı. TLS 1.3 protokolü için OpenSSL Sürümü 3.4 ve PQC algoritmaların uygulanması için liboqs Sürümü 0.12.0 derlendi. TLS sunucu ve istemci bileşenleri OpenSSL 3.4 ve PQC algoritmaların anahtar değişimi ve kimlik doğrulamada kullanılmasına izin veren ve OQS projesi kapsamında geliştirilen bir bileşen olan 'oqs-provider' ile yapılandırıldı. Senaryoların amaçlarına uygun şekilde yazılan test scriptleri WSL kullanırak çalıştırıldı. Her algoritma için yaklaşık 1000 tekrarlı test yapılmış ve algoritmaların el sıkışma süreleri ölçülmüştür. Tüm test sonuçları kaydedilerek, log dosyalarının analizi yapıldı.

Bu tezde yapılan testler deneysel araştırma metodolojisine uygun olarak tasarlandı. Bağımsız değişken ve bağımlı değişken ayrımı tüm testlerde geçerli olacak şekilde kurgulandı. Bağımsız değişken, her senaryoda değiştirilen post kuantum dirençli algoritmalar olarak belirlendi. Örnek verecek olursak anahtar değişiminde Kyber512 bağımsız değişken olduğu durumda, el sıkışma süresi (handshake duration) bağımlı değişken olarak tanımlandı ve her seferinde kimlik doğrulama algoritması değiştirilerek ölçüldü. Bu süre, milisaniye cinsinden ölçülmüştür. İstatistiksel analiz için ortalama, maksimum süre ve varyans gibi parametreler Python scriptler ile hesaplandı. Çıkan sonuçlar ayrıca yine Python scriptler ile oluşturulan görsel grafiklerle desteklenerek değerlendirildi. Bu yaklaşım, TLS protokolünde post-kuantum algoritmaların entegrasyonunun bağlantıya olan etkilerini yalın şekilde incelemeye olanak sağladı. Bu tür kontrollü test ortamları, literatürde de farklı şekillerde görülmektedir. Bu testler modern TLS yapılandırmaları üzerinde post-kuantum geçiş sürecine dair anlamlı karşılaştırmalarla sonuçlar üretebilmek açısından önemlidir. (Alkim et al., 2016; Bos et al., 2015; Paquin, Stebila & Tamvada, 2020).

3.2 Test Senaryoları ve Yapılandırmaları

TLS 1.3 protokolünde, istemci ve sunucu arasında güvenli bir iletişim kanalı kurmak amaçlanır. Bunun için kimlik doğrulama, anahtar değişimi ve veri şifrelemesi gibi temel kriptografik işlevler yerine getirilir. Bu işlevler, hâlihazırda asimetrik kriptografi ve simetrik kriptografiyi birlikte kullanır. Bu işlevlerde kullanılan kriptografik algoritmalar da güvenliğin sağlanmasında temel unsurlardır. Örneğin, el sıkışma işleminde kullanılan ECDHE algoritması, geçici bir ortak anahtar üzerinde anlaşmayı mümkün kılar. ECDSA veya RSA gibi imza algoritmaları, sunucu/istemci kimliklerinin doğrulanmasında kullanılır. El sıkışma sürecinden sonra ise, bu süreçte elde edilen ortak anahtar kullanılarak AES gibi simetrik algoritmalar ile veriler şifrelenir. Böylece iletişimde gizlilik sağlanır.

Bu çalışma kapsamında, TLS protokolünün kuantum saldırılarına en açık olan iki temel bileşeni incelenmiştir: anahtar değişimi ve sertifika tabanlı kimlik doğrulama. Bunun temel nedeni, bu işlevlerde kullanılan kriptografik algoritmaların kuantum saldırılarına en açık noktalar olarak görülmesidir. Bir diğer kriptografik işlev olan ve gizliliği sağlayan veri şifreleme işlemi, yukarıda belirtildiği gibi AES gibi simetrik algoritmalarla yapılmaktadır. Bu simetrik algoritmalar, kuantum saldırılara karşı dayanıklı görülmektedir. Bunun nedeni ise kuantum bilgisayarlar üzerinde yalnızca Grover algoritması gibi teorik hızlandırmalarla saldırıya uğrayabilmeleridir. Grover algoritması, simetrik anahtarlarda arama uzayını karekök oranında azaltarak güvenlik seviyesini yarıya indirir ve bu durum simetrik algoritmalarda anahtar boyutu artırıldığında telafi edilebilir. (NIST, 2016; Menezes et al., 1996). Bu nedenle, bu tezde veri şifreleme kısmı testlerin dışında bırakılarak, kuantum sonrası geçişte en kritik rol oynayan anahtar değişimi ve kimlik doğrulamaya odaklanılmıştır. Tüm yapılan testler iki temel başlık altında sınıflandırılmıştır:

1. Post Kuantum Anahtar Kapsülleme Mekanizmaları KEM (Key Encapsulation Mechanisms) Testleri: TLS el sıkışmasında anahtar değişimi için PQC algoritmalarının performansı
2. Post Kuantum X.509 Sertifika Temelli Kimlik Doğrulama (Authentication with Post Quantum Certificates): Dijital imza algoritmalarının TLS sertifikalarında kullanımı

Ayrıca bu sınıflandırmanın bir diğer yararı ise TLS el sıkışma sürecinde yer alan her iki temel kriptografik işlevin birbirinden bağımsız şekilde de ele alınmasını mümkün kılmasıdır. Her bir senaryoda yalnızca tek bir kriptografik bileşen değiştirilerek (örneğin, yalnızca anahtar değişimi algoritması veya yalnızca dijital imza algoritması), değişimin protokole etkisi analiz edilmeye çalışılmıştır. Bu yöntem, karşılaştırmalı performans ölçümlerinde tutarlılığı sağlamaya yöneliktir. Böylece algoritma kaynaklı gecikmelerin veya uyumsuzlukların daha açık biçimde gözlemlenmesi sağlanmıştır.

3.3 Anahtar Değişim Mekanizmaları

Bu tez kapsamında gerçekleştirilen testlerde, klasik ve post-kuantum anahtar değişim algoritmaları ile hibrit kombinasyonlara dayalı farklı yapılar değerlendirilmiştir. Kullanılan algoritmalar üç ana kategoride sınıflandırılmıştır. (Çizelge 3.1 Test edilen anahtar değişim algoritmaları)

Çizelge 3.1 Test edilen anahtar değişim algoritmaları

Grup	Test Edilen Algoritmalar
Klasik ECDHE	X25519, secp256r1, secp384r1, secp521r1
Post-Kuantum KEM	ML-KEM-512, ML-KEM-768, ML-KEM-1024
Hibrit (ECDHE + PQC)	X25519MLKEM768, SecP256r1MLKEM768

Klasik kategoride PQC karşılaştırmasında referans alacağımız ve yaygın olarak kullanılan modern X25519, secp256r1, secp384r1 ve secp521r1 algoritmaları yer almaktadır. Post-kuantum kategoride ise NIST PQC standardizasyon sürecinde öne çıkan ML-KEM algoritmalarının çeşitli parametre setleri test edilmiştir. Son olarak, hibrit kategoride klasik ECDHE algoritmaları ile PQC tabanlı KEM algoritmalarının birlikte kullanıldığı yapılar incelenmiştir. Bu hibrit yapılar, özellikle kuantum saldırılarına karşı güvenlik sağlarken, mevcut sistemlerle uyumluluğun da korunmasını amaçlamaktadır.
Klasik algoritmaların testlere dahil edilmesinin amacı günümüzde güvenli varsayılan ve en çok kullanılan anahtar değişim algoritmalarının da kurulan test ortamında davranışını izlemek ve kıyaslama için bir referans oluşturulmaktır. Hibrit grup bir klasik bir de PQC algoritmanın kullanılması ile yapılan anahtar değişim senaryolarıdır. Post kuantum grupta

ise yalnızca PQC bir algoritma kullanılarak anahtar değişimi gerçekleştirilmiştir. Her bir grup altında oluşturulan test senaryolarında, TLS protokolünde yalnızca anahtar değişim algoritması değiştirilmiştir, diğer parametreler sabit tutulmuştur. Bu şekilde, test sonuçlarında meydana gelen performans farklılıklarında ve TLS el sıkışmasının başarılı olup olmamasının yalnızca kullanılan anahtar değişimi mekanizmasındaki algoritmadan kaynaklandığından emin olunmuştur.

Bu çalışmada kullanılan tüm anahtar değişim algoritmaları, test senaryoları halinde yapılandırılmıştır. Her senaryo, belirli bir PQC algoritması veya hibrit grup kombinasyonu için ayrı bir yapılandırma ile çalıştırılmış ve istemci-sunucu bağlantısı kurularak TLS el sıkışma süresi ölçülmüştür. Her senaryo için TLS el sıkışma süresi, bağlantı kurulma anından başarılı el sıkışma tamamlanana kadar geçen süre olarak milisaniye cinsinden ölçülmüştür. Başarısız bağlantılar ayrı olarak kaydedilmiş ve analiz dışında bırakılmıştır. Testler, 1000 tekrarlı yapılmış ve her biri için log ve sonuç dosyaları ayrı ayrı oluşturulmuştur. Her test senaryosu karşılık gelen log ve sonuç dosyalarını içerecek biçimde sistematik olarak organize edilmiştir. Bu yapı sayesinde deneylerin yeniden üretilebilirliği ve karşılaştırmalı analizlerin tutarlılığı sağlanmıştır.

