KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

DOKTORA TEZİ

KUANTUM ANAHTAR DAĞITIM SİSTEMLERİNDE VERİMLİ BİLGİ UZLAŞTIRMA İÇİN OPTİMUM CASCADE PROTOKOLÜ

METİN TOYRAN

KOCAELİ 2019

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ

ANABİLİM DALI

DOKTORA TEZİ

KUANTUM ANAHTAR DAĞITIM SİSTEMLERİNDE VERİMLİ BİLGİ UZLAŞTIRMA İÇİN OPTİMUM CASCADE PROTOKOLÜ

METİN TOYRAN

Dr. Öğr. Üyesi Sıtkı ÖZTÜRK
Danışman, Kocaeli Üniversitesi

Prof. Dr. Cabir VURAL
Jüri Üyesi, Marmara Üniversitesi

Dr. Öğr. Üyesi Sultan Aldırmaz ÇOLAK
Jüri Üyesi, Kocaeli Üniversitesi

Doç. Dr. Kerem KÜÇÜK
Jüri Üyesi, Kocaeli Üniversitesi

Doç. Dr. Ali ÇALHAN
Jüri Üyesi, Düzce Üniversitesi

Tezin Savunulduğu Tarih: 22.03.2019

ÖNSÖZ VE TEŞEKKÜR

Bu çalışmada, kuantum haberleşme kanalında güvenli anahtar dağıtımı için yeni bir hata sezme ve düzeltme (bilgi uzlaştırma) protokolü önerilmektedir.

Tez çalışmasının her aşamasında gerekli olan ilgiyi gösteren, desteği ve yardımlarını esirgemeyen ve önümüze çıkan sorunların çözümünde bana yön gösteren değerli hocam ve danışmanım sayın Dr. Öğr. Üyesi Sıtkı ÖZTÜRK'e sonsuz teşekkürlerimi sunarım.

Tez çalışmasının tüm aşamalarında bilgisine, tavsiyelerine ve desteğine başvurduğum, kuantum haberleşme konusunda teorik anlamda önemli bir katkı sağlayan, akademik yayınlarımın tümünde yer alan değerli ağabeyim sayın Dr. Mustafa TOYRAN'a çok kıymetli emekleri ve katkıları için teşekkür ediyorum.

Doktora öğrenimim sürecinde benden hiçbir desteği esirgemeyen, her koşulda bana destek olan, bu uzun süreçte her an yanımda olan değerli eşim Serap TOYRAN'a yardımları ve gösterdiği sabır için şükranlarımı sunuyorum. Benzer şekilde, kendilerine ayırmam gereken zamanın bir kısmını doktora öğrenimime ayırdığım için çocuklarım Serra Ela TOYRAN ve Hasan Ali TOYRAN'a da fedakarlıklarından ötürü şükranlarımı sunuyorum.

Tez çalışmasında değerli yorumlarıyla problemlerime ışık tuttukları ve önerdikleri tavsiyelerle bana yeni yollar gösteren değerli hocalarım sayın Prof. Dr. Cabir VURAL'a ve sayın Dr. Öğr. Üyesi Sultan Aldırmaz ÇOLAK'a teşekkürlerimi sunarım.

Doktora öğrenimim boyunca gerekli kolaylıkları sağlayan, zorlandığım dönemlerde beni cesaretlendiren ve tez çalışmamda gösterdikleri destekten ötürü TÜBİTAK – BİLGEM – UEKAE'deki tüm çalışma arkadaşlarıma teşekkürlerimi sunarım. Doktora öğrenimim ve tez çalışması boyunca sık sık görüşlerine başvurduğum ve derslerine katıldığım Elektronik ve Haberleşme Mühendisliği Bölümü hocalarıma teşekkür ediyorum.

Fen Bilimleri Enstitüsü'ndeki kıymetli çalışanlara göstermiş oldukları ilgi, alaka ve yardımlarından ötürü teşekkür ediyorum. Tüm yaşantım boyunca her koşulda yanımda olan, karşılaştığım bütün zorluklarda desteklerini her zaman yanımda hissettiğim ailemin tüm fertlerine ayrı ayrı teşekkür ve şükranlarımı sunarım.

Nisan - 2019 Metin TOYRAN

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

k_1	:	Birinci raunt blok uzunluğu
k_i	:	i. raunt blok uzunluğu
$p_X(x)$	:	Olasılık kütle fonksiyonu
$p(x)$	:	Olasılık kütle fonksiyonunun kısaltılmış gösterimi
$H(X)$	:	X raslantı değişkeninin entropisi
$h(p)$	:	İkili entropi
$H(X\|Y)$	:	X ve Y raslantı değişkenleri arasındaki koşullu entropi
$I(X;Y)$	:	X ve Y raslantı değişkenleri arasındaki karşılıklı bilgi
$p(x, y)$	:	x ve y raslantı değişkenlerinin birleşik olasılık fonksiyonu
$p(x\|y)$	:	x ve y raslantı değişkenleri arasındaki koşullu olasılık fonksiyonu
C	:	Kanal kapasitesi
E	:	Bilgi uzlaştırma için değiş tokuş edilen fazlalık bit sayısı
İ	:	Mesajın değerler aldığı ikili bit dizisi kümesi
M	:	Mesaja ait bit sayısı
N	:	Göndericiden alıcıya gönderilen toplam bit sayısı
R	:	Bilgi oranı
ε	:	Kanal hata olasılığı
İSK(ε)	:	ε kanal geçiş olasılıklı ikili simetrik kanal
$C_{\text{İSK}}(\varepsilon)$	:	İkili simetrik kanalın kanal kapasitesi
μ	:	Verimlilik sembolü
β	:	Verimlilik sembolü
η	:	Verimlilik sembolü
FER	:	Frame error rate (Mesaj hata olasılığı)
BER	:	Bir error rate (Bit hata olasılığı)
μ_{FER}	:	Hatalardan arındırılmış μ verimlilik ifadesi
β_{FER}	:	Hatalardan arındırılmış β verimlilik ifadesi
η_{FER}	:	Hatalardan arındırılmış η verimlilik ifadesi
BU_H	:	Hatalı sonuçlanan deneme sayısı
BU_T	:	Toplam deneme sayısı
B_H	:	Hatası düzeltilemeyen bit sayısı
B_T	:	Toplam bit sayısı
E_i	:	Protokol boyunca sızan toplam bilgi miktarı
T_i	:	Geriye iz sürme adımlarında sızan toplam bilgi miktarı

Kısaltmalar

ARQ	:	Automatic Repeat reQuest (Otomatik Tekrar İsteği)
AES	:	Advanced Encryption Standard (Gelişmiş Şifreleme Standardı)
BCH	:	Bose, Chaudhuri and Hocquenghem (Bose, Chaudhuri ve Hocquenghem)
BER	:	Bit Error Rate (Bit Hata Olasılığı)

BICONF	:	Binary Confirmation (İkili Onay)
BU	:	Bilgi Uzlaştırma
EDC	:	Error Detection and Correction (Hata Sezme Ve Düzeltme)
FEC	:	Forward Error Correction (İleri Yönlü Hata Düzeltme)
FER	:	Frame Error Rate (Mesaj Hata Olasılığı)
GAU	:	Gizli Anahtar Uzlaştırma
HSD	:	Hata Sezme ve Düzeltme
IR	:	Information Reconciliation (Bilgi Uzlaştırma)
İSK	:	İkili Simetrik Kanal
KAD	:	Kuantum Anahtar Dağıtımı
LDPC	:	Low Density Parity Check (Düşük Yoğunluklu Parite Kontrolü)
PBB	:	Paritesi Bilinen Bloklar
QKD	:	Quantum Key Distrubiton (Kuantum Anahtar Dağıtımı)
SKR	:	Secret Key Reconciliation (Gizli Anahtar Uzlaştırma)
TBB	:	Tamamen Bilinen Bitler
XOR	:	Exclusive Or (Özel Veya)

KUANTUM ANAHTAR DAĞITIM SİSTEMLERİNDE VERİMLİ BİLGİ UZLAŞTIRMA İÇİN OPTİMUM CASCADE PROTOKOLÜ

ÖZET

CASCADE protokolü ilk defa Kuantum Anahtar Dağıtım (KAD) sistemlerinde kullanılmak üzere önerilmiş olan bir Hata Sezme ve Düzeltme (HSD) tekniğidir. Bu protokolde HSD işlemi ilave bilgiler gönderilerek yapılmaktadır. Protokol ham mesajın gürültülü bir kuantum kanaldan gönderilmesiyle başlar. İlave bilgiler ise gürültüsüz ve kimlik doğrulamalı klasik bir kanaldan gönderilmektedir.

KAD'da bu şekilde işletilen HSD yöntemine Bilgi Uzlaştırma (BU) ya da Gizli Anahtar Uzlaştırma (GAU) adı verilir. KAD'da kullanılan bir BU protokolü için, performans ölçütlerinden bir tanesi başarılı bir HSD için gönderilmesi gereken ilave bilgi miktarını temsil eden verimlilik kavramıdır. Bu ilave bilgiler açık bir kanaldan gönderildiği için herkes kolayca ele geçirebilir. Gizli anahtar gönderici ve alıcı dışındaki diğer tüm şahıslardan gizli tutulmalıdır. Gönderilen ilave bilgiler gizli anahtarın içeriği ile ilgili bilgi açığa çıkaracağı için olabildiğince az sayıda ilave bilgi göndermek gerekmektedir. Bu nedenle, mümkün olan en az bilgi gönderimi yapan BU tekniklerine ihtiyaç duyulmaktadır.

Söz konusu protokol kapsamlı bir şekilde analiz edildiğinde geliştirmeye açık noktalar bulunmaktadır. Bunlardan en önemlileri: Tamamen Bilinen Bitler (TBB) ve Paritesi Bilinen Bloklar (PBB)'dir. Bu çalışmada, geliştirmeye açık bu noktalar kullanılarak CASCADE protokolünün daha verimli gerçeklemeleri sunulmaktadır. Yapmış olduğumuz deneyler sonucunda, sunduğumuz Optimum CASCADE protokolünün literatürdeki bütün CASCADE protokollerinden ve diğer birçok BU tekniğinden daha iyi sonuçlar verdiği görülmüştür. Ayrıca, iyileştirmelerin protokolün hızında ne gibi bir etkiye sebep olduğu da incelenmiştir.

Anahtar kelimeler: Bilgi Uzlaştırma (BU), CASCADE Protokolü, Gizli Anahtar Uzlaştırma (GAU), Kuantum Anahtar Dağıtımı (KAD).

OPTIMIZED CASCADE PROTOCOL FOR EFFICIENT INFORMATION RECONCILIATION IN QUANTUM KEY DISTRIBUTION SYSTEMS

ABSTRACT

CASCADE protocol is an Error Detection and Correction (EDC) method proposed firstly for use in Quantum Key Distribution (QKD) systems. It is used to detect and correct all the errors in the keys transmitted over a noisy quantum channel. In CASCADE, this is done by sending some redundant information to the receiver. However, this extra information is sent over another noiseless classical channel after the quantum transmission is completely finished.

In QKD literature, this EDC process is also called as Information Reconciliation (IR) or Secret Key Reconciliation (SKR). For an IR protocol in QKD, one of the main performance measures is efficiency which depends on the amount of the redundant information sent to make the EDC possible. Since this extra information is transmitted over public channels, everyone can get it easily. Because this can damage the secrecy of the keys that must be kept secret from the third parties, more efficient, that is revealing less information about the keys, IR methods are needed.

In this work, we present more efficient implementations of CASCADE protocol, using some inherent information already available in the protocol, exactly known bits and already known parities. Our experiments have shown that our presented protocols are of higher efficiency than both all the previous CASCADE versions and several other more recently proposed IR methods. The effect of each efficiency improvement on the protocol throughput is also analyzed in this work.

Keywords: Information Reconciliation (IR), CASCADE Protocol, Secret Key Reconciliation (SKR), Quantum Key Distribution (QKD).

GİRİŞ

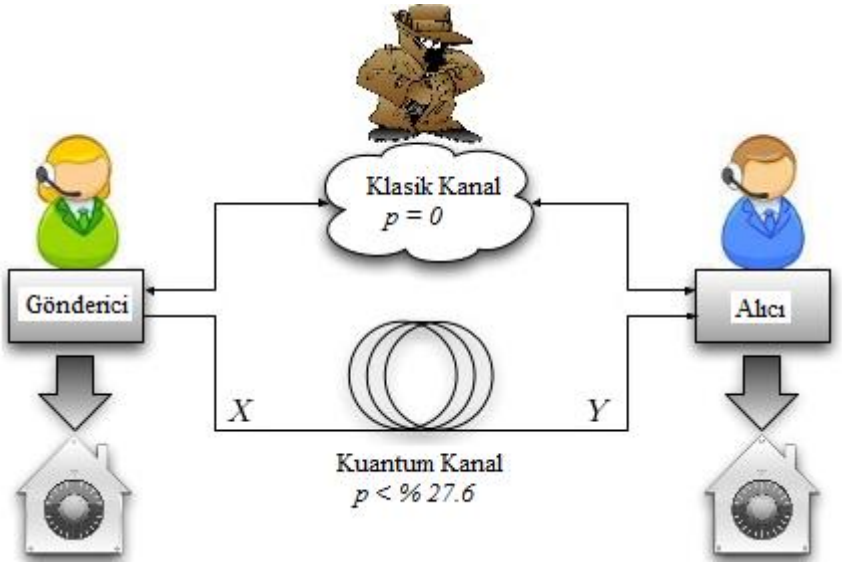
Bir mesajın gönderici ve alıcı arasında iletiminde iki temel problem vardır. Bunlar;

1. Kanalda oluşacak gürültüye rağmen mesajın alıcıya doğru iletilmesi,
2. Gönderici ve alıcı haricinde üçüncü kişilerin mesajı çözememesi veya mesajın gönderici ve alıcıya özel/gizli olması.

Kanal gürültüsünden kaynaklanan hataları gidermek amacıyla gönderici ve alıcı arasında mesaj iletimine ek olarak ekstra bilgi alış verişi yapılmaktadır. Bu maksatla, gönderici tarafından mesaja ilave bilgiler eklenebilir ya da mesaj iletimi bittikten sonra uygun hata sezme ve düzeltme teknikleri kullanılabilir. Gizlilik için ise, mesaj belirli bir yönteme göre şifrelenerek iletilir. Mesajı şifrelemek ya da şifreli mesajdan orijinal mesajı elde etmek maksadıyla kullanılan bilgiye anahtar adı verilmektedir. Şifreleme kuralları üzerinde çalışan bilim dalına da kriptoloji denir. Kriptolojide, gizli anahtarın üretilmesi, bu anahtarın gönderici ve/veya alıcıya güvenli bir şekilde dağıtılması sağlanmalıdır. Bu gizli anahtarın, hattı dinleyen ve hattaki haberleşmeyi çözmeye çalışan saldırganlardan korunması da önemli bir kriterdir.

Anahtar dağıtımı, kriptolojideki önemli problemlerden bir tanesidir [1]. Çözüm olarak geliştirilen çeşitli anahtar dağıtım yöntemleri birbirinden fiziksel olarak çok uzakta olan iki kişinin güvenli bir şekilde ortak bir gizli anahtar üzerinde anlaşabilmesine olanak verir. Şifrelemede ve şifre çözmede bu anahtar kullanılmaktadır [2, 3].