Tüm anahtar değişimi senaryolarında bağlantı kurulurken sabit tutulan parametreler Çizelge 3.2 Anahtar değişimi test senaryoları sabit tutulan parametreler'de özetlenmiştir.

Çizelge 3.2 Anahtar değişimi test senaryoları sabit tutulan parametreler

Parametre	Sabit Değer	Açıklama
TLS sürümü	TLS 1.3	Güncel protokol sürümü
Cipher Suite	TLS_AES_256_GCM_SHA384	Modern AEAD
Sunucu kimlik doğrulama	ECDSA-P256 (secp256r1)	Sertifika imzası (sabit)
Sunucu yazılımı	OpenSSL 3.4 + oqsprovider 0.8.0	İlgili KEM/hibrit etkin
İstemci aracı	Test otomasyon scripti + openssl s_client	1000 tekrar

3.3.1 Anahtar değişimi algoritmalarının seçimi

Testlerde karşılaştırma amaçlı kullanılan klasik grubu oluşturan anahtar değişimi algoritmaları seçilirken günümüzde yaygın kullanılan TLS kütüphaneleri araştırılmıştır. Günümüzde yaygın olarak kullanılan TLS kütüphanelerinde desteklenen ve pratikte sıkça kullanılan anahtar değişim algoritmaları Çizelge 3.3 Yaygın kullanılan klasik anahtar değişimi algoritmaları'nda özetlenmektedir. (OpenSSL, n.d.), (BoringSSL, n.d.), (GnuTLS, n.d.), (Mozilla, n.d.), (wolfSSL, n.d.)

Çizelge 3.3 Yaygın kullanılan klasik anahtar değişimi algoritmaları

Algoritma Adı	Kategorisi	TLS Sürümleri	Kullanım Durumu
ECDHE (P256, P-384, P-521)	Klasik / ECC tabanlı	TLS 1.2, TLS 1.3	En yaygın kullanılmakta olandır. Forward Secrecy sağlaması nedeni ile TLS 1.3 de önerilmektedir.
X25519	Klasik / ECC tabanlı	TLS 1.3	Modern ve görece hızlı olduğundan tercih nedenidir.
X448	Klasik / ECC tabanlı	TLS 1.3	Daha yüksek güvenlik seviyesi sağlar ancak daha az yaygındır.
FFDHE (2048, 3072, 4096)	Klasik / Diffie-Hellman	TLS 1.2, TLS 1.3	Büyük anahtar boyutları nedeni ile tercih edilmez.
RSA (Key Exchange)	Klasik / RSA tabanlı	TLS 1.2 ve öncesi	TLS 1.3'te desteklenmemektedir.

Çizelge 3.3 Yaygın kullanılan klasik anahtar değişimi algoritmaları'nda görüldüğü üzere TLS protokolünün günümüzdeki uygulamalarında klasik anahtar değişim algoritmaları arasında en yaygın olarak kullanılanlar Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) ve X25519 algoritmalarıdır. ECDHE, eliptik eğri kriptografisine dayalı olup, farklı güvenlik seviyeleri sunmakta, P-256, P-384 ve P-521 eğrileri ile kullanılabilmektedir. Geçici anahtar çiftleri kullanması sayesinde perfect forward secrecy (PFS) sağlar (Rescorla, 2018). X25519 ise daha modern bir eliptik eğri algoritması olarak, RFC 7748 ile tanımlanmış Curve25519 temelli anahtar değişim mekanizmasıdır ve düşük işlem maliyeti, sabit zamanlı işlemleri ve yaygın desteklenmesi sayesinde TLS 1.3 içinde

varsayılan seçeneklerden biri haline gelmiştir. (Langley, Turner & Dunkelman, 2016). Bu nedenle, bu tezde gerçekleştirilen performans testlerinde klasik algoritma karşılaştırması için ECDHE ve X25519 temel alınmış, kıyaslama açısından referans senaryoları olarak değerlendirilmiştir.

Anahtar kapsülleme mekanizması (Key Encapsulation Mechanism- KEM), belirli koşullar altında iki tarafça ortak bir gizli anahtar oluşturmak için kullanılabilen bir algoritma kümesidir. Bir KEM kullanılarak güvenli bir şekilde oluşturulan ortak bir gizli anahtar, daha sonra şifreleme ve kimlik doğrulama gibi güvenli iletişimlerde temel görevleri gerçekleştirmek için simetrik anahtarlı kriptografik algoritmalarla birlikte kullanılabilir. Bu tezde kullanılan Post-kuantum anahtar kapsülleme mekanizmalarının seçiminde temel yaklaşım, NIST tarafından yürütülen ve standartlaştırma sürecinde öne çıkmış, çeşitli güvenlik, performans ve uygulanabilirlik analizlerinden geçmiş mekanizmaların tercih edilmesidir. Bu nedenle NIST' in üç tur değerlendirmeden sonra seçmiş olduğu CRYSTALS-KYBER ve CRYSTAL-KYBER'dan türetilen ve onun bir varyantı olan ML_KEM anahtar kapsülleme mekanizmaları testlerde kullanılmıştır. CRYSTALS-KYBER lattice tabanlı bir anahtar değişim mekanizması olup Module-LWE problemine dayalı bir yapı sunmaktadır. ML_KEM ise CRYSTALS-KYBER'in bir varyansı olarak NIST tarafından 2024 yılında FIPS 203 standardı ile resmi olarak standartlaştırılmaktadır. (NIST, 2024).

NIST PQC tüm standardizasyon sürecinde değerlendirilen ve OQS Provider tarafından desteklenen başlıca anahtar değişim algoritmalarını göstermektedir. Bu algoritmaların bazıları ya güvenlik açıkları nedeniyle elenmiş ya da uygulama zorlukları ve performans problemleri sebebiyle tercih edilmemiştir. Algoritmalar, temellendikleri matematiksel probleme, seçilip seçilmediğine ve uygulama özelliklerine göre *Çizelge 3.4 Post Quantum KEM algoritmaları*'nda özetlenmiştir.

Çizelge 3.4 Post Quantum KEM algoritmaları

Algoritma Adı	Matematiksel Temel	Açıklama
CRYSTALS-KYBER	Module-LWE	NIST tarafından üçüncü tur sonunda resmi KEM standardı olarak seçilmiştir. Kyber, daha sonra ML-KEM adıyla FIPS 203 standardı altında yayımlanmıştır.
ML-KEM	Module-LWE	Kyber algoritmasının NIST tarafından resmileştirilmiş ve yayımlanmış sürümüdür. ML-KEM, "Module Lattice-based Key Encapsulation Mechanism" ifadesinin kısaltmasıdır. 2024 yılında yayımlanan resmi standardı [FIPS 203 – *ML-KEM: Module-Lattice-Based Key-Encapsulation Mechanism*] olarak belirlenmiştir.
FrodoKEM	LWE (matrix-based)	Klasik (yapılandırılmamış) LWE tabanlı güçlü bir güvenlik temeli sunmasına rağmen, yüksek işlem maliyeti ve büyük anahtar/kapsülleme boyutları nedeniyle pratikte verimsiz bulunmuş ve resmi standart olarak seçilmemiştir.
NTRU	LWE- or LWR based	Güçlü güvenlik ve rekabetçi performans sunmasına rağmen, Kyber'in dengeli yapısı ve daha yaygın kabul görmesi nedeniyle tercih edilmemiştir.
BIKE	Code-based	NIST'in 4. turunda değerlendirilen BIKE, güvenlik analizi ve performans konularındaki tartışmalar nedeniyle resmi standart olarak seçilmemiştir. 2025'te yapılan nihai açıklamada, BIKE dışarıda bırakılmış ve alternatif algoritma statüsü verilmemiştir.
HQC	Code-based	NIST, 2025 yılında HQC algoritmasını ML-KEM'e alternatif olarak yedek algoritma (backup) statüsünde seçmiştir. Güçlü güvenlik özellikleri sunan kod tabanlı bir yapıya sahiptir.

Bu tez çalışmasında gerçekleştirilen deneylerde kullanılmak üzere seçilen anahtar değişim mekanizmaları ML-KEM algoritmaları OQS Provider (v0.8.0) tarafından tam olarak desteklenmektedir ve TLS 1.3 protokolüyle entegrasyonu teknik olarak mümkündür.

Kuantum sonrası döneme geçiş sürecinde, post-kuantum algoritmaların entegrasyonu sırasında karşılaşılan belirsizlikler ve uygulama geçiş riskleri göz önüne alındığında, hibrit kriptografik yaklaşımlar önemli bir ara çözüm olarak öne çıkmaktadır. Hibrit

anahtar değişim mekanizmaları, klasik kriptografik algoritmalar (örneğin ECDHE, X25519) ile post-kuantum algoritmaların (örneğin Kyber, ML-KEM) birlikte kullanıldığı, böylece her iki algoritmanın güvenlik avantajlarının da bir araya getirildiği yapılar olarak tanımlanmaktadır. (Bindel et al., 2020; Stebila et al., 2016).

Hibrit mekanizmaların temel amacı, algoritmalardan birinin kırılması durumunda dahi iletişimin güvenliğini devam ettirebilmektir. Bir hibrit anahtar değişiminde iki ayrı anahtar anlaşması gerçekleştirilir ve elde edilen ortak sırlardan türetilen birleşik bir oturum anahtarı kullanılır. Bu yaklaşım, özellikle kuantum bilgisayarların yeterli olgunluğa ulaşmadığı ve PQC algoritmalarının uzun vadeli güvenilirliğinin henüz kesinleşmediği geçiş dönemlerinde önerilmektedir (NIST, 2022).

Bu tez kapsamında gerçekleştirilen testlerde, aşağıdaki hibrit kombinasyonlar değerlendirilmiştir. Hibrit kombinasyonlarda X25519MLKEM768, SecP256r1MLKEM768 algoritmaları kullanılmıştır. Bu hibrit kombinasyonlar, klasik algoritmaların mevcut altyapılarla uyumluluğunu devam ettirerek ortak çalışabilirliği sağlarken post-kuantum dirençliliği de sağlamaya yönelik bir geçiş stratejisi olarak öne çıkmaktadır. Bu hibrit yapıların test edilmesiyle hem klasik hem de post-kuantum güvenlik seviyeleri sağlanmış, ayrıca farklı klasik gruplarla hibrit kombinasyonların performans ve uyumluluk düzeyleri gözlemlenmiştir. Hibrit algoritmaların seçimi hem RFC taslakları hem de Open Quantum Safe (OQS) projesi kapsamında desteklenen yapıların uygulanabilirliği doğrultusunda belirlenmiştir (OQS,2024).

3.4 Kimlik Doğrulama ve PQC İmza Algoritmaları Test Senaryoları

TLS protokolünde sunucu (ve gerekli durumlarda istemci) kimlik doğrulaması, X.509 sertifikaları aracılığı ile sağlanmaktadır. Bu sertifikalar, bir Sertifika Otoritesi (CA) tarafından imzalanır ve bu imza sunucuya ait açık anahtarın yer aldığı sertifikaya güvenilmesini sağlar. TLS 1.3 protokolünde kimlik doğrulama için kullanılan dijital imzalar, oturum başlatılırken sunucu tarafından gönderilen sertifika ile istemciye sunulur. Bağlantı kurulurken sunucu, imzayı oluşturmak için kullandığı genel anahtarı içeren bir sertifika ile, şimdiye kadarki iletişimin transkripti üzerinde oluşturulmuş bir imza, el

sıkışma imzası gönderir. Bu, istemcinin sunucuyu doğrulamasını sağlar; istemci genel anahtarı onaylayan sertifika yetkilisine güvenip güvenmediğini ve imzanın şimdiye kadar gönderdiği ve aldığı mesajları doğrulayıp doğrulamadığını kontrol eder. Kontrol başarılı ise sunucu kimliği doğrulanmış olur. TLS 1.3 protokolünde de belirtildiği üzere bu işlem, sunucunun kendi özel anahtarıyla 'CertificateVerify' mesajını imzalaması ve gerçekten sertifikada belirtilen açık anahtara sahip olduğunu kanıtlamasıdır. TLS 1.3 protokolü, dijital imza algoritmalarının sertifikalar üzerinde kullanımı için bir sınırlama getirmez ancak PQC imza algoritmalarının doğrudan TLS protokolüne entegre edilmesi için uyumluluk çalışmaları gerekmektedir. Örneğin kullanılan sertifika boyutları, dijital imzanın oluşturulma ve doğrulama süreleri, sertifika ve sertifika zinciri yönetimi süreleri algoritma seçiminde dikkate alınması gerekli hususlardır. Bu tez kapsamında, aşağıdaki post-kuantum dijital imza algoritmaları ile imzalamaya olanak verecek sertifikalar kullanılarak farklı test senaryoları tasarlanmış ve uygulanmıştır. Her bir senaryoda yalnızca imza algoritması değiştirilmiş; TLS versiyonu, şifreleme algoritması (cipher suite) ve el sıkışma yöntemi sabit tutulmuştur. Bütün kimlik doğrulama senaryolarında sabit tutulan ve değiştirilen parametreler *Çizelge 3.5 Kimlik doğrulama test senaryoları sabit tutulan parametreler*'de özetlenmiştir.

Çizelge 3.5 Kimlik doğrulama test senaryoları sabit tutulan parametreler

Parametre	Sabit Değer	Açıklama
TLS Sürümü	TLS 1.3	Güncel TLS sürümüdür.
Cipher Suite	TLS_AES_256_GCM_SHA384	Güçlü ve modern bir AEAD şifreleme algoritmasıdır. TLS 1.3 protokolü tarafından desteklenir.
Anahtar Değişimi	X25519	X25519, eliptik eğri Diffie-Hellman (ECDH) algoritmalarının modern bir türevidir.
Sunucu Yapılandırması	OpenSSL 3.4 + oqsprovider 0.8.0	Test edilmek istenen Post-kuantum imza algoritmasına sahip x509 sertifikaları sunacak şekilde yapılandırılmıştır.
İstemci Aracı	Test otomasyon scripti+ OpenSSL s_client bileşeni	Testler 1000 iterasyonu sağlayacak şekilde peş peşe ve test edilmek istenen post quantum algoritma kimlik doğrulamada kullanılacak şekilde yapılandırılmıştır.

3.4.1 Post-Kuantum dijital imza algoritmalarının seçimi

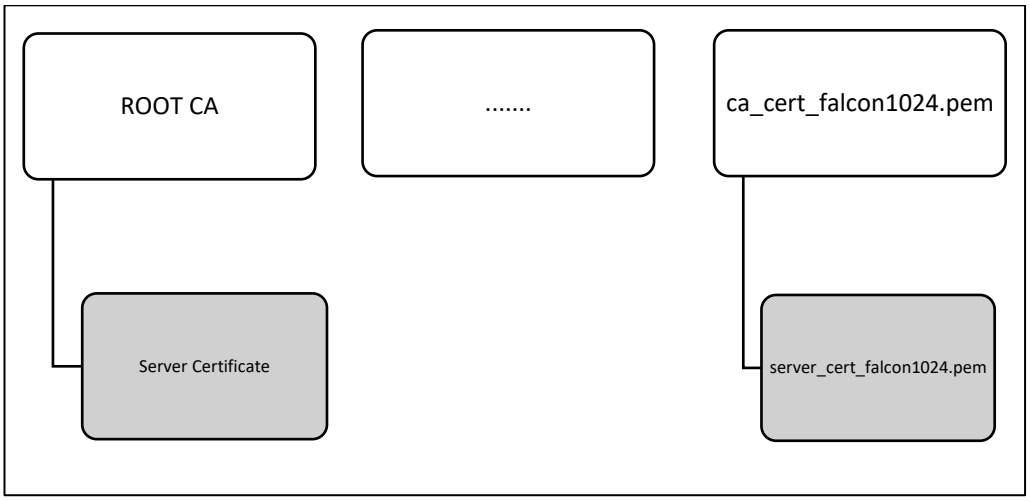
TLS 1.3 Protokolü Eliptik Eğri Kriptografisi (ECC) ve RSA algoritmalarını kullanmaktadır. Bu tez kapsamında yapılan testlerde, TLS 1.3 protokolü kullanılarak kimlik doğrulama amacıyla bunlar yerine farklı post-kuantum dijital imza algoritmaları kullanılmıştır. Bunlar arasında CRYSTALS-Dilithium2 ve Dilithium3 (NIST resmi adı ile ML-DSA), hash tabanlı SPHINCS+ varyantları ile Falcon yer almaktadır. (Çizelge 3.6 PQC dijital imza algoritmaları) Testler sırasında her bir algoritma ile imzalanmış X.509 sunucu sertifikaları ve bu sertifikayı imzalayan kök sertifikaları aynı algoritmalarla oluşturulmuş ve bu sertifikalar OpenSSL ve oqs-provider altyapısı kullanılarak TLS sunucusunda yapılandırılmıştır. Her test senaryosunda yalnızca dijital imza algoritması değiştirilmiş; diğer TLS parametreleri sabit tutulmuştur. Bu sayede, sadece imza algoritmalarının performans ve uyumluluk üzerindeki etkisi izole biçimde analiz edilebilmiştir.