Kuantum Anahtar Dağıtım (KAD) protokollerinde, gönderici gizli anahtarı alıcıya güvenliği tamamen kuantum mekaniğinin özellikleri ile garanti edilen bir iletişimle iletir [4]. KAD protokollerinde saldırganın hesaplama gücü üzerine (teknoloji, zeka, vd.) herhangi bir kısıtlama getirilmez ve anahtarın iletimi esnasında iletişimi dinleyenlerin varlığını tespit edebilme yetenekleriyle de eşsizdirler [5].



Şekil 1.1. Kuantum anahtar dağıtımının (KAD) temel bileşenleri

Bir KAD protokolü, özetle, Şekil 1.1'de gösterilen en temel bileşenlerden oluşur [6]:

- Gönderici, alıcı ve hattı dinleyen saldırgan,
- Bir güvenli kuantum kanal (fiber ya da açık hava),
- Bir kimlik doğrulamalı açık klasik kanal.

Gönderici, gizli anahtarı kuantum sinyaller halinde kuantum kanal aracılığıyla alıcıya gönderir. Hattı dinleyen saldırgan, bu iletişimi ve sonrasındaki tüm iletişimleri dinlemeye çalışarak gizli anahtarı ele geçirmeyi ister. Kuantum mekaniğinin temel yasalarına göre, kuantum kanal üzerindeki kuantum iletişimi dinlemeye çalışan saldırgan tespit edilebilir hatalara neden olur. Çevresel etkilerden, ekipmanlardaki kusurlardan vb. kaynaklanan tüm diğer hatalar da yine saldırgana mal edilir. Eğer hatalar tanımlı bir eşikten daha aşağıda ise gizli anahtar güvenli bir şekilde elde edilmiş olur [7].

Hatalar nedeniyle kuantum iletişimin sonucunda göndericideki ve alıcıdaki anahtarlar farklı olur ve bu haliyle kullanışsızdır. Ortak anahtarın elde edilebilmesi amacıyla, daha sonra, alıcı tarafındaki anahtarın tüm hatalarının bulunup düzeltilmesine KAD literatüründe (gizli anahtar ya da bilgi) uzlaştırma denilmektedir. Bu konu, KAD'ın günümüzde üzerinde en çok çalışılan, darboğaz niteliğindeki, önemli problemlerinden bir tanesidir. Ayrıca bu konu, aslında haberleşme teorisinde bir tür hata sezme ve düzeltme uygulamasıdır [20].

KAD'da gizli anahtarın alıcıya iletilmesi gürültülü kuantum kanal üzerinden yapılır ve kuantum olan kısım da esas olarak sadece burasıdır. Bundan sonraki tüm aşamalar ise tamamen klasiktir ve bu aşamalarda gerekli iletişimlerin tümü klasik kanal üzerinden gerçekleştirilir.

Bilgi Uzlaştırma (BU), KAD'da tamamen klasik olan, herhangi bir kuantum mekaniği/fiziği bilgisi gerektirmeyen, aşamalardan bir tanesidir. BU adımına gelindiğinde karşılaşılan durum, özetle, şudur; Şekil 1.1'de de gösterildiği gibi göndericide N uzunluklu bir rasgele bit dizisi X (orijinali), alıcıda N uzunluklu bir rasgele bit dizisi Y (bozulmuş hali) vardır ve aralarındaki hata olasılığı p kadardır, p < % 27,6 [8]. Bu aşamada, gereken tüm iletişimler gürültüsüz bir klasik kanal üzerinden, örn. internet, yapılmak suretiyle Y = X yapılmaya, alıcı tarafındaki tüm hatalar bulunup düzeltilmeye çalışılır. Burada, dikkat çekilebilecek önemli bazı noktalar ise şunlardır:

- BU'nun başarıyla sonuçlanabilmesi için alıcı tarafındaki tüm hataların sezilmesi ve düzeltilmesi gerekir,
- Hata sezme ve düzeltme gürültülü bir (kuantum) kanal yerine kanal hata olasılığı sıfır olan bir klasik kanal üzerinden yapılmaktadır. Böylece, hatalar ve gönderilen ilave bilginin de daha az olması nedeniyle hata sezme ve düzeltme de çok daha kolay, hızlı, ucuz ve güvenli olacaktır. Bundan ötürü, böylesi bir uygulama daha tercih edilebilir olacaktır,
- Klasik kanal aynı zamanda kimlik doğrulamalı olup üzerindeki trafik (herkesçe ve kolayca) sadece pasif olarak dinlenebilir; diğer bir deyişle değiştirme yapılamaz. Bununla birlikte, hattı dinleyen saldırganın gizli anahtarı asla elde edememesi için hata sezme ve düzeltme amacıyla gönderilen ilave bilginin de mümkün olduğu kadar az olması gerekir [9]. Bu tez çalışmasında, değiş tokuş edilen ilave bilginin minimuma indirilmesi için farklı yöntemler önerilmiştir. Bu konu Bölüm 4'te kapsamlı bir şekilde ele alınmıştır.

KAD'da hatalar sadece kuantum kanaldan kaynaklanır (saldırgan vb. gibi etkilerden kaynaklanan hatalar da kuantum kanala mal edilir). Bu nedenle, kuantum iletim sonunda alıcıdaki bit dizisi göndericidekinin p < % 27,6 kadar bozulmuş halidir [8]. Bu hata daha sonra tamamen klasik kanal üzerinden gerçekleştirilen ilave iletişimlerle düzeltilmeye çalışılır. Klasik kanal ise halihazırda güçlü hata sezme ve düzeltme

teknikleri ile korunmakta olan modern bir iletişim kanalı olduğundan hata olasılığı p = 0 kabul edilir ya da ihmal edilebilecek kadar küçüktür.

Bu tez çalışmasında, KAD'daki BU problemi için CASCADE tabanlı teknikler üzerinde çalışılmıştır. Birinci bölümde, KAD'da BU ve bu yöntemin başarılı bir şekilde gerçekleştirilmesi için yapılan gelişmeler anlatılmıştır. Öncelikle problemin ne olduğu tariflenmiş, ardından problemin literatürde hangi açılardan ele alındığı ve ne gibi katkılar sunulduğu incelenmiştir. Son olarak, bu çalışmada probleme getirilen yenilikler/katkılar açıklanmıştır.

İkinci bölümde, ayrık olasılık teorisi, entropi, temel haberleşme teorisi, kuantum mekaniği ve kriptografi ile ilgili temel bilgiler verilmiştir. Yine ikinci bölümde, literatürde farklı çalışmalarda kullanılan ve BU'da verimlilik performansı için öne çıkan bazı terimler ve kavramlar açıklanmıştır.

Üçüncü bölümde, KAD yönteminin nasıl gerçekleştirildiği konusunda temel bilgiler özet halinde sunulmuştur. Bu bağlamda, fotonların polarize edilmesi, fotonlar ile bilgi taşıma, kuantum kopyalanamamazlık prensibi, hattı dinleyen saldırganların tespiti, gizli anahtar dağıtımı ve pratik kullanımlar ile ilgili bilgi verilmiştir.

Dördüncü bölümde, CASCADE protokolü ve literatürdeki türevleri incelenmiştir. Literatürde önerilen en popüler olarak kabul görmüş bazı CASCADE yöntemleri bu çalışmada gerçeklenmiş ve her biri için kapsamlı analizler yapılarak sonuçları da sunulmuştur. Ayrıca, CASCADE protokolünün üzerine yapılan ve ilk defa bu tez çalışmasında önerilen iyileştirmeler anlatılmıştır. Her bir iyileştirmenin verimlilik performansına olan etkisi kapsamlı deneylerle sunulmuştur ve her bir iyileştirmenin etkisi karşılaştırmalı olarak incelenmiştir. Yine bu bölümde, önerilen iyileştirmelerin literatürdeki diğer çalışmalardan farkları da vurgulanmıştır.

Beşinci bölümde, bu çalışmada önerilen yeni yöntem kapsamlı bir şekilde analiz edilmiş, deneysel sonuçlar tablo ve grafiklerle sunulmuştur. Yine bu bölümde, önerilen yeni protokol, CASCADE olan ve olmayan birçok BU tekniğiyle karşılaştırılmış ve sonuçları sunulmuştur. Bu bölümde, protokol öncelikle kapsamlı bir şekilde verimlilik performansı açısından ele alınmıştır. Öncelikle, her bir iyileştirmenin verimlilik performansına katkıları ele alınmıştır. Daha sonra ise, tüm iyileştirmeleri içeren

CASCADE protokolünün verimlilik performansı kapsamlı deneylerle incelenmiş ve sonuçları sunulmuştur. Ayrıca, bu çalışmada önerilen CASCADE protokolü literatürdeki popüler HSD teknikleriyle karşılaştırılmış ve incelenen her bir tekniğin en başarılı verimlilik sonuçları da sunulmuştur. Son olarak, protokol hız açısından da incelenmiştir. Bu çalışma kapsamında, protokole verimliliği arttırmak maksadıyla eklenen her bir iyileştirmenin protokolün hız performansına olan etkileri de analiz edilmiş ve hız performansları da sunulmuştur.

Altıncı bölümde ise, bu çalışma kapsamında ortaya çıkan sonuçlar özetlenmiş, önemli görülen ve geliştirmeye açık olan noktalar vurgulanmıştır.

1. CASCADE PROTOKOLÜNÜN GELİŞİMİ

1.1. Problemin Tanımı

KAD'da temel amaç kriptografik bir gizli anahtarın göndericiden alıcıya üçüncü kişiler tarafından hiçbir şekilde ele geçirilemeden iletiminin sağlanmasıdır. İlk KAD protokolü Charles H. Bennett ve Gilles Brassard tarafından 1984 yılında ortaya atılmıştır [11]. Yazarlar bu protokolü geliştirirken Stephen J. Wiesner'in 1960'ların sonlarında taklit edilemez para yapımı için ortaya attığı yeni ve özgün fikirlerden faydalanmışlardır [10]. İlk defa ortaya atılan bu protokole yazarların soyisimlerinin ilk harfleri ve çalışmanın yayın yılı kullanılarak BB84 adı verilmiştir. Bu protokolde gizli anahtardaki her bir bit alıcı tarafa foton iletimine imkan veren bir kanal üzerinden uygun şekilde polarize edilmiş tek bir foton ile güvenli bir şekilde gönderildiği belirtilmiştir.

Gönderilen bitin güvenliği taşıyıcı olarak kullanılan fotonun doğasından gelmektedir. Fotonlarla bilginin taşındığı böyle bir kanala kuantum kanal adı verilir ve fiberoptik bir kablo ya da açık hava olabilir. Böyle bir kanaldan taşınan bitlere klasik fizik kuralları ile hiçbir şekilde erişilemez. Literatürde KAD'da kullanılan kuantum kanal, ayrık ve sürekli olarak modellenebilmektedir. Bu çalışmada, gizli anahtarın ikili bitlerden, 0 ve 1, oluştuğu kabul edilmiştir ve BU için ayrık zamanlı bir kuantum kanal olan İkili Simetrik Kanal (İSK) modeli incelenmiştir.

Kuantum kanal üzerinde bir bit için bir foton kullanan bu güvenli haberleşme yöntemine kuantum iletim adı verilir. Bu iletişim gürültüye karşı çok hassastır. Kanal üzerindeki tüm çevresel faktörler (güneşten veya diğer foton kaynaklarından gelen fotonlar), araya giren saldırganlardan kaynaklı etkiler, kuantum kanaldan kaynaklı kusurlar ve sistemdeki elektriksel ve/veya optik bileşenlerden kaynaklanan diğer tüm etkiler (sıcaklık, parazit vb.) kuantum kanal üzerinden iletilen fotonu bozabilir. Bu nedenle, alıcı tarafa ulaşan ikili mesaj dizisinde hatalı bitler oluşabilir. Genellikle, kuantum kanal gürültülü bir kanal olarak kabul edilir ve alıcı tarafa ulaşan bit dizisindeki hataların bu etkiye bağlı olarak oluştuğu kabul edilir.

Kuantum kanalın kanal hata olasılığı arada saldırgan ve benzeri gibi etkiler yok iken % 1-4 arasında değişmektedir. Ancak, arada saldırgan ve benzeri gibi diğer bozucu etkilerin bulunması durumunda dahi, % 30 mertebelerinde hata olasılıklarına kadar bile, güvenli anahtar dağıtımı yapılabilmektedir [26]. Protokolün çalıştırılması sonucunda anahtar dağıtımının başarılı olabilmesi için gönderici ve alıcı tarafındaki bit dizilerinin tamamen aynı olması gerekmektedir. Ayrıca, gönderici ve alıcıdaki bit dizisinin kanallarda açık olarak yapılan haberleşmeleri dinleyebilen saldırganların elde ettiği bit dizisinden de farklı olması gerekmektedir. Bu nedenle, gürültülü kuantum kanal üzerinden yapılan kuantum gizli anahtar iletimi aşamasından sonra, alıcı tarafında hatalı bitler içeren bit dizisindeki hataları gideren güvenilir bir mekanizmaya ihtiyaç vardır. Diğer bir deyişle, alıcı tarafındaki bit dizisindeki bütün hataları sezen ve düzelten, böylece gönderici ve alıcıdaki bit dizilerini eşit hale getiren bir mekanizmaya ihtiyaç vardır. KAD'da, bu mekanizmaya BU ya da Gizli Anahtar Uzlaştırma (GAU) ya da sadece uzlaştırma adı verilmiştir. Bu mekanizma ilk olarak KAD'ın mucitleri ve onların öğrencileri tarafından tartışılmıştır [12-14]. BU protokollerinin temel olarak iki performans kısıtı vardır. Bunlardan birisi verimliliktir. Verimlilik, BU işlemi için ne kadar fazlalık bilgi değiş tokuş edildiğinin bir ölçütüdür. Hız ise bir diğer performans kısıtı olup BU protokolünün saniyede kaç tane bit düzeltilebildiğinin bir ölçütüdür. Literatürde farklı BU protokolleri bu ölçütleri ele almış ve bunları iyileştirmeye odaklanmıştır.

Hata kodlama teorisinde Hata Sezme ve Düzeltme (HSD) işlemi alıcıya mesajla ilişkili bazı ilave bilgilerin gönderilmesiyle sağlanır [15]. Bu ilave bilgiler alıcı tarafta mesajdaki hataların sezilmesi ve düzeltilmesini mümkün kılar. Klasik HSD tekniklerinde bu ilave bilgiler genellikle mesajla birlikte gönderilir. Ancak, KAD sistemlerinde, önce gürültülü bir kuantum kanal üzerinden bir kuantum gizli anahtar iletimi yapılır ve bu iletim sonucunda muhtemelen gizli anahtardaki bazı bitler bozulur. Bu işlem bittikten sonra ise, gönderici HSD için alıcının talep ettiği ilave bilgileri klasik kanal üzerinden alıcıya gönderir.