Çizelge 3.6 PQC dijital imza algoritmaları

Algoritma Adı	Matematiksel Temel	Açıklama
CRYSTALS-Dilithium	Modüler kafes (M-LWE, M-SIS)	NIST tarafından ML-DSA olarak standardize edilmiştir. Dengeli güvenlik ve performans sunar, birçok uygulama için önerilen algoritmadır.
SPHINCS+	Hash tabanlı (Merkle, HORST)	NIST tarafından SLH-DSA olarak standardize edilmiştir. En yüksek güvenlik sağlamayı hedefler, imza boyutu büyüktür ancak güvenliği yüksek kabul edilir ve yedek algoritma rolü üstlenir.
Falcon	Kafes (NTRU, GPV tabanlı)	NIST tarafından FN-DSA olarak geliştirilmektedir. Küçük imza boyutlarına sahiptir ancak karmaşık yüzen nokta (floating point) hesapları içerir.
Rainbow	Multivaryant polinom tabanlı	Ward Beullens'in pratik anahtar kurtarma saldırısı nedeniyle elendi. Bu saldırı SL I parametresi için ortalama 53 saatte private key'i kırabiliyordu.
GeMSS	Multivaryant polinom tabanlı	Güçlü matematiksel yapıya rağmen performans sorunları ve güvenlik endişeleri nedeniyle elendi.
PICNIC	Sıfır Bilgi Kanıtları + Blok şifre	Yenilikçi yapıya sahip olsa da yavaş performans ve büyük imza boyutları nedeniyle elendi.

3.4.2 PQC sertifikaları

Testlerdeki sunucu yapılandırmasında TLS 1.3 protokolünde kimlik doğrulama testleri için sertifika hiyerarşileri oluşturulmuştur. Burada gerçek bir açık anahtar altyapısına uygun şekilde kök ve son kullanıcı sertifikasından oluşan sertifika zincirleri kullanılmıştır. Ara sertifika otoritesi (Intermediate CA) kullanılmamış; doğrudan kök CA tarafından imzalanmış sunucu sertifikaları ile sunucular yapılandırılmıştır. (Şekil 3.1 PQC *test sertifika hiyerarşileri*) Sunucu sertifikalarına karşılık gelen özel anahtarlar da ayrıca oluşturulmuştur. Tüm bu dosyalar, sistematik olarak adlandırılmış ve PQC TLS test paketi oluşturulmuştur. Test paketinde önemli bileşenler Çizelge 3.7 PQC TLS sertifikaları ve Çizelge 3.7 PQC TLS sertifikaları (devamı)'nda verilmiştir.



Şekil 3.1 PQC test sertifika hiyerarşileri

1. **Kök Sertifika (CA – Certificate Authority)**: Her bir imza algoritması için ayrı ayrı oluşturulmuştur. Kendinden imzalı (self-signed) bu sertifika, zincirin güven kökünü oluşturur. Örneğin: dilithium3_CA.crt.Kök sertifikada kullanılan algoritma ile sunucu sertifikası algoritması aynı olacak şekilde tasarlanmıştır.
2. **Sunucu Sertifikası (Server Certificate)**: CA kök sertifikası tarafından tarafından imzalanmış, test sunucusunun kimliğini temsil eden sertifikadır. İstemciye el sıkışma sırasında sunulmaktadır. Sertifika imzalarında post-kuantum algoritmalar kullanılmıştır.

Bu sertifikalar genellikle server_cert_<algoritma>.pem adıyla oluşturulmuştur. Örneğin: server_cert_dilithium3.pem. (*Çizelge 3.7 PQC TLS sertifikaları*)

3. **Sertifika Zinciri (Bundle)**: TLS istemcisine (client) gönderilen ve istemcinin güvenli doğrulama yapabilmesini sağlayan dosyadır. Bu dosya, CA ve sunucu sertifikalarının zincir şeklinde birleştirilmiş halidir.
4. **Özel Anahtar (Private Key):** Sertifikalara ait özel anahtarlardır ve yalnızca test ortamında kullanılmak üzere üretilmiştir. (*Çizelge 3.7 PQC TLS sertifikaları*)

Bu bileşenlerden oluşan test paketi sunucu yapılandırmasında kullanılmıştır. Openssl 's_server' komutu ile başlatılan test sunucularında '-cert', '-key' ve '-CAfile' parametreleri aracılığıyla yüklenmiştir. İstemci (openssl s_client) bu zinciri doğrulayarak kimlik doğrulamayı gerçekleştirmiştir. Sertifika zincirlerinin eksiksiz biçimde doğrulanması, algoritmanın doğru uygulanabilirliğini göstermektedir. Bu yöntem aynı zamanda, post-kuantum imza algoritmaları ile oluşturulmuş sertifikaların doğrulama süreleri ve dosya boyutlarının performansa etkisinin görülmesine de olanak sağlamıştır. İlerleyen bölümlerde test sertifikaları ile gerçekleştirilen testlerin karşılaştırmalı analizleri sunulacaktır.

Çizelge 3.7 PQC TLS sertifikaları

Algoritma	Dosya Adı	Dosya Türü
ECDSA-P256	ca_cert_ecdsa_p256.pem	Root CA Sertifikası
	server_cert_ecdsa_p256.pem	Sunucu Sertifikası
	ecdsa_p256.key	Özel Anahtar
ML-DSA-44	ca_cert_mlds44.pem	Root CA Sertifikası
	server_cert_mlds44.pem	Sunucu Sertifikası
	mlds44.key	Özel Anahtar
ML-DSA-65	ca_cert_mlds65.pem	Root CA Sertifikası
	server_cert_mlds65.pem	Sunucu Sertifikası
	mlds65.key	Özel Anahtar
ML-DSA-87	ca_cert_mlds87.pem	Root CA Sertifikası
	server_cert_mlds87.pem	Sunucu Sertifikası
	mlds87.key	Özel Anahtar

Çizelge 3.7 PQC TLS sertifikaları (devamı)

Algoritma	Dosya Adı	Dosya Türü
Falcon-512	ca_cert_falcon512.pem	Root CA Sertifikası
	server_cert_falcon512.pem	Sunucu Sertifikası
	falcon512.key	Özel Anahtar
SPHINCS+-SHA2-128f	ca_cert_sphincs128f.pem	Root CA Sertifikası
	server_cert_sphincs128f.pem	Sunucu Sertifikası
	sphincs128f.key	Özel Anahtar
SPHINCS+-SHA2-192f	ca_cert_sphincs192f.pem	Root CA Sertifikası
	server_cert_sphincs192f.pem	Sunucu Sertifikası
	sphincs192f.key	Özel Anahtar

4. BULGULAR ve TARTIŞMA

4.1 Anahtar Değişim Mekanizmaları Test Sonuçları

Elde edilen TLS 1.3 el sıkışma (handshake) süresi sonuçlarını değerlendirirken iki bakış açısı benimsendi. Birincisi, PQC geçiş döneminde kullanılabilecek hibrit ve salt post-kuantum yaklaşımların, klasik anahtar değişimiyle TLS el sıkışma performansı açısından kıyaslanmasıdır. Amaç, kuantum geçiş sürecine yönelik gerçek dünya yaygınlaştırma senaryolarında öngörü oluşturmaktır. İkincisi ise bu algoritmaların birbirleriyle karşılaştırılması; el sıkışma süresine etkilerinin ve aralarındaki farkların nicel olarak belirlenmesidir. Bu ikinci çerçeve, algoritma seçiminde pratik yol gösterici sonuçlar üretmeyi hedefler. Bu çalışmada raporlanan değerler, yalnızca kriptografik çekirdeğin saf hesaplama süresini değil; TLS 1.3 protokol akışı, ağ gecikmesi (loopback'te minimal), protokol ek yükü ve kriptografik işlemlerin toplamını yansıtan tam el sıkışma süresidir. Bu yaklaşım, pratik dağıtım ortamlarındaki nihai kullanıcı deneyimini daha gerçekçi biçimde temsil eder.