KAD sistemlerinde, ilave bilgilerin gönderimi başka bir kanaldan yapılır. Bu kanal klasik, kimlik doğrulamalı, modern ve güçlü HSD teknikleriyle (ARQ ve FEC gibi) donatılmış gürültüsüz bir kanaldır. Örneğin, böyle bir klasik kanal internet ve radyo yayını şeklinde olabilir. Böylece, ilave bilgiler, kuantum kanal üzerinden yapılan

gürültülü iletimin aksine, alıcı tarafına kesin olarak hatasız ulaşır [80]. Ancak, teorik olarak bütün klasik kanallar araya giren saldırganlar tarafından kolayca dinlenebilir. Diğer bir deyişle, klasik kanal üzerinden taşınan, gizli anahtarla ilgili bilgi ihtiva eden ve başka hiçbir kimsenin eline geçmemesi gereken bu ilave bilgileri araya giren saldırgan pasif olarak dinleyebilir. Bu nedenle, KAD literatüründe araya giren saldırgan da bu bilgileri bildiği için ilave bilgilere aynı zamanda açığa çıkan (sızan) bilgi de denilmektedir. Buradan kolayca şu sonuca varılabilir ki: ne kadar fazla ilave bilgi gönderilirse, anahtarın gizliliği o kadar zaafiyete uğrar.

HSD işleminde gönderilen ilave bilgi miktarı anahtarın gizliliğini zaafiyete uğrattığı için, KAD sistemlerinde BU'yu mümkün olan en az sayıda ilave bilgi ile sonuçlandırabilen, diğer bir deyişle mümkün olduğunca az sayıda bilgi açığa çıkaran, BU tekniklerine ihtiyaç vardır. KAD sistemlerinde, açığa çıkarılan bilgi miktarı BU protokollerinin verimlilik performansının bir ölçütüdür. Yani, ne kadar fazla bilgi açığa çıkıyorsa, BU protokolü o kadar verimsizdir. Diğer bir deyişle, ne kadar az bilgi açığa çıkıyorsa, BU protokolü o kadar verimlidir.

1.2. Literatür Taraması

KAD ile gizli anahtarın dağıtılması sırasında karşılaşılan BU problemi ilk olarak BB84 protokolünün yazarları tarafından çalışılmıştır [13, 14, 16]. Problemin ilk çözümü olan BBBSS protokolünde, yazarlar bu problemi çözmek için bir dizi parite değiş tokuşunu tarif eden bir teknik önermiştir [14]. Bu yöntemde, gizli anahtar üzerinde blok parite hesaplamaları yapılmıştır ve paritelerin uyuşmadığı durumda ilgili bloğun hatalı bir bit içerdiği anlaşılmıştır. Bu durumda, ilgili blok üzerinde ikili arama yapılarak hatalı bitin bulunması ve alıcı tarafta düzeltilmesi sağlanmıştır. Burada dikkat edilmesi gereken nokta, parite kontrolünün doğasından dolayı, her parite kontrolü işleminde sadece tek sayıda hatalı bitin sezilebiliyor olmasıdır ve ikili arama sırasında ise sadece bir tanesinin düzeltilebiliyor olmasıdır. Bu bahsedilen adımlar iteratif olarak tekrar edilmiş ve parite kontrolü tabanlı tekniklerin daha iyi çalışabilmesi için her iterasyon öncesi gizli anahtardaki bitler karıştırılarak hataların gizli anahtar içinde rasgele olarak dağılması sağlanmıştır.

Daha sonra, [16]'daki çalışmanın yazarları düzeltilen her hatalı bitin daha önceki iterasyonlarında geçtiği yeri bulup orada kullanılabileceğini farketmişlerdir. Diğer bir

deyişle, ikinci ve daha sonraki iterasyonlarda düzeltilen her yeni bir hatanın önceki iterasyonlardaki yeri bulunur ve orada da yeniden hata düzeltme işlemleri başlatılır. Protokolün birbirini tetikleyen hata düzeltme özelliğinden ötürü bu yeni yönteme CASCADE ismi verilmiştir. Protokol çok basit tasarlanmıştır ve sadece bir parametre ile ilklendirilebilmektedir. Bu parametre blok uzunluğu şeklinde isimlendirilir ve bir blokta bulunan bitlerin sayısını temsil etmektedir. İlk iterasyondaki blok uzunluğunu gösteren k_1'in değeri ε ile gösterilen kuantum kanalın bit hata olasılığı kullanılarak hesaplanır. Değeri deneysel olarak $k_1 \cong 0{,}73/\varepsilon$ olarak seçilir. İlk iterasyondan sonraki iterasyonlarda blok uzunluğu iki katına çıkar. Örneğin, *i*. iterasyondaki blok uzunluğu $k_i = 2k_{i-1}$ şeklinde hesaplanır.

BBBSS ve [16]'daki orijinal CASCADE protokolleri yayınlandıktan sonra literatürde bu protokollerde iyileştirmelerin yapıldığı yeni teknikler önerilmiştir [17-27]. Ancak, tüm bu teknikler çok fazla interaktiftir ve BU için gönderici ve alıcı arasında çok miktarda veri alışverişi gerektirmektedir. Yine de, CASCADE protokolünün fazlasıyla interaktif olmasına rağmen, basit olması ve diğer protokollerle kıyaslandığında daha iyi verimlilik değerleri üretmesi nedeniyle KAD'da BU için halen en yaygın olarak kullanılan protokollerden biri olmaktadır. Literatürde, CASCADE protokolünün değişik türevleri de önerilmiştir. Örneğin, 2000 yılında Sugimoto ve arkadaşları [22] yeni bir CASCADE protokolü önermişlerdir. Bu çalışmada, ilk iki roundda orijinal CASCADE protokolü çalıştırılmıştır. Ancak, iki roundda tüm hatalar düzeltilemediği için yöntem burada sonlandırılmamıştır. Bu aşamadan sonra BICONF adını verdikleri yeni bir yöntemle hataların tümü çözülmeye çalışılmıştır. Yazarlar bu yaklaşımla orijinal CASCADE'den daha verimli sonuçlar elde etmişlerdir, ancak teorik limite halen çok uzak kalmışlardır. Daha sonra 2008 yılında, Yan ve arkadaşları orijinal CASCADE protokolüne bağlı kalarak yeni bir yöntem önermişlerdir [23]. Bu yöntem literatürde yeni bir bakış açısı açmış olup, daha sonraki çalışmalarda da referans alınmıştır. Bu tez çalışmasında önerilen yeni CASCADE yöntemi de kendisine [23]'te önerilen versiyonu referans almıştır ve yeniliklerini bu versiyon üzerine yapmıştır [9]. Bu tez çalışması boyunca bu protokole referans protokol adı verilmiştir. Yazarlar orijinal CASCADE'in hafızalı yapısını genişletmişlerdir. Orijinal CASCADE'de her roundun başında oluşan bloklar hafızaya alınmaktadır. Yazarlar bu bloklara ek olarak BINARY işleminde oluşan küçük blokları da hafızaya almayı denemişlerdir. Bu da

hatanın arandığı blok uzunluğunu azalttığı için gönderici ve alıcı arasında değiş tokuş edilen parite bilgisinin miktarını azaltmıştır. Böylece verimlilik anlamında hem orijinal CASCADE hem de [22] çalışmasından daha iyi sonuçlar elde edilmiştir. Ancak, teorik limite (Shannon limiti) halen ulaşılamamıştır [46]. Bu da protokolde halen verimsizliğe neden olan noktalar olduğunu göstermektedir. 2015 yılına gelindiğinde, Jesus ve arkadaşları literatürdeki CASCADE tekniklerini ve önerilen iyileştirmeleri incelemiş ve bu iyileştirmelerin etkilerini birçok açıdan analiz etmişlerdir [24]. Bunlardan başlıcaları şöyledir:

- CASCADE'de her iterasyonun başında kullanılan karıştırıcıların daha iyi seçilmesi,
- Her iterasyonda elde edilen tekil bloğun bir sonraki iterasyonda oluşan bloklardan çıkarılması,
- CASCADE protokolünde tanımlı olan blok uzunluklarının optimizasyonu,
- BINARY'de oluşan alt-blokların hata aramada kullanılması [24].

Yazarlar, yapılan iyileştirmelerin önemine vurgu yapmış, ancak protokolden en yüksek verimlilik performansını alabilmek için parametre seçiminin çok daha önemli olduğunu vurgulamışlardır. Hatta, CASCADE protokolünün sadece ilk round blok uzunluğunun yeterince iyi seçilmesiyle bile çok başarılı verimlilik sonuçları elde edebileceklerini ifade etmişlerdir. Çalışmalarındaki nihai ve en başarılı sonuçları [23]'te önerilen CASCADE protokolünü kendi önerdikleri parametre kümesi ile çalıştırarak vermişlerdir.

[24]'teki çalışmada, [23]'te önerilen CASCADE yöntemini kullanmış ve bu teknikle birlikte kullanmak üzere yeni bir parametre kümesi önermişlerdir. Yazarlar bu yeni parametre kümesi ile literatürdeki en iyi verimlilik sonuçlarını elde etmişlerdir ve teorik limite epey yaklaşmışlardır. Ancak, benzer şekilde, halen teorik limitle aralarında bir miktar fark da bulunmaktadır. Bu da, yine benzer şekilde, protokolde iyileştirilebilecek başka noktaların da varolduğuna dair fikir vermektedir. Bu tez çalışmasının amaçlarından biri de, literatürde şu ana kadar görülemeyen bu noktaların tespit edilmesidir. Bu bağlamda, bu tez çalışmasında, Tamamen Bilinen Bitler (TBB) ve Paritesi Bilinen Bloklar (PBB) iyileştirmeleri önerilmiştir ve bu iyileştirmeler CASCADE protokolüne uygulanarak literatürdeki en verimli CASCADE yöntemi elde edilmiştir [9]. Bu iyileştirmeler Bölüm 4'te açıklanmıştır. Bu tez çalışmasında, bu yeni yönteme optimum CASCADE protokolü adı verilmiştir. Literatürde, başka

araştırmacılar tarafından da CASCADE protokolünün verimlilik performansını arttırmak için birçok yeni yöntem yayınlanmıştır. Bunlardan bazıları protokolü değiştirmişlerdir [18, 19, 22]. Bazıları ise, protokolü değiştirmeden parametre kümesini değiştirerek ya da protokolün kendisini iyileştirerek optimize etme yoluna gitmişlerdir [17, 21, 23, 24].

KAD'da BU maksadıyla kullanılan diğer popüler çalışmalar LDPC (Düşük Yoğunluklu Parite Kontrolü, Low Density Parity Check) ve Polar kodlardır. 2003 yılının başlarında, ilk defa haberleşme sistemlerinde hata sezme ve düzeltme amacıyla parite tabanlı kontrollerin yapılabileceği fikri Los Alamos'taki DARPA (The Defense Advanced Research Projects Agency) grubundan Chip Elliot tarafından ortaya atılmıştır [28]. Ancak, bu grup bu iddia için o yıl herhangi bir analiz veya çalışma sunmamıştır. Bir yıl sonra, haberleşme sistemleri için LDPC yöntemini tanıtmışlardır [29, 30]. Bu çalışma, [24]'te anlatılan modern kodlama teorisinin KAD'daki BU problemine uygulandığı ilk pratik yöntemlerden biri olmuştur. Ancak, LDPC kodlarının bu alanda kullanımı için 2009 yılına kadar pek ilerleme sağlanamamıştır. Bu yıldan itibaren, farklı bilgi iletim hızları için özel kodların tanımlanabilmesi ile birlikte KAD'da yeniden kullanılmaya başlanmıştır [40-42, 46]. Polar kodlar ise, ikili hafızasız ve simetrik kanal üzerinden iletilen bilgiler için kullanılan doğrusal blok hata düzeltme kodlarının bir türüdür [34]. n bit sayısı olmak üzere ve $N = 2^n$ de iletilen bilginin uzunluğunu göstermek üzere, N değeri arttıkça polar kodların verimlilik performansı Shannon limitine yaklaşmaktadır. Literatürde, polar kodların pratik olarak gerçeklenmesi için gerekli bazı hususların, örneğin N'nin uzunluğu, polar kod çözücünün doğruluğu ve polar kodlar için kabul edilebilir performans ölçütlerinin ele alındığı çalışmalar da yapılmıştır [35, 36]. Polar kodlar KAD sistemlerinde BU maksadıyla da kullanılmaktadır [37, 38]. LDPC ve Polar kodları kullanan çalışmalarda CASCADE protokolünün fazlasıyla interaktif olması nedeniyle yüksek gecikmelere neden olduğu öne sürülmüştür. Ancak son yıllarda yapılan çalışmalar ile CASCADE protokolünün gerçek manada gecikme sorunu olmadığı gösterilmiştir [24, 26].

Ayrıca, literatürde BU için başka HSD stratejileri kullanan teknikler de mevcuttur. Bunlar [18, 27]'de anlatılan protokoller, BCH [39], Turbo [40] vb.'leridir.

1.3. Bu Çalışmanın Amacı

Mesaj alıcıya ulaştıktan sonra hata düzeltme amacıyla alıcı tarafından CASCADE protokolü çalıştırılır. CASCADE protokolünde, gönderici ve alıcı gizli anahtarı oluşturan bit dizisini öncelikle karıştırır ve bloklara böler. Her iki taraf da her blok için parite değiş tokuşu gerçekleştirir ve blokta parite uyuşmazlığı varsa hata düzeltme adımına geçerler. Bu adımda, benzer şekilde blok daha küçük bloklara bölünür ve yine her blok için parite kontrolü yapılır. Bu işlem, paritenin uyuşmadığı tüm bloklar üzerinde yinelemeli olarak devam ettirilir. Uzunluğu bir bit olan ve paritenin uyuşmadığı bit bulunduğunda işlem sona erdirilir ve hatalı olan bu bit düzeltilir. Burada dikkat edilmesi gereken nokta, her parite kontrolü işleminde sadece tek sayıda hatalı bitin sezilebiliyor olmasıdır ve ikili arama sırasında ise sadece bir tanesinin düzeltilebiliyor olmasıdır. Gönderici ve alıcı arasında gerçekleştirilen bu iletişimler modern, güçlü kimlik doğrulamalı ve gürültüsüz bir klasik kanal üzerinden gerçekleştirilir. Bu kanalın teorik olarak herkese açık olduğu ve araya giren saldırganların pasif olarak (mesajları okuyabilir ama değiştiremez) hattı dinleyebildiği kabul edilir. Bu teknik belirli bir sayıda tekrarlanır. Birbirini takip eden adımlarda gizli anahtara ait bit dizisi karıştırılır ve her adımda farklı parçalara da bölünebilir. Her adımda hata sezme ve düzeltme işlemleri için gönderici ve alıcı arasında fazlasıyla parite değiş tokuşu yapılmaktadır ve bu adım defalarca tekrar edilebilmektedir. Bu da değiş tokuş edilen parite bilgisinin çok fazla olabileceğini göstermektedir. Diğer bir deyişle, CASCADE protokolü fazlasıyla interaktif olan bir protokoldür. Bu da protokolün ağ hızlarından olumsuz yönde etkilenebileceği anlamına gelmektedir. Diğer bir deyişle, yoğun olarak gerçekleştirilen bu haberleşmeler CASCADE protokolünün hızını ve verimliliğini düşürmektedir [20, 33, 37, 41]. Literatürde KAD'da BU protokolü olarak kullanılan LDPC ve Polar kodlar gibi farklı protokoller de vardır. Bu protokoller ileri yönlü hata düzeltme teknikleri kullanırlar ve hata sezme ve düzeltme işlemi için CASCADE gibi interaktif haberleşmeler yapmazlar. Bu tekniklerde, gönderici mesajla birlikte HSD işlemi için gereken ilave bilgiyi de alıcıya iletişimin başında tek seferde gönderir ve sonrasında yeni bir haberleşme yapılmaz. Ancak, bu tekniklerde de CASCADE'e göre daha fazla hesaplama yapılır ve gönderilen ilave bilgi gerektiğinden fazla olabilir [9]. Bu da tekniklerin daha verimsiz ve güvensiz olmasına neden olur. Bu durum, Bölüm 5'te Tablo 5.4'te gösterilmiştir.