Çizelge 4.1 Anahtar değişim algoritmaları test sonuçları

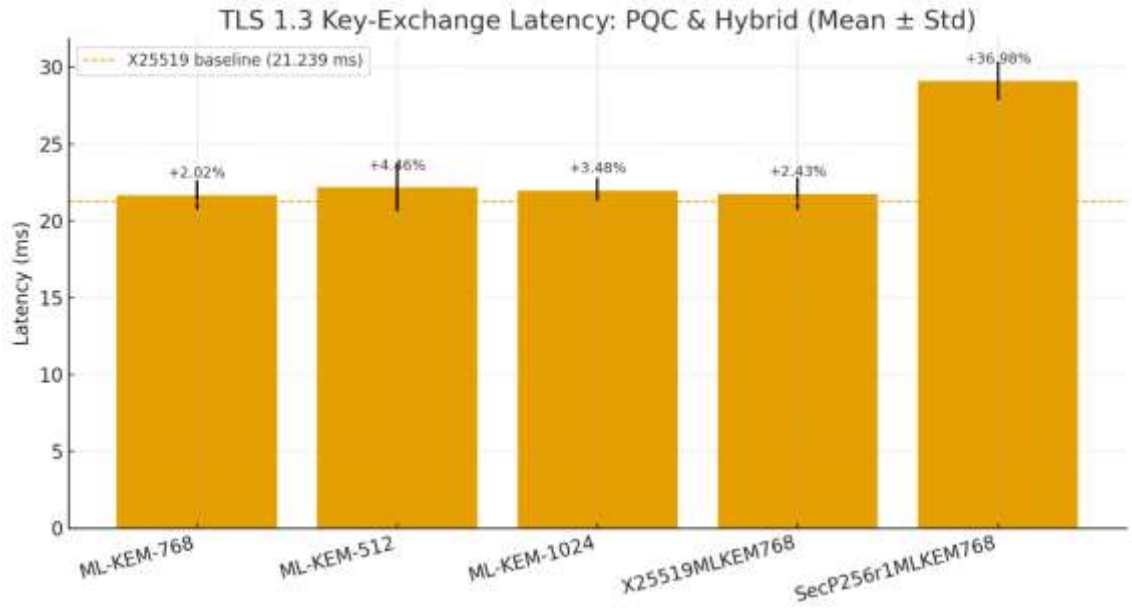
Grup	Ortalama ± Std (ms)	Δ% (X25519'a göre)
X25519	21.239 ± 0.761	---
secp256r1	19.803 ± 73.006	−6.76%
secp384r1	21.931 ± 71.634	+3.26%
Secp521r1	25.583 ± 73.114	+20.45%
ML-KEM-768	21.669 ± 0.945	+2.02%
ML-KEM-512	22.186 ± 1.577	+4.46%
ML-KEM-1024	21.977 ± 0.806	+3.48%
X25519MLKEM768	21.755 ± 1.066	+2.43%
SecP256r1MLKEM768	29.093 ± 1.248	+36.98%

Bu ölçümler Çizelge *4.1* Anahtar değişim algoritmaları test sonuçları ve Şekil *4.1* TLS 1.3 Protokolü anahtar değişim algoritmaları el sıkışma süreleri grafiğinde özetlenmiştir. Bulgular, ML-KEM ailesinin TLS 1.3 el sıkışmasında X25519'a yalnızca birkaç yüzde puan farkla yakın gecikmeye sahip olduğunu göstermektedir. Lokal ölçümlerde X25519 için 21.239 ± 0.761 ms olarak belirlenen referans değere karşılık, ML-KEM-768 21.669 ± 0.945 ms ile +%2.02, ML-KEM-1024 21.977 ± 0.806 ms ile +%3.48, ML-KEM-512 ise 22.186 ± 1.577 ms ile +%4,46 düzeyinde ek maliyet üretmiştir. Bu sapma aralığı, uygulama katmanında hissedilebilirlik bakımından çoğu pratik senaryo için düşüktür; özellikle istemci/sunucu kaynaklarının modern donanım özellikleri (örn. AES-NI, çok çekirdek) ile desteklendiği ortamlarda fark, son kullanıcı deneyimi açısından görünmez düzeye yaklaşır.

Bir diğer önemli bulgu, güvenlik seviyesi arttıkça maliyetin doğrusal olmayan ancak kararlı biçimde seyretmesidir: ML-KEM-768 ve ML-KEM-1024'ün X25519'a yakın kalması, yüksek güvenlik seviyeleri hedefleyen kurulumlarda dahi saf PQC'nin pratikte hesaplama açısından diğerleri ile rekabetçi olduğunu teyit eder. Bu durum, orta vadede saf PQC tasarımların yalnızca kriptografik sağlamlık nedeniyle değil, performans rasyonalitesi nedeniyle de benimsenebileceğini göstermektedir. Operasyonel açıdan saf PQC tercihi; protokol basitliği, konfigürasyon yönetimi ve hata ayıklama süreçlerinde de ek faydalar sunar; hibrit yapıların getirebileceği karmaşıklık burada söz konusu değildir.

Hibrit yaklaşım, geriye dönük uyumluluk ve geçiş kolaylığı nedeniyle kurumsal ortamlarda güçlü bir seçenek olarak görünmektedir. Bu çalışmada test edilen X25519MLKEM768 kombinasyonu 21.755 ± 1.066 ms ile X25519'a göre yalnızca +%2.43 ek maliyet üretmiş; böylece "hemen kullanılabilir" / "hemen dağıtılabilir" izlenimini sayısal olarak desteklemiştir. Bu bulgu iki açıdan kritiktir: (i) Hibrit, kuantum sonrası döneme hazırlık amacıyla risk azaltımı sağlar; klasik güvenliğin yanına PQC güvenliğini ekler. (ii) Bunu yaparken gecikme maliyetini tüketmez; özellikle kısa etkileşimli bağlantıların yoğun olduğu web servislerinde ek yükün %1–3 bandında kalması, kapasite planlaması açısından yönetilebilir bir etkidir. Ayrıca hibritin aşamalı geçiş senaryolarında sağladığı esneklik (örneğin, istemci popülasyonunun bir kısmı PQC desteklerken bir kısmının klasik kalabilmesi) gerçek dünyada kurumsal dönüşümlerin

kesintisiz ilerlemesine yardımcı olur. Güvenlik mimarisi açısından ise hibritin kriptografik çeşitliliği artırması ve tek arıza noktası (single point of failure) riskini azaltma etkisi de değerlidir. Klasik ECDHE ve KEM birlikte kullanıldığında, tek bir varsayımın kırılmasına karşı savunma derinliği oluşacaktır. Hibrit tasarımlarda ECDHE eğri seçimi performans üzerinde belirleyici olduğu görülmüştür. Bu çalışma kapsamında incelenen SecP256r1MLKEM768 kombinasyonu, 29.093 ± 1.248 ms ile X25519 referansına kıyasla +%36,98 ek gecikme üretmiştir. Bulgular, hibrit maliyetinin yalnızca KEM tarafından değil; ECDHE tarafındaki eğri seçimi ve iki mekanizmanın kümülatif işi tarafından belirlendiğini göstermektedir.



Şekil 4.1 TLS 1.3 Protokolü anahtar değişim algoritmaları el sıkışma süreleri

4.2 Post-Kuantum Dijital İmza Algoritmalarına Dayalı Sunucu Kimlik Doğrulama Test Sonuçları

Bu bölümde, anahtar değişimin X25519'da sabit tutulduğu tam el sıkışma (full-handshake) ölçümleri altında sunucu kimlik doğrulama algoritmalarının gecikmeye etkisi karşılaştırılmıştır. Tüm değerler N=1000 tekrarın ortalama ± std (ms) biçimindedir; Δ% değerleri ECDSA-P256 temel alınarak hesaplanmıştır. Ölçümler önceki bölümle aynı test

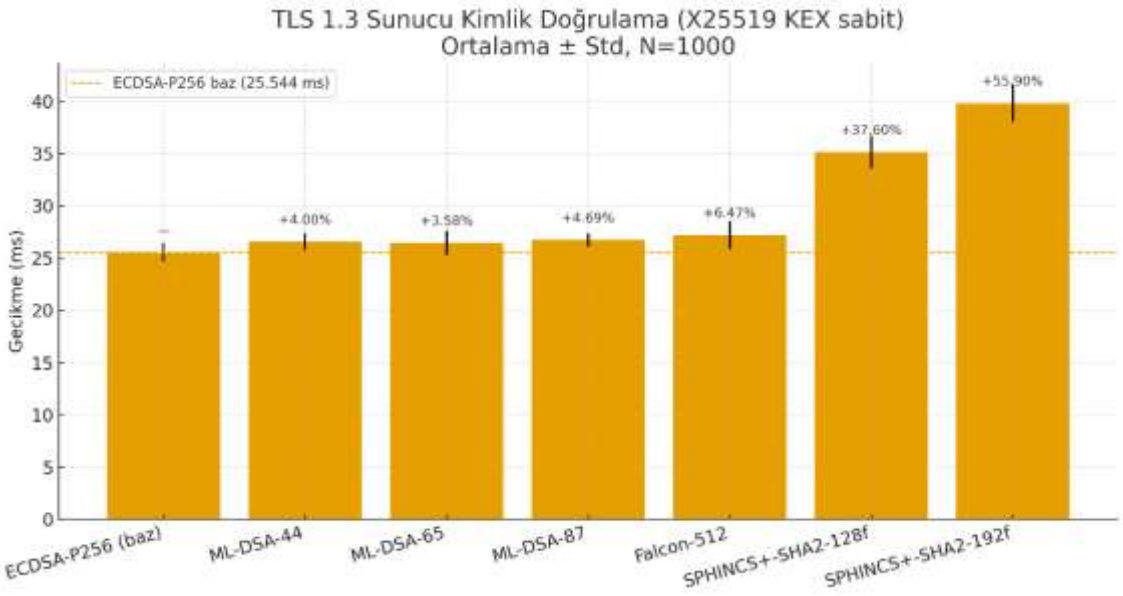
koşullarında (TLS 1.3, AES-256-GCM, ECDSA tabanlı sertifika zinciri, loopback, başarısız denemeler hariç) yürütülmüştür.