Literatürde KAD alanında yapılan araştırmalarda en çok ilgi çeken ve iyileştirilmesi konusunda en fazla ele alınan performans ölçütü verimliliktir [16, 20-33, 42, 43]. Ayrıca, pratik anlamda bir BU protokolü seçilirken her tekniğin haberleşme verimliliği/hızı ve işlemsel karmaşıklığı kriterleri dikkatli bir şekilde değerlendirilir [37, 42]. Örneğin, belirli bir V hızından düşük olmaması istenen bir BU protokolü için aranan en önemli performans ölçütü eldeki mevcutlar arasında en verimli olanı olacaktır. Eğer, seçilen BU protokolü V hızından yüksek ise, verimliliği daha da arttırmak adına aradaki hız farkı gözden çıkarılabilir. Diğer bir deyişle, hız V'ye kadar düşürülüp mümkün olduğu kadar verimlilik arttırılabilir. Eğer, BU protokolünün hızı kanaldan gelen veri hızından azsa, performansı belirlerken hız ve verimlilik birlikte ele alınır. Örneğin, gizli mesajın güvenliğini ihlal etmeyecek şekilde verimlilik performansı gerektiği ölçüde düşürülerek BU protokolünün hızının arttırılması sağlanabilir [37, 42].

Bu konu literatürde de incelenmiş, ancak ağ gecikmeleri ve işlemsel karmaşıklık konusunda henüz net bir görüş elde edilememiştir. Düşük gecikmeli ağlarda çalışan, işlemsel karmaşıklığı az ve yüksek ağ haberleşmesi gerektiren BU protokolleriyle işlemsel karmaşıklığı yüksek ancak çok fazla ağ haberleşmesi gerektirmeyen BU protokollerinin birbirlerine üstünlüğü henüz ispatlanamamıştır. Bu çalışmanın amaçlarından biri de [16, 20, 21]'de sunulan CASCADE gibi yoğun ağ haberleşmesi gerektiren protokollerin bu derecede interaktif olmayanlara göre verimlilik performansı açısından daha kötü olmadığını deneysel sonuçlarla destekleyerek göstermektir. Bu maksatla, CASCADE protokolü kapsamlı bir şekilde analiz edilecek, verimli bir şekilde gerçeklendiğinde KAD için önerilen ve [33, 37, 41, 44, 45]'te de sunulan en güncel BU protokollerinin tümünden daha başarılı verimlilik sonuçları verebileceği gösterilecektir.

Yukarıda bahsedilen bilgiler ışığında, bu tez çalışmasının temel amacı, KAD'da BU için kullanılan CASCADE protokolünün daha verimli bir gerçeklemesini önerebilmektir. Bu bağlamda, başta CASCADE olmak üzere literatürdeki tüm BU yöntemlerinden daha verimli bir CASCADE protokolünün tasarlanması hedeflenmiştir. Bu nedenle, öncelikle literatürdeki CASCADE protokolü üzerinde yapılan çalışmalar incelenmiştir ve protokolde zaman içerisinde iyileştirilen noktalar belirlenmiştir. Birçok çalışma eski yöntemleri iyileştirmiş ve verimlilik değerlerini

teorik limite biraz daha yaklaştırmıştır. Ancak, sunulan sonuçlara bakıldığında teorik limite halen ulaşılmadığı gözlemlenmiştir. Diğer bir deyişle, protokolde halen yapılabilecek iyileştirmeler bulunmaktadır. Bu tespitlerden yola çıkarak, orijinal CASCADE protokolü üzerinde kapsamlı analizler yapılmıştır ve sonucunda protokol içinde gizli olan ve akıllıca kullanıldığında verimlilik performans ölçütünün iyileştirilmesine fayda sunacak bazı içsel bilgiler bulunmuştur. Bu iyileştirmeler sırasıyla protokole uygulanmış ve her bir iyileştirmenin protokolün verimlilik performansı üzerindeki etkileri kapsamlı simülasyonlarla incelenmiştir. Ayrıca, tüm iyileştirmeleri de içeren optimum bir CASCADE protokolü önerilmiş ve bu protokolün de verimlilik değerleri ölçülmüştür. Bu çalışmada ayrıca, protokole verimliliği arttırmak maksadıyla yapılan iyileştirmelerin her birinin bir diğer performans ölçütü olan hız üzerindeki etkileri de ele alınmıştır. Her bir iyileştirmenin ve tüm iyileştirmeleri içeren optimum CASCADE versiyonunun protokolün hızında ne gibi etkilere sebep olduğu da kapsamlı analizlerle incelenmiş ve simülasyon sonuçları sunulmuştur.

2. TEMEL KAVRAMLAR

Bu bölümde bilgi kuramı kapsamında bahsi geçen temel kavramlardan bahsedilecektir. Bu konu ile ilgili detaylar Claude E. Shannon'ın 1940 yılında yayınladığı [46] ve [47] çalışmalarında incelenmiştir. Yazar bu çalışmalarında hem bilgi teorisini hem de gizli anahtarın ele geçirilmesi ile ilgili olarak bilgi teorisinde güvenlik hususlarını sunmuştur. Burada bahsi geçen teorik bilgiler hakkında daha detaylı bilgi almak için [48]'deki kaynağa da başvurulabilir.

2.1. Ayrık Olasılık Teorisi ve Entropi

X değişkeni, ayrık zamanlı bir raslantı değişkenini göstersin ve bu kümedeki her $x \in X$ değeri için bir $p_X(x)$ olasılık kütle fonksiyonu tanımlı olsun. Bu durumda, $\sum p_X(x) = 1, x \in X$ olmak üzere olasılık kütle fonksiyonu $p_X(x) = Pr(X = x)$ şeklinde ifade edilebilir. Gösterimde kolaylık olması açısından tezin kalan kısımlarında olasılık kütle fonksiyonu için $p_X(x)$ yerine $p(x)$ kullanılacaktır.

Entropi: X ayrık zamanlı bir raslantı değişkeni olsun. p(x) de X'in olasılık kütle fonksiyonunu temsil ediyor olsun. Bu durumda, X'in [46]'da anlatılan Shannon entropisi, ya da kısaca entropisi, şu şekilde ifade edilir,

$$H(X) = - \sum p(x) \log p(x), \quad x \in X \tag{2.1}$$

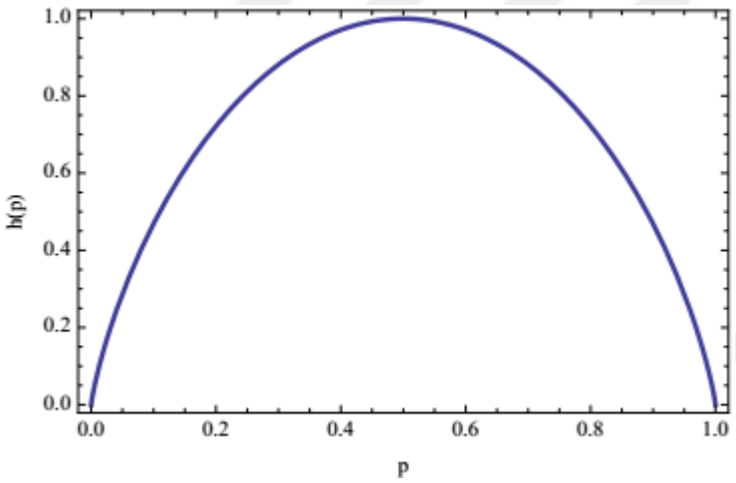
Entropi kavramı bir raslantı değişkeninin ortalama belirsizliği ile ilgili bilgi veren bir ölçüttür.

X raslantı değişkeni sadece iki farklı eleman içeren bir kümeden değer alıyorsa, örn. 0 ve 1, Shannon entropisi bit bazında ölçülür. Denklem (2.1)'de verilen entropi ifadesi $h(p)$ ile temsil edilir ve Denklem (2.2)'deki gibi gösterilir. Bu ifadeye ikili Shannon entropisi adı verilir [46].

$$h(p) = - p\log_2 p - (1 - p)\log_2 (1 - p) \quad (2.2)$$

Bu ifadede p, 0 ve 1'in olasılıklarını temsil etmektedir. $p = Pr(x = 0)$ ve bunun tümleyeni olan $(1 - p) = Pr(x = 1)$ olmak üzere $Pr(x = 0) + Pr(x = 1) = 1$ olacaktır.

Şekil 2.1'de p ile h(p) arasındaki ilişki gösterilmiştir. Burada dikkat edilmesi gereken bir husus; normalde $0\log_2 0$ matematiksel olarak tanımlı olmamasına rağmen geleneksel olarak $0\log_2 0$ sıfır olarak kabul edilmiştir. Ayrıca, şekilden de görüldüğü gibi, $p = 1/2$ için, yani ikili değerlerin eşit olasılıklar aldığı durumda, entropi maximum değerini alır.



Şekil 2.1. p raslantı değişkeni ile h(p) entropi arasındaki ilişki

Koşullu Entropi: X ve Y değişkenleri, iki ayrık zamanlı raslantı değişkenleri olmak üzere, $x \in X, y \in Y$ ve $p(x, y)$ birleşik olasılık dağılım fonksiyonu tanımlı olmak üzere, Y raslantı değişkeninin bilindiği durumda X'in koşullu entropisi $H(X \mid Y)$ şöyle tanımlanır,

$$H(X \mid Y) = - \sum_{x \in X} \sum_{y \in Y} p(x,y) \log p(x \mid y) \quad (2.3)$$

Karşılıklı Bilgi: X ve Y raslantı değişkenlerini ve $p(x, y)$ de bunların birleşik olasılık dağılım fonksiyonunu göstermek üzere, X ve Y'nin karşılıklı bilgisi şu şekilde hesaplanır,

$$I(X;Y) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (2.4)$$

İki raslantı değişkeni arasındaki karşılıklı bilgi entropi ve koşullu entropi kullanılarak aşağıdaki gibi de hesaplanabilir,

$$I(X;Y) = H(X) - H(X \mid Y) \quad (2.5)$$

$$= H(Y) - H(Y \mid X) \quad (2.6)$$

$$= I(Y;X) \quad (2.7)$$

İfadelerden de görülebileceği gibi X ve Y arasındaki karşılıklı bilgi simetriktir. Karşılıklı bilgi değeri negatif değer olamaz ve X ile Y birbirinden bağımsız değişkenler ise sıfır değerini alır.

2.2. Temel Haberleşme Teorisi

KAD'da BU probleminde, kuantum kanaldan iletilen mesajlarda oluşan hataları modellemek için genellikle ayrık hafızasız kanal kullanılır. Kanalın giriş ve çıkış mesajları, X ve Y, ikili değerlerden oluştuğu için ve bunların koşullu olasılıkları simetrik olduğu için, $p(x \mid y) = p(y \mid x)$, bu ayrık hafızasız kanal literatürde genellikle İkili Simetrik Kanal (İSK) olarak modellenir. Bir kanaldan bilinen bir x mesajı iletildiğinde çıkışta *y* mesajı gözlemlenmesi olasılığı $p(y \mid x)$ kanalın daha önceki giriş ve çıkış mesajlarından bağımsız ise, bu kanala hafızasız kanal denir [46].

Haberleşme Kanalı: Haberleşme kanalı girişine verilen bir mesaj için bir çıkış üreten ve çıkışı girişine olasılıksal olarak bağımlı olan bir sistemdir. Bu durum, x giriş mesajını ve y de çıkışı göstermek üzere olasılık geçiş matrisi $p(y \mid x)$ ile tanımlanır ve belirli bir giriş için çıkışın koşullu dağılımını gösterir.

Kanal Kapasitesi: X ve Y raslantı değişkenlerini göstermek üzere; kanal kapasitesi *C*, giriş ve çıkış mesajları arasındaki karşılıklı bilginin maksimum değeri olarak tanımlanmaktadır.

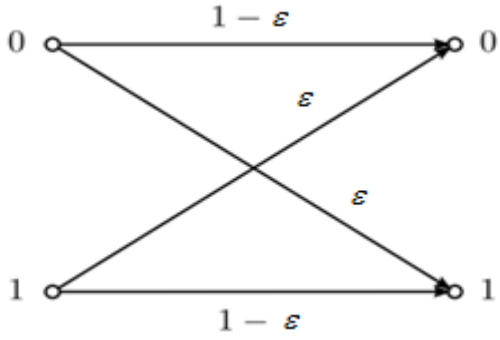
$$C = \max_{p(x)} I(X;Y) \quad (2.8)$$

Kanal kapasitesi, x giriş mesajını göstermek üzere, bütün p(x) dağılımlarından karşılıklı bilgiyi maksimum yapan değer kullanılarak hesaplanır.

Bilgi Oranı: M bilgi taşıyan bitleri, E bilgi uzlaştırma işlemi için gereken fazlalık/ilave bit sayısını ve N de gönderilen toplam bit sayısını göstermek üzere, R bilgi oranı şöyle tanımlanır,

$$R = \frac{M}{N} = \frac{N - E}{N} = 1 - \frac{E}{N} \tag{2.9}$$

İkili Simetrik Kanal: İSK kanal üzerinde yapılan haberleşmede hataların olabileceği kabul edilen en basit kanallardan biridir. Bu ayrık ve hafızasız kanaldaki giriş ve çıkış mesajları sadece $\text{İ} = \{0, 1\}$ kümesinden ikili değerler alır. Bu kanalda oluşan bir hata sonucu bitin değeri ikili kümedeki diğer bit değerine dönüşür, örneğin $0 \rightarrow 1$. Bu durum Şekil 2.2'de gösterilmiştir. İ kümesindeki her bit için hata oluşma ihtimali (geçiş olasılığı), ε, sabit olduğundan ötürü böyle bir kanala simetrik kanal denilebilir. Böyle bir kanalı ε geçiş olasılık değeri ile temsil etmek yeterli olacaktır ve genellikle $\text{İSK}(\varepsilon)$ ile gösterilir.



Şekil 2.2. Kanal hata olasılığı ε olan ikili simetrik kanal için geçiş olasılıkları

Böyle bir kanalda, giriş ve çıkış mesajları arasındaki karşılıklı bilginin üst sınırı şöyle hesaplanır [48],

$$I(X;Y) = H(Y) - H(Y \mid X) \tag{2.10}$$

$$= H(Y) - \sum p(x)H(Y \mid X = x) \tag{2.11}$$

$$= H(Y) - \sum p(x)H(\varepsilon) \tag{2.12}$$

$$= H(Y) - H(\varepsilon) \tag{2.13}$$

$$\leq 1 - H(\varepsilon) \tag{2.14}$$

Böylece, h(ε) Denklem (2.2)'de verildiği gibi ikili Shannon entropisini göstermek üzere, Denklem (2.8)'de verilen kapasite ifadesi şu hale gelecektir,

$$C_{İSK(\varepsilon)} = 1 - h(\varepsilon) \tag{2.15}$$

2.3. Verimlilik Tanımları

Bu tez çalışmasında kuantum kanalın giriş ve çıkış mesajlarının sadece $İ = \{0, 1\}$ kümesinden ikili değerler aldığı kabul edilmektedir. Bu nedenle kuantum kanal, ikili simetrik kanal olarak modellenecektir ve bu bölümde verilen verimlilik ifadeleri ayrık değişkenli KAD durumu için geçerlidir.

N uzunluklu A ve B mesajları, sırasıyla gönderici ve alıcı tarafındaki gizli mesajları ve ε da kuantum kanalın bit hata olasılığını göstersin. Bu durumda, birbiriyle kısmen ilintili A ve B gibi iki raslantı değişkeni arasındaki koşullu entropi şöyle hesaplanır,

$$H(A \mid B) = Nh(\varepsilon) \tag{2.16}$$

A ve B mesajları arasındaki başarılı bir BU işlemi için en az $Nh(\varepsilon)$ kadar ilave bilgi gönderilmesi gerekmektedir. Bu alt sınıra Shannon limiti ya da teorik alt limit adı da verilir. Gönderici ve alıcı arasında BU işlemi için değiş tokuş edilen ilave bit sayısı E ile gösterilirse, verimliliği hesaplamanın bir yolu şöyle olur [23],

$$\mu = 1 - \frac{E}{N} \tag{2.17}$$

E/N açığa çıkan (sızan) bilgi oranı, N uzunluklu bir mesaj için BU işleminde ne oranda ilave bilgi değiş tokuşu yapıldığını gösterir.

Literatürdeki bir diğer verimlilik hesabı ise Denklem **(**2.15)'te verilen kanal kapasitesi üzerinden yapılır [24]. Denklem (2.18)'de, paydadaki ifade kanal kapasitesidir ve μ'nün teorik üst limitini ifade eder ($\mu \leq C$). Bu oran ise, BU verimliliğinin teorik üst limite ne oranda yaklaştığını gösterir ve ideal durumda bu oran 1'e eşittir.

$$\beta = \frac{\mu}{1 - h(\varepsilon)} \tag{2.18}$$

Üçüncü ve son verimlilik ifadesi de başarılı bir BU işlemi için gerekenden ne kadar fazla bilgi değiş tokuşu yapıldığının bir ölçüsü olarak tanımlanmıştır. Shannon teorisine göre kanal hata olasılığı ε olan bir kanalda başarılı bir BU için en az $Nh(\varepsilon)$ kadar ilave bilgi değiş tokuşu yapılması gerekiyordu (teorik alt limit). Aşağıdaki verimlilik ifadesi bu limitin ne kadar aşıldığını gösterir [24],

$$\eta = \frac{E}{Nh(\varepsilon)} = \frac{1 - \mu}{h(\varepsilon)} \tag{2.19}$$

BU tekniklerinde yöntemler her zaman başarılı olmayabilir. Diğer bir deyişle, tüm hatalar düzeltilemeyebilir. Bu nedenle, BU tekniklerinin ürettiği sonuçların kalitesi/doğruluğu problemi ortaya çıkar. Bu nedenle, herhangi bir BU tekniği için verimlilik değerleri verilirken genellikle tekniğin ne oranda başarılı olduğunu gösteren Mesaj Hata Olasılığı (FER: Frame Error Rate) ve Bit Hata Olasılığı (BER: Bit Error Rate) gibi ifadeler de sunulur. BU tekniklerde verimlilik hesabı yapılırken bir mesaj üzerinde defalarca BU yöntemi çalıştırılır, örn. 10000 defa. Bu denemelerden bazıları başarısız olabilmektedir. FER, bu denemeler sonucunda en az bir hata içeren deneme sayısının toplam deneme sayısına oranını ifade eder. BER ise, tüm bu denemeler sonrasında düzeltilemeyen (hatalı) bitlerin sayısının toplam bit sayısına oranını gösterir. Verimlilik değerlerinde FER ve BER değerleri çok belirleyici olmaktadır. FER yüksekse, verimlilik değerlerinin yüksek olması çok önemli olmayacaktır. Çünkü yüksek sayıdaki BU denemesi başarısız sonuçlanmıştır ve bu mesajlar için yapılan fazlalık bilgi değiş tokuşu verimlilik hesabına katılmamıştır. Bu nedenle, verimlilik değerleri FER ve BER ile birlikte hesaplanmalıdır. Diğer bir deyişle, FER ve BER'in ortaya çıktığı her durumda yukarıda verilen verimlilik ifadeleri aşağıdaki gibi olacaktır,

$$\mu_{FER} = (1 - FER)\mu \tag{2.20}$$

$$\beta_{FER} = (1 - FER)\beta \tag{2.21}$$

$$\eta_{FER} = \frac{(1 - FER)(1 - \mu) + FER}{h(\varepsilon)} \tag{2.22}$$

Denklem (2.20), (2.21), (2.22)'deki (1 - FER) ifadesi başarılı BU deneme oranını, (1 - μ) ise değiş tokuş edilen ilave bilgi oranını (sızan bilgi miktarı) temsil etmektedir.

Bu ifadelerdeki FER yerine BER de kullanılabilmektedir. BU_H hatalı sonuçlanan deneme sayısını ve BU_T ise toplam deneme sayısını göstermek üzere, FER aşağıdaki gibi hesaplanabilir,

$$FER = \frac{BU_H}{BU_T} \tag{2.23}$$

B_H hatası düzeltilemeyen bit sayısını ve B_T ise toplam bit sayısını göstermek üzere, BER aşağıdaki gibi hesaplanabilir,

$$BER = \frac{B_H}{B_T} \tag{2.24}$$

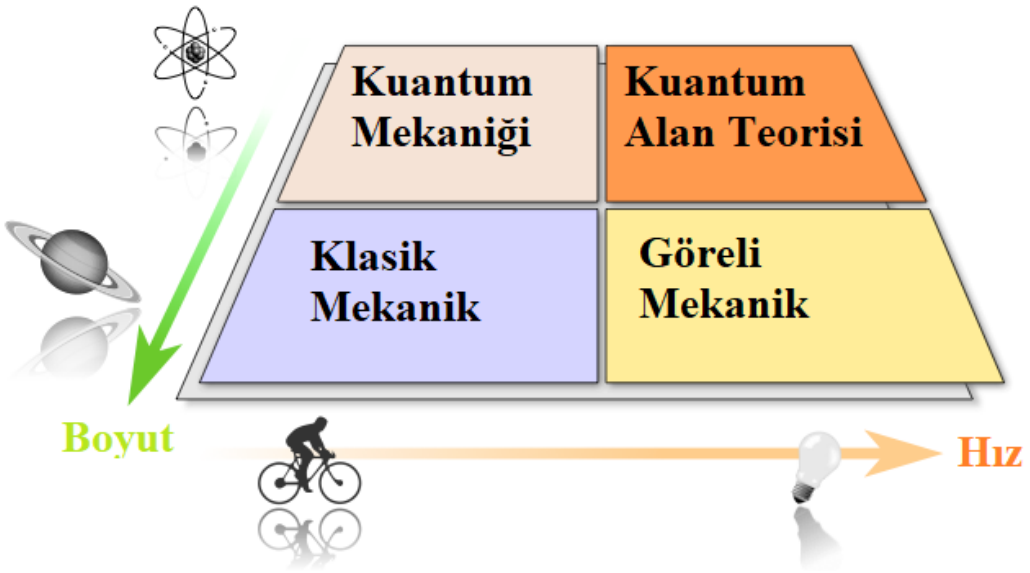
2.4. Kuantum Mekaniği

Atomların ve atom altı parçacıkların (foton, elektron, çekirdek vb.) fiziksel olarak tarifine imkan sunan teoriye kuantum mekaniği adı verilir. Latince nicelik anlamı taşıyan "quantus" kelimesinden dilimize geçmiştir. Bazı maddelerin ısıtılması sonucu ışık yayması fikri üzerinden yola çıkılmış olup, ilk defa 1900 yılında Max Planck tarafından ortaya atılan Planck yasası ile kuantum mekaniğinin temelleri atılmıştır. Daha sonra bu konuda Albert Einstein, Niels Bohr, Werner Heisenberg, Max Born, John Von Neumann, Paul Dirac ve Wolfgang Pauli'nin çalışmaları olmuştur. Bugünkü manasıyla incelediğimiz kuantum mekaniği ise Erwin Schrödinger tarafından 1927 yılında ortaya atılmıştır ve Schrödinger denklemleriyle modellenmektedir. Schrödinger denklemleri ile doğadaki çok küçük bileşenler (mikro düzeyde) üzerinde yapılan deneyler uyumlu sonuçlar verdiği için yöntem başarılı olarak kabul görmektedir [49].

Kuantum mekaniğinde klasik fizikte hiç bahsi geçmeyen yeni kavramlar ele alınmaktadır. Bunlar: ışığın parçacıklardan/taneciklerden oluşması, bu parçacıkların dalga özelliğine de sahip olması ve neredeyse tüm fiziksel maddelerin kuantum yapısında olmasıdır. Bu konuda ortaya atılan temel problemlerde genellikle aktörler boyutu çok küçük ancak hızı çok yüksek olan atom ve atom altı parçacıklardır. Problemler bu atom altı parçacıkların ışık ve elektromanyetik alanlara maruz kaldığı durumlarda ortaya çıkmıştır. Bu problemlerden en meşhur olanları elektronlarla kırınım, fotoelektrik olay, siyah cisim ışıması, Compton olayı vb. gibi sayılabilir. Bahsi geçen bu problemler ilk aşamada bilimsel dayanağı olmayan ve varsayımlara dayanan fikirlerle açıklanmaya çalışılmıştır. Ancak, yukarıda da bahsedildiği gibi 1927 yılında ortaya atılan Schrödinger denklemleriyle kuantum mekaniği teorisi önerilmiş ve bu problemlere bilimsel bir açıklama getirilmiştir.

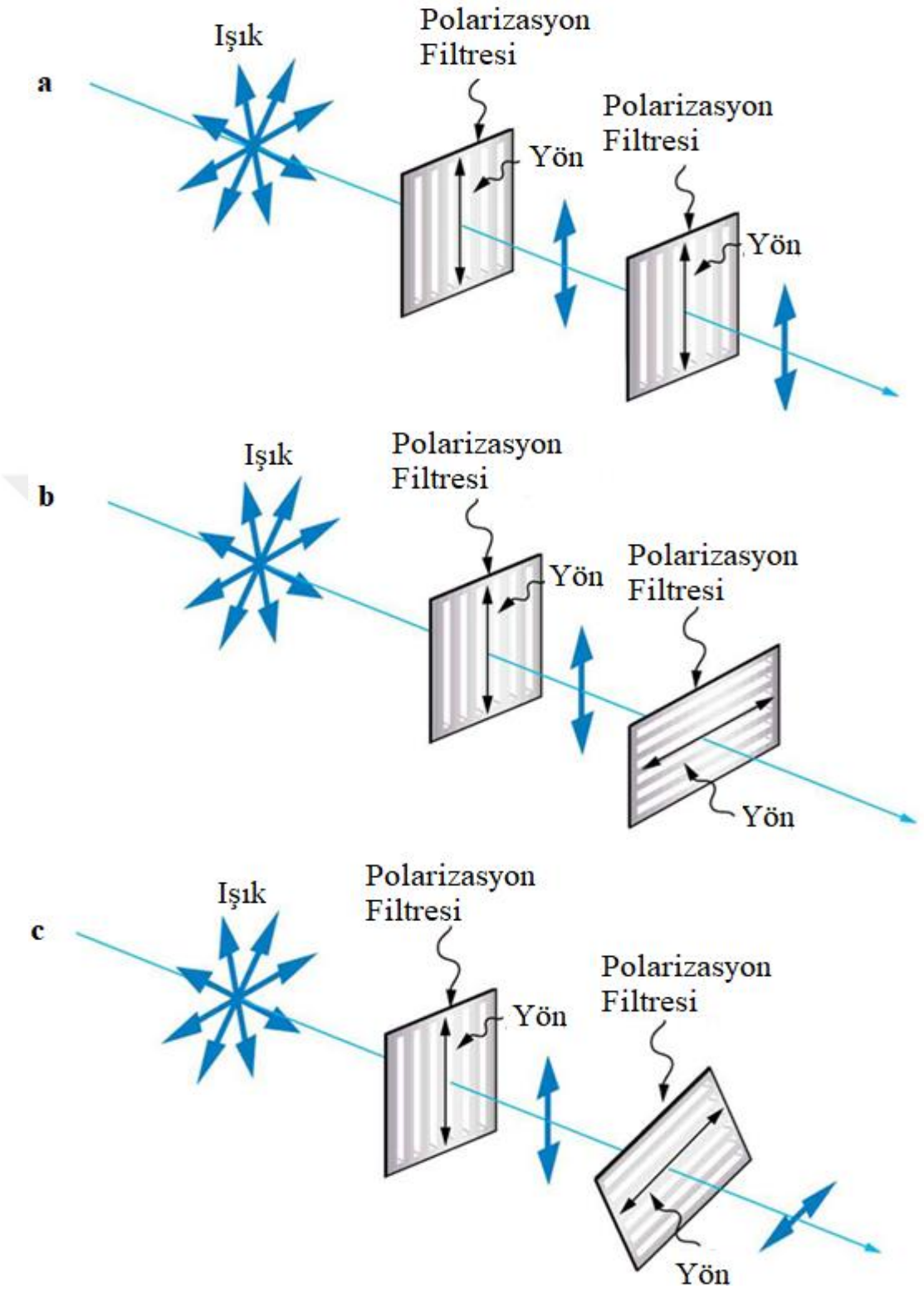
Mikro düzeydeki bu cisimlerle çalışırken klasik mekanik kuralları geçerli olmamaktadır. Bu ölçekteki cisimlerin hareketlerini modellemek için Şekil 2.3'te gösterilen ve yeni bir model olan kuantum mekaniği kullanılmaktadır.

Kuantum mekaniği atom ve atom altı parçacıklarla ilgilendiği için klasik fiziktekinin aksine bu alanda yapılan deneylerde özel araç ve cihazlara ihtiyaç vardır. Diğer bir deyişle, yapılan deneylerde kullanılmak üzere öncelikle elektron ve foton gibi parçacıkların görüntülenmesine imkan veren donanımlara ihtiyaç vardır.



Şekil 2.3. Mekaniğin 4 büyük uğraş alanı [49]

Kuantum mekaniği klasik mekaniğe nazaran anlaşılması zor bir alandır. Literatürde kuantum mekaniğinin etkilerini göstermek üzere yaygın olarak kullanılan deneylerden biri filtreler deneyi olarak geçer [50]. Şekil 2.4'te gösterildiği gibi, filtreler deneyinde öncelikle ışık kaynağının önüne dikey polarizasyon filtresi koyularak dikey polarizasyonlu bir foton elde edilmiştir. Ardından, deneyde kullanmak üzere dikey, yatay ve 45^o polarizasyona sahip üç adet polarizasyon filtresi hazırlanmıştır. Öncelikle, dikey polarizasyonlu fotonun önüne dikey polarizasyonlu bir filtre koyulmuştur. Foton ve filtrenin yönü aynı ve dikey olduğu için foton filtreden geçebilmiştir. Daha sonra, fotonun hemen arkasına yatay polarizasyonlu filtre koyulmuştur. Fotonun yönü dikey ve polarizasyon filtresinin polarizasyon yönü yatay olduğu için filtre çıkışında herhangi bir foton gözlemlenmemiştir. Son olarak dikey polarizasyonlu fotonun önüne 45^o'ye sahip olan polarizasyon filtresi eklenmiştir. Bu polarizasyon filtresinin yönü 45^o'dir ve ilginç bir şekilde filtre çıkışında foton gözlemlenebilmiştir. Bu durum kuantum mekaniği ile açıklanabilmektedir ve kuantum mekaniğinde bunun benzeri birçok durum mevcuttur. Kuantum mekaniğinin bu gariplikleri kullanılarak, kırılması çok zor denilen kodların kırılabilmesi, rasgele sayı üretme, hattı dinleyen bir saldırganın olup olmadığını anlayabilme ve bilginin ışınlanabilmesi gibi uygulamalar yapılabilmektedir [51].