Çizelge 4.2 Sunucu kimlik doğrulama PQC algoritmaları test sonuçları

Algoritma	Ortalama ± Std (ms)	Δ% (ECDSA-P256'a göre)
ECDSA-P256	25.544 ± 0.885	—
ML-DSA-44	26.564 ± 0.828	+4.00%
ML-DSA-65	26.458 ± 1.166	+3.58%
ML-DSA-87	26.742 ± 0.652	+4.69%
Falcon-512	27.197 ± 1.360	+6.47%
SPHINCS+-SHA2-128f	35.136 ± 1.565	+37.60%
SPHINCS+-SHA2-192f	39.813 ± 1.769	+55.90%

Tüm sonuçlar Çizelge 4.2: Sunucu Kimlik Doğrulama PQC Algoritmaları Test Sonuçları ve Şekil 4.2: Sunucu Kimlik Doğrulaması PQC Algoritmaları Test Sonuçları'nda ayrıntılı olarak sunulmuştur. Gerçekleştirilen ölçümler, post-kuantum imza algoritmalarının klasik ECDSA-P256 temelli kimlik doğrulama süreciyle kıyaslandığında ortaya koyduğu hesaplama yükünü ve performans etkilerini göstermektedir. Elde edilen bulgular, ML-DSA ailesinin performans bakımından oldukça dengeli olduğunu ortaya koymaktadır. Bu algoritma, ECDSA-P256 referansına kıyasla yalnızca yaklaşık %3–5 aralığında sınırlı bir ek işlem maliyeti (overhead) oluşturmuş; farklı güvenlik seviyeleri arasında da benzer ölçüde tutarlı bir performans farkı gözlenmiştir. Bu durum, ML-DSA algoritmasının hem güvenlik hem de verimlilik açısından kurumsal TLS altyapılarına entegrasyon açısından pratik bir seçenek olduğunu ortaya koymaktadır. Benzer şekilde, Falcon-512 algoritması da yaklaşık %6,5 düzeyinde bir ek maliyet üretmiştir. Buna karşın, SPHINCS+ ailesi hash tabanlı yapısının doğası gereği belirgin düzeyde hesaplama gecikmesi yaratmıştır. İmzalama ve doğrulama süreçlerindeki bu yüksek işlem maliyeti, SPHINCS+ algoritmasının genel kullanımlardan ziyade özel kullanım alanlarına daha uygun olabileceğini göstermektedir. Sonuç olarak, yapılan karşılaştırmalar post-kuantum kimlik

doğrulamasının ML-DSA ve Falcon algoritmaları ile pratik olarak dağıtılabilir ve performans açısından sürdürülebilir bir çözüm olduğunu göstermektedir. SPHINCS+ ise güvenlik/performans dengesi bakımından daha özel uygulamalarda değerlendirilebilecek bir alternatif olarak düşünülebilir. Bu bulgular, PQC tabanlı kimlik doğrulama mekanizmalarının TLS 1.3 bağlamında gerçek ortam koşullarında uygulanabilirliğini desteklemektedir.



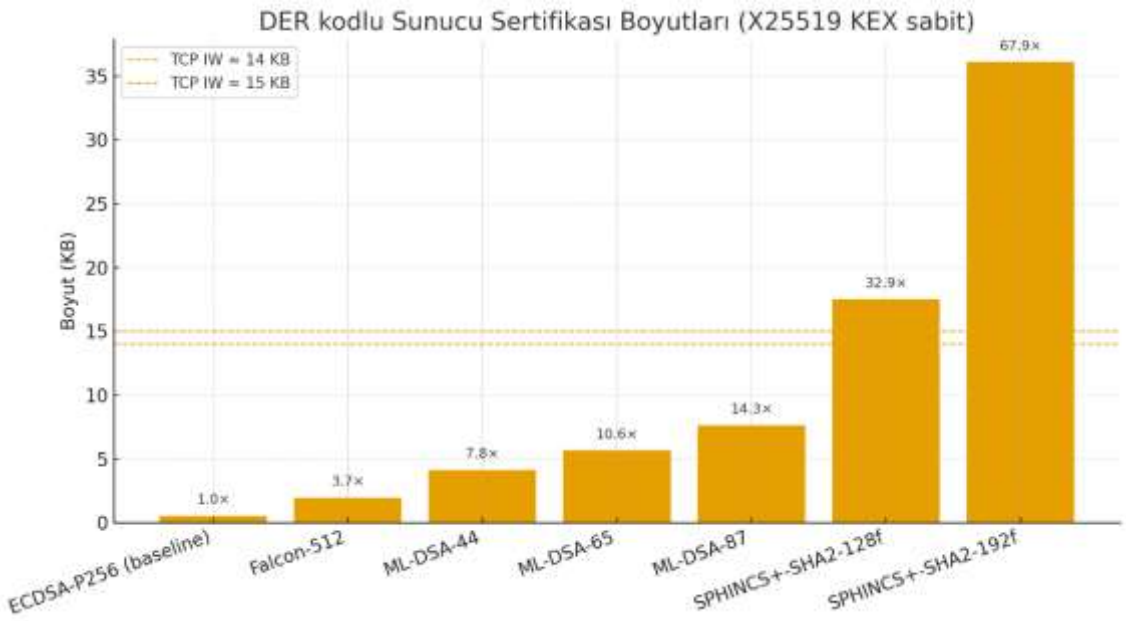
Şekil 4.2 Sunucu kimlik doğrulaması PQC algoritmaları test sonuçları

Çizelge 4.3 DER kodlu sunucu sertifikası boyutları

Algoritma	Boyut (KB)	Baz'a göre (×)
ECDSA-P256 (baz)	0.532	1.0×
Falcon-512	1.946	≈ 3.7×
ML-DSA-44	4.130	≈ 7.8×
ML-DSA-65	5.659	≈ 10.6×
ML-DSA-87	7.617	≈ 14.3×
SPHINCS+-SHA2-128f	17.510	≈ 32.9×
SPHINCS+-SHA2-192f	36.102	≈ 67.9×

Test sonuçlarına göre bir diğer değerlendirme ise kimlik doğrulama sertifikalarının boyutu üzerine yapılmıştır (Çizelge *4.3* DER kodlu sunucu sertifikası boyutları ve Şekil *4.3* DER kodlu sunucu sertifikası boyutları). Tipik bir ağda MSS ≈ 1460 B ve başlangıç tıkanıklık penceresi (IW) ≈ 10 segment ≈ 14–15 KB kabul edildiğinde, ECDSA-P256 ve Falcon-512 sertifikaları tek uçuşta (single flight) taşınabilir. Buna karşılık SPHINCS+-128f (~17.5 KB) IW'yi aşabilir, SPHINCS+-192f (~36 KB) ise kesinlikle aşar; bu da ek RTT'ler gerektirebilir. Gerçek dağıtımlarda zincirde ara sertifikalar (intermediate) da gönderildiğinden, zincir uzunluğu ağ üzerinden iletilen toplam veri miktarını katlayarak artırır. Ayrıca, gerçek ortamlarda sertifikalardaki CT (Certificate Transparency) SCT uzantıları gibi ilave kayıtlar da el sıkışmadaki aktarım hacmini büyütebilir.

Sonuç olarak, hesaplama maliyeti düşük bile olsa (örn. ML-DSA, Falcon), sertifika boyutu özellikle yüksek RTT ya da dar bant (mobil/IoT) bağlantılarda uçtan uca el sıkışma süresinin baskın bileşeni hâline gelebilir.Bu nedenle, dağıtım açısından zinciri yalın tutmak (gereksiz ara sertifikalardan kaçınmak), küçük imza/sertifika profilleri sunan şemaları önceliklendirmek (örn. ECDSA-P256 → geçişte Falcon-512 veya ML-DSA), mümkün olduğunda sertifika sıkıştırma (istemci desteği varsa) ve kısa yaşam süreli ara sertifikalar gibi uygulamak çözüm olarak düşünülebilir. Bu tedbirler, hem tek uçuş olasılığını artırarak ek RTT riskini azaltır, hem de yüksek gecikmeli bağlantılarda sertifika boyutunun el sıkışmaya etkisini sınırlayacaktır.



Şekil 4.3 DER kodlu sunucu sertifikası boyutları

4.3 Ekosistem Uyumluluğu

Bu tezde ölçülen değerler, PQC'nin teknik olarak uygulanabilir olduğunu gösterirken, geniş ölçekli yaygınlaştırma için TLS ekosisteminin da uyumlanması zorunludur. Bunun için X.509 sertifika profillerinin PQC algoritmaları ve ilgili parametreleri destekleyecek biçimde güncellenmesi; tarayıcılar ve işletim sistemlerinin doğrulaması (zincir inşası), OCSP/CRL, Certificate Transperency (CT) log gibi mekanizmaları PQC imza şemalarıyla uyumlu hale gelmesi; ayrıca sertifika otoritelerinin kök ve ara seviyelerde PQC veya hibrit sertifikalar üretmesi gereklidir. Ekosistem otoriteleri (Tarayıcı/OS üreticileri ve CA/Browser Forum) de güncel algoritmalar ile uyumluluk için net bir yol haritası sunmalıdır.