Şekil 2.4. Filtreler deneyi: ışığın yönü ve filtrelerin polarizasyonuna göre oluşan durumlar [50, 51]

2.4.1. Heisenberg belirsizlik yasası

Kuantum mekaniğinin temel yasalarından biridir. Alman fizikçi Werner Heisenberg atom altı parçacıklar üzerinde yürüttüğü deneylerde klasik fizikte yaygın olarak kullanılan momentum ve konum hesaplarının mikro ölçekteki parçacıklar üzerinde hesaplanamadığını farketmiştir. 1927 yılında ortaya attığı bu fikre göre atom altı parçacıklarının momentumu ne kadar doğru ölçülebiliyorsa konumu da bir o kadar belirsiz olmaktadır [52, 53]. Benzer şekilde, konumu ne kadar doğru ölçülebiliyorsa momentumu da bir o kadar belirsiz olmaktadır [52, 53].

[52, 53]'te de anlatılan Heisenberg'in yaptığı bu deneylerde de, atom altı parçacık olarak bir elektron seçilmiştir ve elektronun ölçülmesi hedeflenen fiziksel nitelikleri de konum ve momentum olarak belirlenmiştir. Burada, elektron kuantum mekaniğinin kurallarının geçerli olduğu kuantum sistemini ve konum ve momentum da kuantum sisteminde ölçülecek fiziksel nitelikleri temsil etmektedir. Klasik fizikte kütlesi *m* olan bir cisme F kuvveti uygulandığında cismin ivmelenmesi hakkında kolayca yorum yapılabilmektedir ($a = F/m$, a: ivme). Benzer şekilde, Heisenberg elektrona bir etkide bulunup bunun sonucunda elektronun yeni konumunu ve momentumunu ölçmek istemiştir. Elektrona etkide bulunmak üzere çeşitli dalga boylarında ışık göndermiştir. Deneyleri sonucunda momentumun hesaplanması için uzun dalga boyundaki ışıklara ihtiyacı varken konumun hesaplanması için kısa dalga boylarında ışığa ihtiyacı olmuştur. Örneğin, momentumdaki belirsizliği minimuma indirgemek için bir o kadar uzun dalga boyunda ışık göndermesi gerekmektedir. Benzer şekilde, konumu doğru ölçebilmek için çok kısa dalga boylarında ışık kullanmıştır. Bu örnekten kolayca görülebileceği gibi, bir fiziksel büyüklüğü hesaplamak için yapılan deney diğer fiziksel büyüklüğün doğru bir şekilde hesaplanmasını imkansız hale getirmektedir.

Makro boyutlarda konum ve momentum değişikliklerindeki belirsizlik miktarı çok küçük olduğu için ihmal edilmektedir. Ancak mikro boyutlarda çalışırken belirsizlik ilkesi önemli bir duruma gelmektedir ve ihmal edilememektedir. Burada dikkat edilmesi gereken bir husus da bu durumun ölçüm yapılan sistemlerle ilgili değil de, doğanın bir kuralı olarak ortaya çıkan bir sonuç olduğudur. Deneylerde olumsuz gibi görünen durum günümüzde gizli anahtarın güvenli dağıtımı problemini çözmek için kullanılmaktadır. Atom altı bir parçacığın fiziksel niteliklerinden birinde yapılan bir

ölçüm diğer niteliklerinin ölçülememesine neden olacağı için böyle sistemler tam manasıyla kopyalanamaz ve böyle sistemlerden taşınan bilgiler doğanın koyduğu kuralların bir sonucu olarak ele geçirilemez [52, 53].

2.4.2. Kuantum mekaniği kopyalanamazlık teorisi

Kuantum mekaniğinin bir diğer temel yasası bu teoridir. William Kent Wootters ve Wojciech Hubert Żurek tarafından 1982 yılında, kuantum mekaniğinin kullanıldığı kuantum sistemlerin kopyalanmasının mümkün olmadığı ortaya atılmıştır [54]. Eğer bilgi böyle bir sistemin bir parçası olarak temsil edilebilirse, bu sistem üzerinde taşınan bilgilerin kopyalanması da mümkün olmayacaktır. Örneğin, bir bilgi atom, elektron, foton gibi kuantum parçacıklarının spin, polarizasyon vb. gibi belirli fiziksel özellikleri ile taşınırsa, kuantum parçacıklarının durumları eş zamanlı olarak ölçülemeyeceği için kuantum parçacığı üzerinde taşınan bilgi de kopyalanamayacaktır [80]. Bilinmeyen/ölçülemeyen bir fiziksel nitelik kopyalanamayacağı için saldırganlar böyle bir sistemdeki bilgiyi de ele geçiremezler.

Daha öncede bahsedildiği gibi, kuantum mekaniğinde tanımlı olan kurallara göre modellenen kuantum parçacıkları üzerinde taşınan bilgiye kuantum bilgisi adı verilmektedir. Klasik matematikte iki seviyeli bir durum basitçe 0 ve 1'lerle temsil edilmektedir. Bu şekilde iki seviyeden oluşan durumlar kuantum mekaniğinin geçerli olduğu bir sistemde yine bitlerle temsil edilebilmektedir. Kuantum bit ya da kübit olarak adlandırılan bu sistem klasik bitlerde olduğu gibi 0 ve 1'lerden oluşur [80].

Kuantum kopyalanamamazlık teorisi BB84 protokolünün güvenliğinin omurgasını oluşturmaktadır. Bu teorem kuantum parçacıklarının fiziksel özelliklerinin kopyalarının alınabilmesinin mümkün olmadığını ifade etmektedir [54, 80].

2.5. Kriptografi

Kriptografi gizli bir bilginin sadece alıcıları tarafından erişilebilmesine imkan sağlayan bilgi gizleme sanatıdır. Kriptografinin tarihi eski çağlara kadar uzanmaktadır. Tarihte bir mesajı gizli bir şekilde karşı tarafa aktarma ihtiyacı hep olmuştur ve bu ihtiyaç için farklı yollar denenmiştir. Milattan önce 2000'li yıllara ait eski Mısır mezarlarındaki rasgele düzenlenmiş hiyerogliflerden günümüzün gelişmiş şifreleme

standardartlarının tamamı kriptografiye birer örnek olarak verilebilmektedir. Kriptografi zamanın teknolojik şartlarına göre şekil değiştirse de, nasıl yapıldığından bağımsız olarak amaç her zaman aynı olmuştur; bilgiyi yetkisiz kişilerin erişiminden korumak. Bu maksatla, açık bir metin bir şifre kullanılarak şifrelenmekte veya belirli bir algoritmaya göre kodlanarak okunamaz bir forma sokulmaktadır. Bilgiyi alan taraf sahip olduğu şifreyi kullanır ve şifrelenmiş metnin şifresini çözerek yeniden okunabilir metni elde etmektedir.

Kriptografi temel olarak dört amaç için kullanılır. Orijinal ve en eski amacı gönderici ve alıcı arasındaki bilgi aktarımını güvenli ve gizli bir şekilde gerçekleştirmektir. Böylece, yetkisiz bir alıcı veya haberleşmeyi dinleyen bir saldırgan şifrelenmiş bilgiyi elde etse dahi bunu anlayamayacak ve kodlanmış verinin şifresini çözemeyecektir. Kriptografi günümüzde ise, kimlik doğrulama, bütünlük ve inkar edilememezlik için de kullanılmaktadır [43]. Kimlik doğrulama ile, alıcı bir mesajın gerçekten iddia edilen gönderici tarafından gönderildiğini kanıtlayabilmektedir. Bütünlük ise alıcı tarafa ulaşan mesajın iletimi boyunca değiştirilmediğinin kanıtını ifade eder. Eğer mesajın bütünlüğünde bir değişiklik olduysa, bu alıcı tarafından anlaşılabilmektedir. İnkar edilememezlik ise, mesajın gönderici tarafından gönderildiğini ve alıcı tarafından da alındığını ispat etmektedir. Diğer bir deyişle, gönderici mesajı gönderdiğini ve alıcı da aldığını inkar edemez. Teknolojik gelişmelerle birlikte gelişmiş bilgisayarlar kompleks şifreleri kırabilir hale gelmiş ve böylece şifreleme ve şifre çözme yöntemleri de oldukça karmaşıklaşmıştır.

Modern kriptografi sistemlerinde aşağıdaki yöntemlerden bir veya birkaçı birlikte kullanılmaktadır:

- simetrik veya gizli anahtar,
- asimetrik veya açık anahtar,
- özet fonksiyonları [43].

Gizli anahtarlı sistemlerde, hem gönderici hem de alıcı aynı gizli anahtara sahiptir ve bir mesajı şifrelemek ya da onun şifresini çözmek için aynı gizli anahtarı kullanmaktadır [43]. Sistemde birbiriyle haberleşen birden fazla kullanıcı varsa, kimlik doğrulama ve inkar edilememezlik gereksinimlerini de karşılamak için hem gönderici hem de alıcı bir dizi gizli anahtar tutacaktır (her alıcı için bir tane). Sonuç

olarak, bu gizli anahtarların yönetimi de karmaşıklaşmaktadır. Eğer sistemde sızan bir bilgi olursa, tüm gizli anahtarların yeniden üretilip tüm kullanıcılara yeniden dağıtılması gibi problemler de ortaya çıkmaktadır. Bu duruma gizli anahtar dağıtımı da denilmektedir. Bu işlemdeki temel sorun ise, şifreli haberleşmeye başlamadan önce bu gizli anahtarların tüm kullanıcılara güvenli bir şekilde dağıtılması gerekmektedir. Gizli anahtarlar özel bir kurye ile veya bir ajan ile dağıtılarak bu problem çözülmeye çalışılmak istenebilir. Ancak, bu çok önemli bir zaafiyet ortaya çıkarmaktadır. Özellikle birisinin bu gizli anahtar(lar)ı ele geçirmek istemesi durumunda anahtar dağıtımı için daha güvenilir yolların bulunması gerekliliği aşikar olmaktadır.

Simetrik şifrelemedeki anahtar dağıtımı probleminin çözümüne önerilen bir yöntem bir diğer şifreleme tekniği olan asimetrik veya açık anahtar şifreleme yöntemidir [43]. Açık anahtar şifreleme tekniğinde herkesin erişebildiği bir açık anahtar ve bir de sadece kullanıcının sahip olduğu gizli anahtar vardır. Kullanıcılar bu anahtarlardan biriyle şifreledikleri bir mesajı sadece diğer anahtarı kullanarak çözebilmektedirler. Asimetrik şifreleme teknikleri anahtar yönetimi açısından karşılaştırıldığında simetrik tekniklere göre daha gürbüzdür. Ancak, bu sistemler üretilen anahtarların birbirinden elde edilmesinin mutlaka önüne geçmelidirler. Bunun için özet fonksiyonları kullanılmaktadır. Özet fonksiyonları uzun açık bir metni şifreli kısa bir metne dönüştürürler [43]. Özet fonksiyonları tasarlanış amaçları gereği özet bilgisinden orijinal metne dönüş mümkün değildir.

Günümüzde birçok kriptografik protokoller ve yöntemler gizli anahtarın gizliliği ve güvenliğini sağlamak adına matematiksel fonksiyonların karmaşıklığına güvendiği için, yeteri kadar hesaplama imkanı sunan sistemler kullanıldığı takdirde hepsi olmasa da bu sistemlerin tamamına yakını kırılabilecektir. Hatta birçok sistem hattı dinleyen saldırganların varlığını da tespit edememektedir. Böyle sistemlerde, hattı dinleyen bir saldırgan muhtemelen gizli anahtarı elde etmiş ve hattı gerçekten de dinliyordur. Bütün bu kısıtlamalara göğüs germek için, bir dizi kuantum kriptografi ve KAD teknikleri kullanılarak gizli anahtarlar üretilir ve gönderici ile alıcı arasında anahtar dağıtımı yapılır [43].

3. KUANTUM ANAHTAR DAĞITIMI

Gizli bir haberleşmede kullanılacak gizli anahtarın gönderici ve alıcıya güvenli bir şekilde ulaştırılması her zaman önemli bir problem olmuştur. Bir önceki bölümde bahsi geçen Heisenberg belirsizlik ilkesi ve kuantum kopyalanamazlık teorisi gibi fiziksel yasaların kullanıldığı kuantum mekaniği ile bu problem çözülebilmektedir. Bu yasaların pratikte kullanıldığı lüteratürdeki en popüler uygulama KAD'dır. Böylece, iletişim kanalını dinleyen saldırgan ya hattı hiç dinleyemez ya da dinlese bile herhangi bir bilgi elde edemeyecektir. Ayrıca, hattı dinlemek için yapılan müdahaleler tespit edilebileceğinden ötürü iletişim kanalını dinleyen bir saldırganın olduğunun ortaya çıkmasını sağlayacaktır [55].

KAD'da ya da daha genel bir ifade ile kuantum kriptografide, bilginin korunması için matematiksel kurallar yerine fiziksel yasalar kullanılmaktadır. Bu da klasik kriptografik sistemlerden farklılık göstermektedir. Bilgi kuantum parçacıkları ve bunların kuantum nitelikleri kullanılarak taşınır. Kuantum yasalar atom altı parçacıkların doğası sonucu ortaya çıktığı için fizik alanındaki ve diğer teknolojik gelişmelerden/değişikliklerden de bağımsızdır.

Gönderici ve alıcı arasındaki anahtar dağıtımı problemini çözmek için herkesin erişimine açık optik kanallar (uçtan uca fiber optik kablo veya açık hava) kullanılır. Bilginin taşınması için kuantum parçacıklar (foton, elektron vb.) kullanıldığından kuantum mekaniğini kullanan bir iletişim tekniğidir. Yine kuantum mekaniğinin doğası gereği iletişim sırasında fotonlara yapılan herhangi bir müdahale veya ölçüm tespit edilebilmektedir. Böylece, hattı dinleyen bir saldırgan olsa dahi hattan iletilen bilgiyi kopyalayamayacak veya elde edemeyecektir. KAD'da kuantum parçacığı olarak genellikle fotonlar tercih edilmektedir. Işığı oluşturan parçacıkların her birine foton adı verilir. Teorik olarak elektron veya diğer atom altı parçacıklar da bu maksatla kullanılabilir. Ancak, fotonlar kuantum mekaniği yasalarına tümüyle uyan atom altı parçacıklardır ve kullanımı da diğerlerine göre daha iyi anlaşılmış durumdadır. Ayrıca, fotonların taşınması için halihazırda pratik altyapılar mevcuttur.

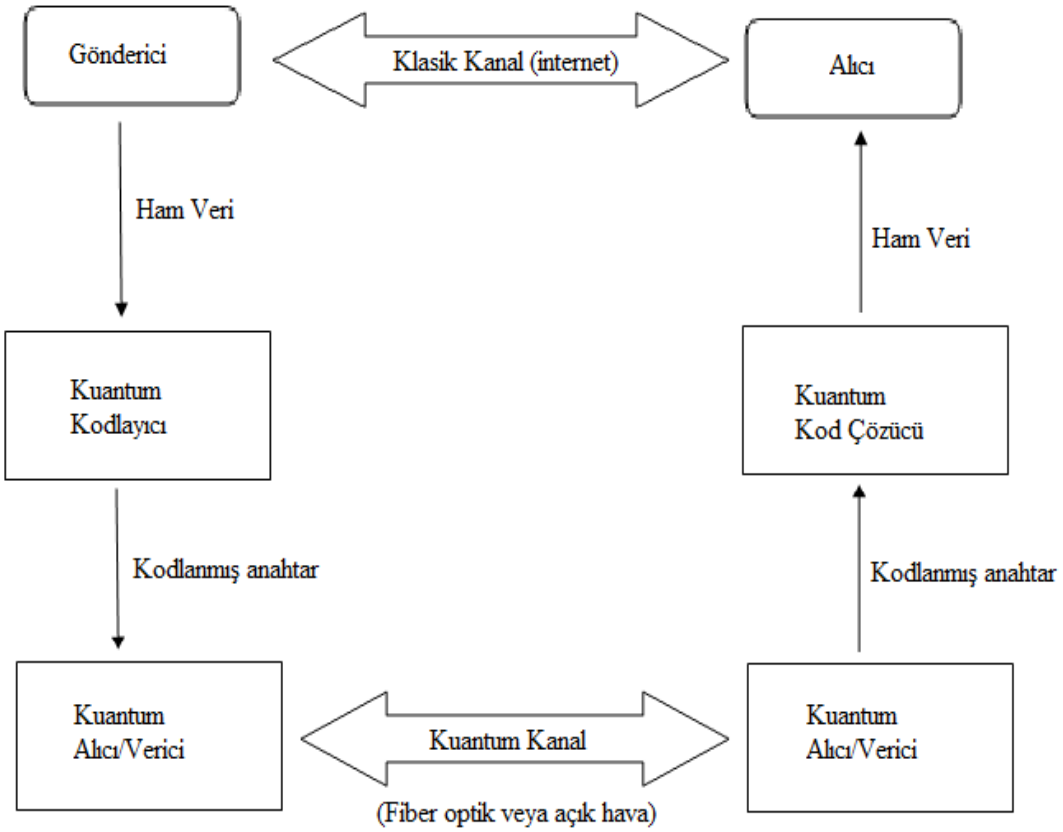
Günümüzde foton iletimine imkan veren ve çok yüksek bant genişliklerine sahip fiber optik kablolarla hızlı bir şekilde bilgi taşınabilmektedir. Fotonların dalga özelliği de gösterebilmesi sayesinde bilginin taşınması için dalga boyu da kullanılabilmektedir. Elektronlarda ise bu durum biraz farklıdır. Elektronların iletim ortamı ile etkileşime girmeden uzak mesafelere gönderilmesi güç ve pahalı bir işlemdir. Ayrıca, bir elektronun teorik olarak hızı ışık/foton hızına çıkabilse dahi pratikte daha düşüktür. Kuantum mekaniğinde bilgi kuantum parçacıklarının kuantum nitelikleri kullanılarak taşınıyordu. Bilginin, örn. bir bit, foton ile taşınması için kuantum nitelikler fotonun polarizasyonu, dalga boyu, frekansı ve faz bilgisi olabilir. Ancak, KAD uygulamalarında genellikle polarizasyon niteliği tercih edilmektedir [50, 51, 60, 80, 81].

KAD, güvenli anahtar dağıtımı problemini çözmek maksadıyla 1984 yılında ABD'li fizikçi Charles Henry Bennett ve Kanadalı kriptografi uzmanı Gilles Brassard tarafından önerilen bir yöntemdir. Bennett ve Brassard soyisimlerinin ilk harfleri ve yapılan çalışmanın yıl bilgisi birleştirilerek protokole BB84 adı verilmiştir [11]. Yazarlar, BB84 protokolünü 1989 yılında laboratuvar ortamında açık hava üzerinden denemiş ve aralarında 30cm'lik bir mesafe bulunan iki uç arasında 10 bit/saniye hızıyla bilgi aktarmayı başarmışlardır [56]. Günümüzde bu değerler fiber optik kablo ile yapılan haberleşmelerde 260km'lere [57], açık hava için ise 144km'lere [58] kadar çıkmaktadır ve 1 Mbps'lik hızlara da ulaşılmıştır [59].

BB84 protokolünde güvenli bir iletişim için kullanılacak gizli anahtarın her bir bitinin taşınması maksadıyla kuantum parçacığı olarak foton ve onun polarizasyon niteliği kullanılmaktadır [60, 80, 81].

Bu bölümde KAD probleminin kuantum mekaniği kullanılarak nasıl çözüldüğü BB84 protokolü üzerinden anlatılmaktadır. Şekil 3.1'de BB84 protokolünün aktörleri ve birçok KAD'ın kullandığı temel akış gösterilmektedir. Öncelikle, gönderici gizli anahtarı elde etmek maksadıyla rasgele bir bit dizisi üretir. Bu rasgele dizi gizli anahtarın boyundan daha büyüktür ve ham veri olarak adlandırılmaktadır. Kuantum kodlayıcı, bu ham bit dizisini belirli bir kodlama kuralına göre kodlayarak kuantum kanaldan iletime uygun bir hale getirir. Diğer bir deyişle, bu bitler için fotonlar üretilir. Rasgele bit dizisi için rasgele üretilen foton dizisi kuantum verici ile kuantum kanala

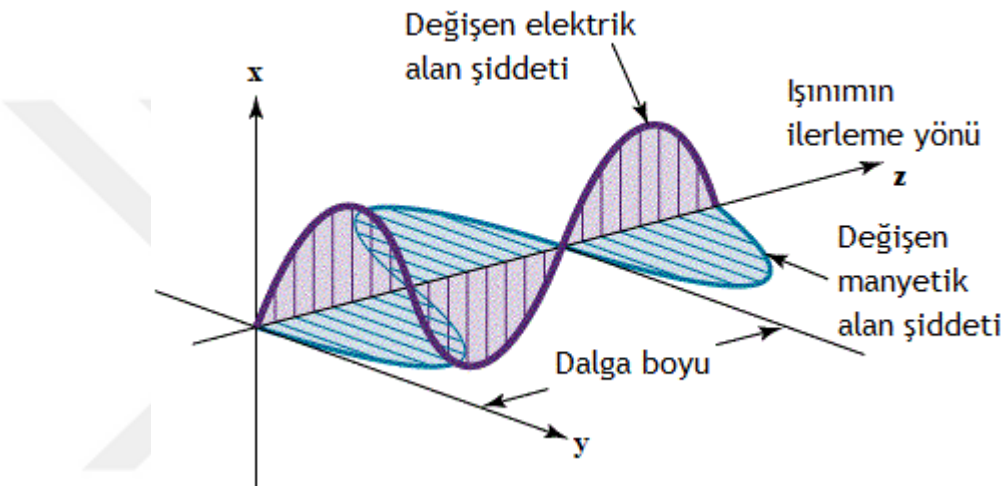
iletilir. Alıcı tarafında ise, kuantum alıcılar vasıtası ile ölçülen fotonlara göndericide yapılan işlemlerin tersi uygulanır. Alıcı tarafında, kodlama kuralına göre ölçülmüş olan bu fotonlardan bit dizisi elde edilir. Bu aşamada, gönderici ve alıcı aynı uzunluğa sahip ama muhtemelen birbirinden farklı bazı bitler içeren bit dizilerine sahip olmuşlardır. Bitlerdeki bu farklılık kuantum kanaldaki gürültü ve hattı dinleyen olası saldırganlardan kaynaklanmaktadır. Gönderici ve alıcı saldırgan tespiti için bazı bitleri feda ederler. Bu nedenle, gizli anahtar olarak kullanacakları bit dizisi ilk aşamada üretilen bit dizisinden daha az olacaktır. Son adımda ise, gönderici ve alıcı gürültülü kuantum kanal üzerinden ilettikleri bit dizisindeki hataları tespit etme ve düzeltme aşamasına geçmiştir. Bu hata sezme ve düzeltme adımı ise klasik/gürültüsüz bir kanaldan yapılmaktadır. Bu işlem HSD teknikleri, örn. CASCADE protokolü, kullanılarak yapılmaktadır. Başarılı bir HSD işlemi sonucunda, gönderici ve alıcı tamamen aynı değerlere sahip bir bit dizisini (gizli anahtar) elde etmektedir.



Şekil 3.1. KAD'daki bileşenler ve iletişim kanalları

3.1. Foton Polarizasyonu

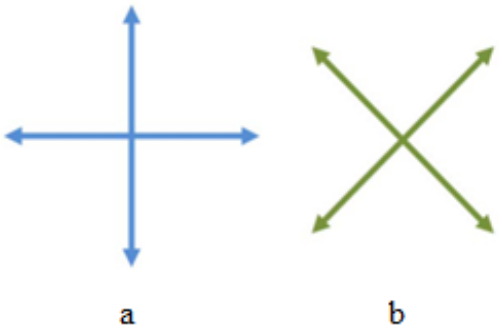
Bir kuantum parçacığı olan foton ile bilgi taşımak için kullanılan niteliklerinden biri polarizasyondur. Bir fotonun polarizasyonu x-y-z düzleminde Şekil 3.2'de gösterildiği gibidir. Şekilden de görülebileceği gibi z ekseninde ilerleyen bir fotonun y-z düzleminde manyetik alanı ve x-z düzleminde de elektrik alanı vardır. Bir fotonun polarizasyon niteliği ise elektromanyetik alanı içinde elektrik alanının nasıl davranış gösterdiği ile ilgilidir.



Şekil 3.2. Foton, elektrik alanı, manyetik alanı [80, 81]

Bir foton yayılırken elektrik alanı aynı düzlemde kalıyorsa polarizasyonu doğrusaldır. Eğer elektrik alanı belirli bir frekansta dönüyorsa dairesel polarize olmuştur. KAD'da, daha genel ifadeyle kuantum kriptografide, foton polarizasyonu için bu iki polarizasyon tipi de veya bunların kombinasyonu da kullanılabilmektedir. Anlaşılması basit olması açısından bundan sonraki örnekler doğrusal polarizasyon üzerinden anlatılacaktır.

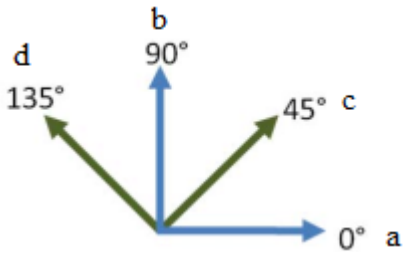
Hatırlanacağı üzere, kuantum mekaniğinde bilgi kuantum nitelikler üzerinde taşınmaktaydı. Bahse konu olan doğrusal polarizasyonda foton ile bilgi taşımada, diğer bir deyişle fotonun kodlanmasında, ise Şekil 3.3.a'da gösterildiği gibi kenarsal ve Şekil 3.3.b'de gösterildiği gibi diagonal polarizasyon tabanları kullanılır.



Şekil 3.3. Doğrusal polarizasyon tabanları. a. Kenarsal, b. Diagonal [80, 81]

Foton kodlamasında iki tane taban kullanılması kuantum kriptografi açısından çok önemlidir. Tahmin edilebileceği gibi, bu iki taban Heisenberg belirsizlik yasasındaki kuantum niteliklere karşılık gelmektedir. Dikkat edilirse, doğrusal polarizasyon ile foton kodlanmasında kullanılan bu iki taban pratik uygulamalarda ikili bitler ile temsil edilebilir [80, 81].

Doğrusal polarizasyonda tabanlar birbirine dik iki yönden oluşmaktadır. Bunlar Şekil 3.4'te gösterildiği gibi yatay, dikey ya da diagonal olabilir. Bu durumda, bir foton her bir taban için iki farklı şekilde polarize olabilir ve toplam dört farklı polarizasyon durumu olacaktır.

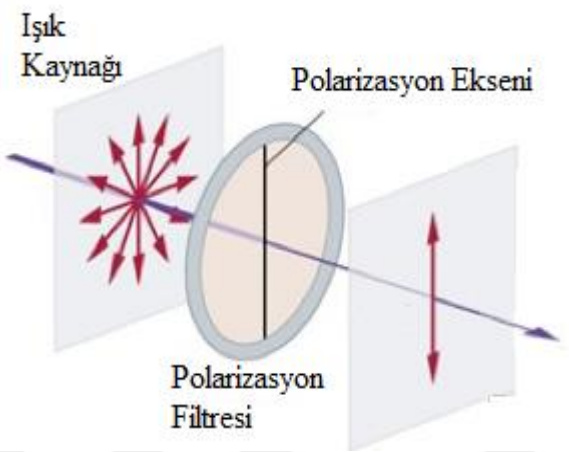


Şekil 3.4. Doğrusal polarizasyonda polarizasyon durumları. a. Yatay, b. Dikey, c. 45^{o} ve d. 135^{o}'lik diagonal

Değeri 0 veya 1 olan bir biti bir fotonun polarizasyon durumu olarak kodlayabilmek için öncelikle fotonun o polarizasyon durumuna getirilebilmesi gerekmektedir. Bir fotonun polarizasyonu elektrik alanının davranışı ile ilgili olduğu için belirli bir polarizasyon durumunu ortaya çıkarmak maksadıyla elektrik alanı istenilen düzlemde ilerleyen bir foton oluşturmak gerekmektedir. Bunu yapabilmek için de polarize

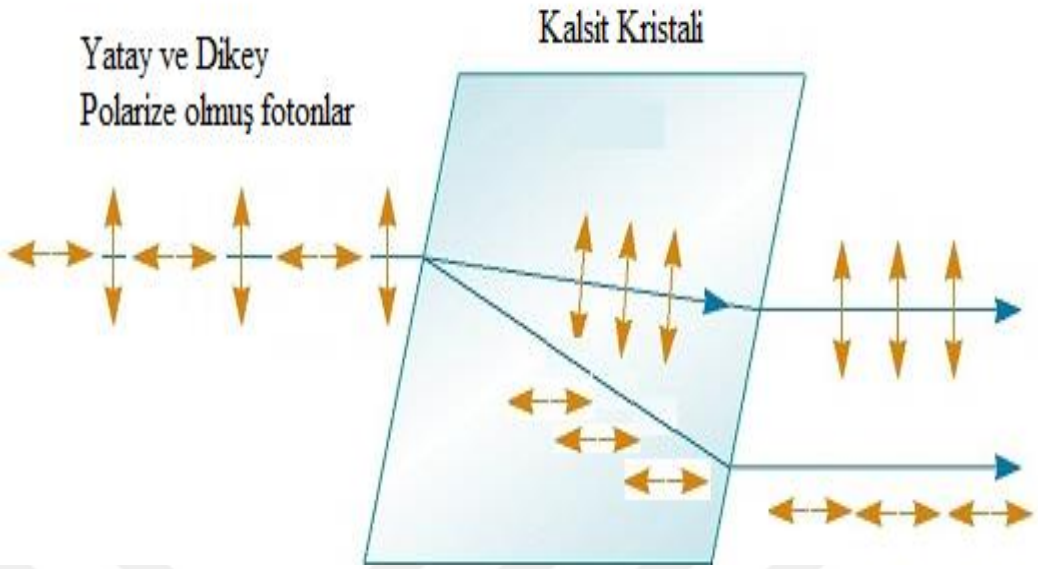
ediciler, örn. polarizasyon filtreleri, kullanılır. Polarize edicinin polarizasyon ekseni istenilen açıya ayarlandığında istenilen açıyla bir foton oluşturulmuş olur.