Performans bulgularına dayalı yol haritası üç noktada özetlenebilir:

(i) Hesaplama maliyeti düşük PQC/hibrit seçenekler (ör. ML-DSA, Falcon) bile sertifika boyutu ve zincir uzunluğu nedeniyle ağ üzerinden iletilen toplam veri miktarını anlamlı biçimde artıracaktır. Özellikle yüksek RTT veya dar bant bağlantılarda el sıkışma süresi de buna bağlı artacaktır. Bu durum göz önünde bulundurulmalıdır.

(ii) Geçiş stratejisi olarak, önce hibrit (ör. X25519MLKEM768 + PQC imza) ve küçük imza/sertifika profili sunan şemalarla başlamak; ekosistem desteği olgunlaştıkça saf PQC yapılarına kademeli geçiş izlemek, riski azaltarak uyumluluğu artıracaktır.

(iii) Operasyonel olarak; pilot uygulamalar ile trafikte zincir boyutu ve RTT etkisini ölçmek, kısa zincir ve optimize edilmiş sertifika profilleri tercih etmek düşünülebilir.

5. SONUÇ

Tez çalışmamda gerçekleştirdiğim testler, post-kuantum kriptografi (PQC) algoritmalarının TLS 1.3 protokolünde kullanılabilir olduğunu göstermiştir. Elde edilen veriler, klasik ve post-kuantum yaklaşımların el sıkışma süresi bakımından benzer düzeyde olduğunu göstermiştir.

Anahtar değişimi açısından, ML-KEM (512/768/1024) saf PQC anahtar değişimi X25519 referansına yalnızca +%2–%4,5 aralığında gecikme üreterek hesaplama açısından rekabetçi bir profil sunmaktadır. Hibrit tarafta X25519MLKEM768 yalnızca +%2.43 ek yükle hemen dağıtılabilir bir seçenekken, SecP256r1MLKEM768'in +%36,98 ek maliyeti, hibritte eğri seçiminin kritik olduğunu ve pratikte hibritte X25519'un tercih edilmesi gerektiğini göstermektedir. Klasik grupta X25519 en düşük gecikmeyi verir; secp384r1 sınırlı artış, secp521r1 belirgin ek maliyet üretir; secp256r1 ortalamada düşük görünse de yüksek varyans nedeniyle X25519'un temel referans kabul edilmesi daha uygundur. Dağıtım açısından, kısa vadede X25519MLKEM768 ile başlamak ve orta vadede ML-KEM-768/1024 saf PQC'ye kademeli geçiş makul bir yol haritası sunar. Bu bulgular TLS 1.3, ECDSA-P256, AES-256-GCM ve loopback ortamında N=1000 iterasyonla elde edilmiştir; gerçek ağ koşullarında mutlak süreler artsa da göreli sıralamanın büyük ölçüde korunması beklenir.

Sunucu kimlik doğrulaması sabit tutularak (X25519 anahtar değişimi ile) incelendiğinde, ML-DSA ailesinin minimal ek gecikme ile post-kuantum kimlik doğrulaması sağladığı görülmektedir. ML-DSA-44/65/87 algoritma varyantlarına bakıldığında ECDSA-P256 (25.544 ± 0.885 ms) üzerine yalnızca +%3–%5 artışa neden olmuştur. Falcon-512 ise (27.197 ± 1.360 ms; +%6.47) ML-DSA ailesine göre daha maliyetlidir. SPHINCS+ (SHA2-128f ≈ +%37,6; SHA2-192f ≈ +%55,9) hash-tabanlı olması sebebiyle yüksek hesaplama maliyeti üretmektedir. Bu yönü ile daha çok performans duyarlılığı düşük ancak güvenlik gereksinimi yüksek uygulamalara uygun görünmektedir.

Test ortamında sertifika boyutlarına göre de bir değerlendirme yapılmıştır. DER kodlu sunucu sertifikalarında ECDSA-P256 ≈ 0.532 KB boyutuna karşı Falcon-512 ≈ 1.946 KB

(~3.7×), ML-DSA sertifikaları ≈ 4.13/5.66/7.62 KB (~7.8×/~10.6×/~14.3×) olarak ölçülmüştür. SPHINCS+ tarafında SHA2-128f ≈ 17.5 KB (~32.9×) ve SHA2-192f ≈ 36.1 KB (~67.9×) en büyük boyutlara sahiptir. Bir sunucu – istemci bağlantısında MSS ≈ 1460 B ve başlangıç tıkanıklık penceresi (IW) ≈ 10 MSS ≈ 14–15 KB'dir. Bu, sunucunun ilk RTT içinde ACK beklemeden en fazla ~14–15 KB veri gönderebileceği anlamına gelir; kısaca tek uçuşta veri gönderilir. ECDSA-P256 (~0.532 KB) ve Falcon-512 (~1.946 KB) bu sınırın çok altındadır; sertifika tek uçuşta sığacaktır. Ancak SPHINCS+-128f (~17.5 KB) bu sınırı aşabilir, SPHINCS+-192f (~36 KB) ise belirgin şekilde aşar. IW aşıldığında, sunucu kalan baytları göndermeden önce ACK beklemek zorunda kalacak; bu da en az bir ek RTT (hatta zincirde ara sertifikalar da varsa birden fazla RTT) anlamına gelecektir. Örneğin RTT = 50 ms ise tek ek RTT ≈ 50 ms ek süre yarattığı düşünüldüğünde; mobil/uzak bağlantılarda RTT = 150 ms olduğunda bu etki 150–300 ms düzeyine çıkabilir. Kısacası, hesaplama maliyeti düşük olsa bile, sertifika boyutu/zincir uzunluğu IW'yi aştığında uçtan uca el sıkışma süresine belirgin gecikme ekleyecektir. Bu yüzden küçük boyutlu sertifikalar ve kısa zincirler tercih edilmelidir. Gerçek ortamlarda sertifika hiyerarşisinde ara sertifikalar da bulunabileceğinden zincir uzunluğu katlanarak artacak ve etki daha büyük olacaktır. Dolayısıyla, hesaplama maliyeti düşük olsa bile sertifika boyutu da el sıkışma süresinin belirleyici bir özelliği hâline gelebilir. Zinciri sade tutmak (yalnızca kök ve son kullanıcı hiyerarşisi kullanmak, ara sertifika makamı kullanmamak) ve mümkünse küçük boyutlu imza ailelerini (örn. Falcon-512 veya ML-DSA) tercih etmek yerinde olacaktır.

Genel olarak, bu çalışma PQC'nin TLS 1.3'e entegrasyonunu test ortamında doğrulamış, farklı senaryolarda algoritma-bazlı kıyaslamalara değinmiştir. Teknik sonuçlar, PQC'nin TLS 1.3 protokolünde gerçek ortamda dağıtılabilir olduğunu göstermiştir. Ancak ekosistemin hazır olması, ortak çalışabilirlik ve standartlaşma yönü de geçiş için önemlidir. PQC algoritmalarının X.509 profillerine uygulanması, tarayıcıların ve işletim sistemlerinin yeni algoritmaları desteklemesi ve bu hususta birincil otorite olan CA/Browser Forum'un yönlendirmeleri gereklidir. Geniş ölçekli geçişlerin zaman alacağı da düşünülerek performansı uygun şemalarla başlanması ve uyumluluk gereksinimlerinin gözetilmesi ekosistemde PQC geçişi için planlama yapılması ihtiyacı bulunmaktadır.

KAYNAKLAR

Albrecht, M. R., et al. 2022. A survey on post-quantum cryptography landscape. ACM Computing Surveys, 55(7), Article 142.

Alkim, E., Ducas, L., Pöppelmann, T., & Schwabe, P. 2016. Post-quantum key exchange – a new hope. In 25th USENIX Security Symposium (pp. 327–343). USENIX Association.

Anonymous. 2021. IT Security Guidelines for Transport Layer Security 2.1. National Cyber Security Centre (NCSC-NL), Hollanda. Erişim: https://english.ncsc.nl/publications/publications/2021/january/19/it-security-guidelines-for-transport-layer-security-2.1 (Erişim Tarihi: 07.04.2025).

Anonymous. 2024. TLS 1.2 Vulnerability. Web Sitesi: https://software.land/tls-1.2-vulnerability/ (Erişim Tarihi: 08.12.2024).

AppViewX. 2025. Why Is TLS 1.3 Better And Safer Than TLS 1.2? Erişim: https://www.appviewx.com/blogs/why-is-tls-1-3-better-and-safer-than-tls-1-2/

Beullens, W. 2022. Rainbow is broken. IACR ePrint Archive, Report 2022/214. Erişim: https://eprint.iacr.org/2022/214.pdf

Bernstein, D. J., Buchmann, J., & Dahmen, E. 2008. Post-Quantum Cryptography. Springer.

Bernstein, D. J., & Lange, T. 2017. Post-quantum cryptography. Nature, 549(7671), 188–194. https://doi.org/10.1038/nature23461

Bernstein, D. J., Lange, T., & Peters, C. 2008. Classic McEliece: conservative code-based cryptography.

Bernstein, D. J., et al. 2008. Classic McEliece. Cryptology ePrint Archive, Report 2009/272.
Erişim: https://eprint.iacr.org/2009/272

Bos, J. W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., & Stehlé, D. 2018. CRYSTALS-Kyber: A CCA-Secure Module-Lattice-Based KEM. IEEE European Symposium on Security and Privacy (Euro S&P).

Bos, J. W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., & Stebila, D. 2019. Post-quantum key exchange for the TLS protocol from the NIST Round 2 candidates. IEEE Symposium on Security and Privacy, 524–539.

BSI (Federal Office for Information Security). 2018. Technical Guideline BSI TR-03116-4: Cryptographic Mechanisms – Recommendations and Key Lengths, Part 4. Erişim:

https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TR03116/BSI-TR-03116-4.pdf

BoringSSL. (n.d.). BoringSSL. Erişim: https://boringssl.googlesource.com/boringssl

Castryck, W., et al. 2022. An Efficient Key Recovery Attack on SIDH (SIKE). EUROCRYPT 2023. Erişim: https://eprint.iacr.org/2022/975

Castryck, W., & Decru, T. 2022. An Efficient Key Recovery Attack on SIDH. IACR ePrint Archive, Report 2022/975.

Chen, L. K., Jordan, S., Liu, Y. K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. 2016. Report on Post-Quantum Cryptography. NISTIR 8105. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8105

Couveignes, J.-M. 1997. Hard Homogeneous Spaces. Cryptology ePrint Archive.

Dierks, T., & Rescorla, E. 2008. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246. Internet Engineering Task Force (IETF). Erişim: https://www.rfc-editor.org/info/rfc5246

Dierks, T., & Rescorla, E. 2018. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. Internet Engineering Task Force (IETF). Erişim: https://www.rfc-editor.org/info/rfc8446 (Erişim Tarihi: 07.12.2024).

Ding, J., Gower, J. E., & Schmidt, D. S. 2004. Multivariate Public Key Cryptosystems. Springer.

Ducas, L., et al. 2018. CRYSTALS-Dilithium: Digital Signatures from Module-Lattices. Erişim: https://pq-crystals.org/dilithium/

Fluhrer, S., Kampanakis, P., & Stebila, D. 2022. Hybrid Key Exchange in TLS 1.3 (draft-ietf-tls-hybrid-design-05). IETF Internet-Draft. Erişim: https://datatracker.ietf.org/doc/html/draft-ietf-tls-hybrid-design

GnuTLS. (N.d.). GnuTLS- Transport Layer Security Library. Erişim: https://www.gnutls.org

Gueron, S., & Schanck, J. M. 2019. Falcon: Fast-Fourier Lattice-Based Compact Signatures over NTRU. In Post-Quantum Cryptography (pp. 111–129). Springer. https://doi.org/10.1007/978-3-030-25510-7_7

Hoffstein, J., Pipher, J., & Silverman, J. H. 1998. NTRU: A ring-based public key cryptosystem. Lecture Notes in Computer Science, 1433, 267–288.

Hülsing, A., Butin, D., Gazdag, S., Rijneveld, J., & Mohaisen, A. 2019. SPHINCS+: Submission to the NIST Post-Quantum Project, Version 3.1. Erişim: https://sphincs.org/data/sphincs+-reference-implementation-20220329.pdf

Hülsing, A., Butin, D., Gazdag, S., et al. 2018. XMSS: Extended Hash-Based Signatures. RFC 8391.

Hülsing, A., Rijneveld, J., Schwabe, P., & Szydlo, M. 2020. High-speed key encapsulation from NTRU. IEEE European Symposium on Security and Privacy (EuroS&P), 150–167. https://doi.org/10.1109/EuroSP48549.2020.00017

IETF. 2022. RFC 9180 – Hybrid Public Key Encryption. Internet Engineering Task Force.
Erişim: https://www.rfc-editor.org/info/rfc9180

Jao, D., & De Feo, L. 2011. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In PQCrypto 2011 (pp. 19–34). Springer.

Kee, J., Wouters, P., Fu, D., & Bajaj, S. 2016. Compression of Raw Public Keys in Elliptic Curve Cryptography (ECC). RFC 7924. Internet Engineering Task Force. Erişim: https://www.rfc-editor.org/info/rfc7924

Krawczyk, H., Eronen, P., & Arkko, J. 2010. HMAC-based Extract-and-Expand Key Derivation Function (HKDF). RFC 5869. Erişim: https://www.rfc-editor.org/info/rfc5869

Langley, A., Turner, S., & Dunkelman, O. 2016. RFC 7748: Elliptic Curves for Security. IETF. https://doi.org/10.17487/RFC7748

McEliece, R. J. 1978. A Public-Key Cryptosystem Based on Algebraic Coding Theory. DSN Progress Report, 42–44.

Merkle, R. C., & Hellman, M. E. 1978. Hiding information and signatures in trapdoor knapsacks. IEEE Transactions on Information Theory, 24(5), 525–530.

Mosca, M. 2018. Cybersecurity in an era with quantum computers: Will we be ready? IEEE Security & Privacy, 16(5), 38–41.

Mosca, M., Stebila, D., & Whyte, W. 2019. Hybrid key exchange in TLS 1.3. Cryptology ePrint Archive, Report 2019/1447. https://eprint.iacr.org/2019/1447

Mozilla. (n.d.). Network Security Services (NSS). Erişim: https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS

MbedTLS. (n.d.). Mbed TLS. Erişim: https://github.com/Mbed-TLS/mbedtls

National Institute of Standards and Technology (NIST). 2022. Status Report on the Third Round of the NIST PQC Standardization Process. Erişim: https://csrc.nist.gov/

NIST. 2023. Post-Quantum Cryptography Standardization Project. Erişim: https://csrc.nist.gov/projects/post-quantum-cryptography

NIST. 2024a. FIPS 203: Module Lattice-Based Key-Encapsulation Mechanism (ML-KEM). Erişim: https://doi.org/10.6028/NIST.FIPS.203

NIST. 2024b. FIPS 204: CRYSTALS-Dilithium. Erişim: https://doi.org/10.6028/NI .FIPS.204

NIST. 2024c. FIPS 205: Stateless Hash-Based Signature Standard (SPHINCS+). https://doi.org/10.6028/NIST.FIPS.205

NIST. 2024. Transition to Post-Quantum Cryptography. NIST IR 8547 (Initial Public Draft).

NIST. 2024. Post-Quantum Cryptography Project. Erişim: https://csrc.nist.gov/projects/post-quantum-cryptography (Erişim Tarihi: 10.12.2024).

NIST. 2024. Post-Quantum Cryptography Standardization Project – Round 3 Finalists Announcement. Erişim: https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization#round-3 (Erişim Tarihi: 11.05.2025).

Okamoto, T., Tanaka, S., & Uchiyama, S. 2003. New knapsack type public-key cryptosystem based on MOD knapsack. IEICE Transactions on Fundamentals, E86-A (1), 1–8.

Open Quantum Safe (OQS). 2023. The Open Quantum Safe Project. Erişim: https://openquantumsafe.org

Open Quantum Safe Project. 2024. OQS-Provider: A Post-Quantum Cryptography Provider for OpenSSL. Erişim: https://github.com/open-quantum-safe/oqs-provider

OpenSSL. (N.d.). The OpenSSL Project. Erişim: https://www.openssl.org

OpenSSL Project. 2023. OpenSSL 3.0 and Beyond: Provider-Based Architecture and FIPS. Erişim: https://www.openssl.org/docs/man3.0/man7/provider.html

Rescorla, E. 2018. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, IETF. https://doi.org/10.17487/RFC8446

Rostovtsev, A., & Stolbunov, A. 2006. Public-key cryptosystem based on isogenies.

Shamir, A. 1982. A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem. 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982).

Shor, P. W. 1994. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings 35th Annual Symposium on Foundations of Computer Science. IEEE, 124–134. https://doi.org/10.1109/SFCS.1994.365700

SPHINCS+ Team. 2019. SPHINCS+ Submission to the NIST Post-Quantum Cryptography Project. Erişim: https://sphincs.org/data/sphincs+-submission-nist.pdf

Stebila, D., Campbell, B., Hülsing, A., et al. 2021. Open Quantum Safe Project: Making Post-Quantum Cryptography Usable. IEEE Security & Privacy, 19(5), 66–71. https://doi.org/10.1109/MSEC.2021.3106181

Wouters, P., Tschofenig, H., Gilmore, J., Weiler, S., & Kivinen, T. 2014. Using Raw Public Keys in the Transport Layer Security (TLS) Protocol. RFC 7250, IETF. Erişim: https://www.rfc-editor.org/info/rfc7250

WolfSSL. (N.d.). Library.Erişim:https://www.wolfssl.com Embedded SSL/TLS Library .Erişim:https://www.wolfssl.com