Bir ışık kaynağı Şekil 3.5'teki gibi dikey polarize edici özelliğine sahip bir filtreden geçirilir ve filtre çıkışında kenarsal polarizasyon tabanlarından dikey (90^o) polarizasyon durumuna sahip bir foton elde edilmiş olur. Benzer şekilde, amaca uygun polarize ediciler kullanılarak diğer polarizasyon durumlarına sahip fotonlar da üretilebilir.



Şekil 3.5. Dikey polarizasyon durumuna sahip foton üretme [80, 81]

Fotonlarla taşınan bu bilgilerin elde edilmesi için öncelikle her fotonun polarizasyonunun ölçülmesi gerekmektedir. Bunun için fotonun polarizasyonuna uygun polarizasyon ölçücüler kullanılmaktadır, örn. Kalsit kristali. Polarizasyon filtrelerine benzer olarak Kalsit kristali içerisinden geçen bir foton, kendisinin ve kristalin polarizasyon eksenine/açısına bağlı olarak ya doğrudan geçer ya da belirli bir eksen kaymasına uğrayarak kristalden çıkar [80, 81]. Şekil 3.6'da da gösterildiği gibi polarizasyon ekseni dikey olarak ayarlanmış bir Kalsit kristalinden sadece dikey olarak polarize edilmiş fotonlar dosdoğru geçebilmektedir. Yatay olarak polarize edilmiş fotonlar ise belirli bir eksen kaymasıyla filtreden geçebilmiştir. Bunun sebebi fotonun polarizasyon düzlemi ile kristalinkinin farklı olmasıdır.



Şekil 3.6. Fotonların polarizasyonlarının ölçülmesi [80, 81]

Uygun kristaller kullanılarak polarize edilmiş fotonlar ayırt edildikten sonra kristal çıkışlarına eklenen foton ölçücüler/dedektörler sayesinde foton olup olmadığı da tespit edilebilmektedir. Fotonun varlığı tespit edilen dedektörden fotonun polarizasyonu ve böylece o polarizasyonun taşıdığı bit değeri de ölçülmüş olur. Böyle bir ölçümde her polarizasyon tabanı için bir kristal kullanılır ve her polarizasyon durumu için de kristal çıkışına iki tane dedektör koyulur [80, 81].

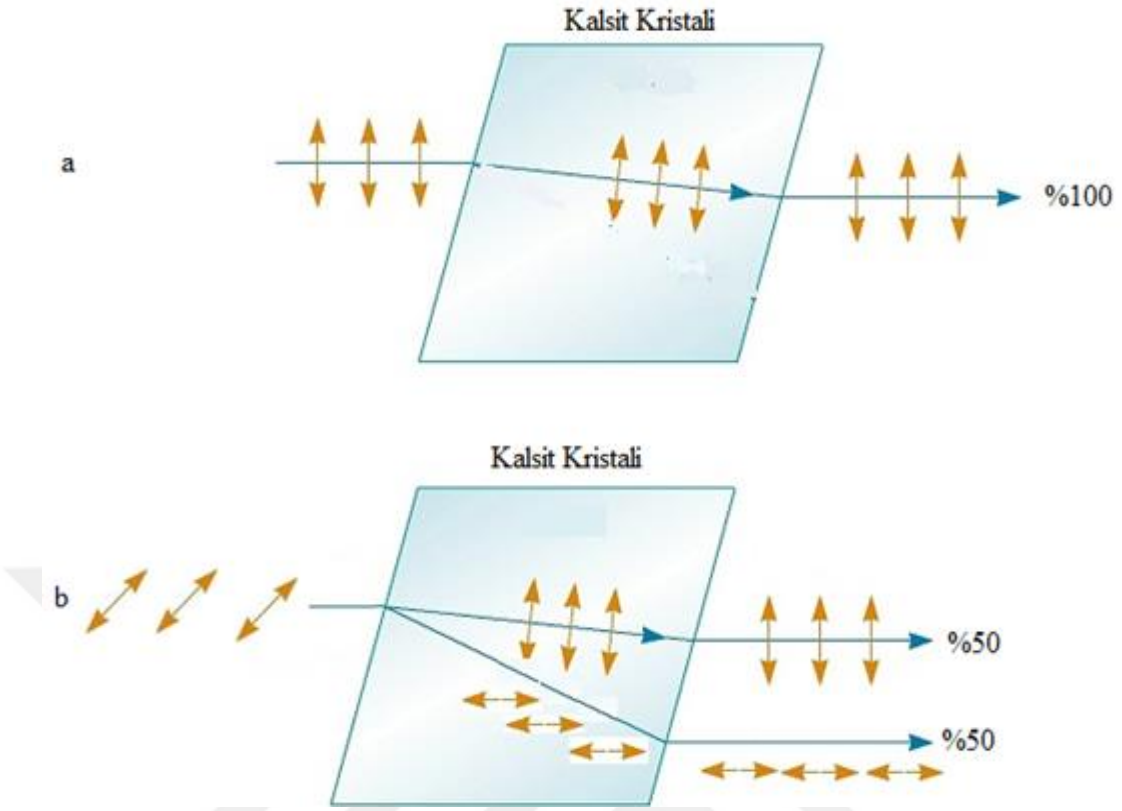
3.2. Fotonlarda Heisenberg Belirsizlik İlkesi ve Kopyalanamazlık Teoremi

Bir foton çok farklı şekillerde polarize edilmiş olabilir. Özellikle doğadaki fotonlarda bu durum çokça görülür. Polarizasyonu bilinmeyen bir fotonun polarizasyonunun ölçülebilmesi için önce polarizasyon ölçücüyü de belirlemek gerekmektedir. Bu ise mümkün değildir. Çünkü, fotonun polarizasyonunu bulmak için yanlış bir polarizasyon ölçücü kullanıldığında fotonun orijinal polarizasyonu bozulacak ve bu noktadan sonra fotonun orijinal polarizasyonu ölçülemeyecektir. Dolayısıyla, nasıl polarize edildiği bilinmeyen bir fotonun polarizasyonunu bulmak mümkün değildir. Ayrıca, bir fotonun polarizasyonu için doğru polarizasyon ölçücü kullanılmış olsa dahi foton ölçümden sonra dedektörde ısıya dönüşerek yok olmaktadır. Bundan dolayı aynı foton birden fazla ölçülememektedir [80, 81].

Daha önce de bahsedildiği gibi, foton da kuantum mekaniği alanında kullanılan bir kuantum parçacık olarak kabul edilmektedir. Dolayısıyla, kuantum mekaniğindeki kurallar, örn. Heisenberg Belirsizlik İlkesi, fotonlar için de geçerlidir. Örneğin, bir mesaj fotonlar ile kenarsal ve diagonal polarizasyon tabanları ile taşınıyor olsun. Her taban için ayrı bir Kalsit kristaline ihtiyaç olacaktır. Bu tabanlar tek kristal ile aynı anda ölçülemezler. Heisenberg belirsizlik ilkesi ile benzerlik kurulursa bu örnekteki kuantum nitelik çifti kenarsal ve diagonal polarizasyon tabanları olmaktadır. Özetle, bir fotondaki bilgiyi ele geçirmek için iki kuantum niteliğinin de aynı anda okunabiliyor olması lazım. Herhangi bir anda çiftlerden biri ölçüldüğünde diğer özelliğin ölçülmesi doğa kuralları gereği mümkün olmadığı için fotondaki bilginin tek bir ölçümle elde edilmesi mümkün değildir. Her polarizasyon tabanı için ayrı ayrı Kalsit kristali üzerinden ölçüm yapılması gerekir.

Şekil 3.7'de kenarsal ve diagonal olarak polarize edilmiş bir fotonun uygun ve uygun olmayan polarizasyon ölçücülerle ölçülmesi durumunda ortaya çıkan sonuçlar bir örnek üzerinden gösterilmeye çalışılmıştır. Şekil 3.7.a'da kenarsal (dikey) olarak polarize edilmiş bir foton polarizasyon ölçücüye gönderilmektedir. Polarizasyon ölçücü de kenarsal polarizasyona sahip olduğu için foton bu ölçücüden geçebilecektir. Ölçücünün arkasına koyulacak bir detektörle de fotonun taşıdığı bilgi okunabilecektir ve foton bu aşamadan sonra yok olacaktır. Şekil 3.7.b'de ise diagonal (45^o) olarak polarize edilmiş bir foton kenarsal olarak ayarlanmış bir polarizasyon ölçücüye gönderilmektedir. Bu ölçücünün çıkışında eşit olasılıklı olarak dikey ve yatay polarizasyona sahip iki foton ortaya çıkmaktadır. Diğer bir deyişle, hatalı (rasgele) bir ölçüm yapılmış olur. Yapılan bu hatalı (rasgele) ölçüm fotonun orijinal polarizasyonunu bozacaktır (belirsiz hale getirecektir) ve bu noktadan sonra fotonun gerçek polarizasyonunu ölçmenin imkanı yoktur.

Burada foton polarizasyonunun ölçülmesinde bahsedilen bu belirsizlik gizli bir anahtarın dağıtımında iletişim hattını dinleyen bir saldırganın olup olmadığını anlamak için kullanılabilir.

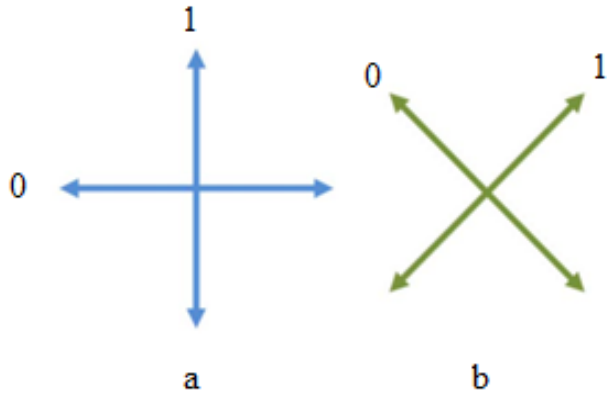


Şekil 3.7. Fotonların polarizasyonlarının ölçümü: a. Doğru ölçücü kullanımı, b. Yanlış ölçücü kullanımı [80, 81]

İletim hattında fotona herhangi bir etki olmadığı ve iletim hattının da fotonun polarizasyonunu bozmadığı (gürültüsüz olduğu) kabulü altında, eğer alıcı uca bir foton bozulmuş olarak geliyorsa iletim hattı boyunca fotona bir müdahale olduğu anlaşılacaktır. Bu da hattı dinleyen bir saldırganın varlığı hakkında bilgi verecektir.

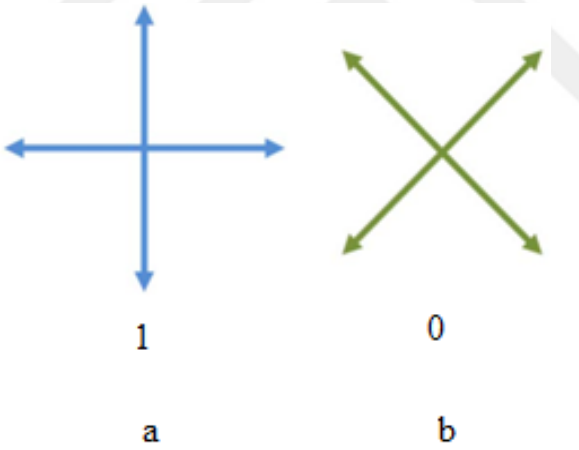
3.3. Fotonlarla Anahtar Dağıtımı

Farklı açılarda foton üretiminin mümkün olmasıyla birlikte, örn. foton tabancaları, belirli bir polarizasyona sahip foton üretimi de sağlanabilmektedir. Böylece, fotonlar üzerinde bilgi taşımak da mümkün hale gelmektedir. Örneğin, kenarsal ve diagonal polarizasyon tipleri kullanılarak fotonlar üzerinden ikili bilgi (0 ve 1) iletimi de mümkün olabilmektedir. 0 bitini temsil etmek için 0^o kenarsal (yatay) ve 135^o diagonal tabanlı polarizasyona sahip bir foton, 1 bitini/değerini temsil etmek için 90^o kenarsal (dikey) ve 45^o diagonal tabanlı polarizasyona sahip bir foton kullanılabilir. Bu durum Şekil 3.8'de de gösterilmiştir.



Şekil 3.8. {0, 1} bitlerinin farklı şekilde polarize edilmiş fotonlarla temsil edilmesi: a. Kenarsal polarizasyonlar, b. Diagonal polarizasyonlar

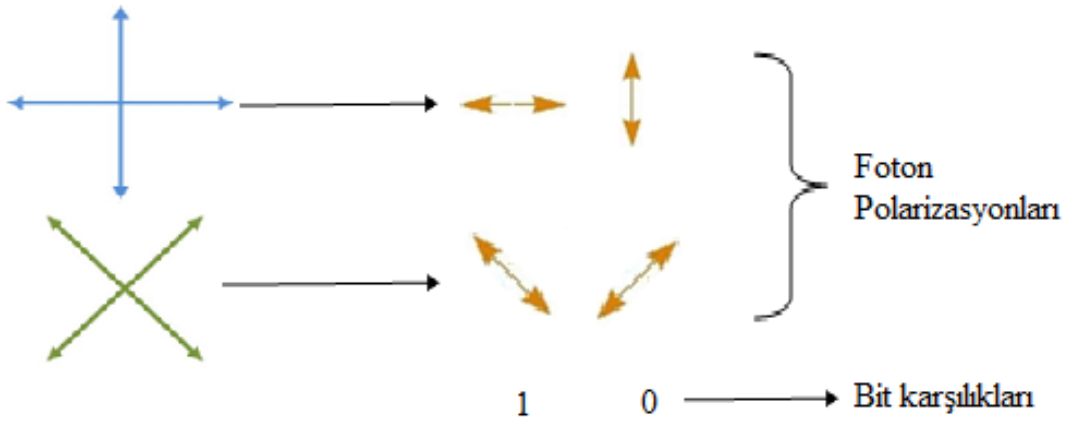
Bir bit dizisinde her bir biti kodlamak maksadıyla kullanılacak olan foton polarizasyonunu belirlemek için polarizasyon tabanları ile bitleri eşleştiren kodlama kuralları da tanımlanabilmektedir. Bu durum Şekil 3.9'da gösterilmiştir.



Şekil 3.9. Polarizasyon tabanlarının ikili bitlerle temsili: a. Kenarsal taban, b. Diagonal taban

Bir mesaj ikili bir bit dizisine dönüştürüldüğünde, bu bitleri kodlamak için kullanılacak fotonların polarizasyonlarını belirlemek gerekmektedir. Bu amaçla, polarize edicide mesajdaki bitlerle aynı uzunlukta rasgele polarizasyon taban dizisi üretilir. Böylece, bitlere karşı düşecek fotonların hangi polarizasyon tabanıyla polarize edileceğine rasgele olarak karar verilmiş olur. Burada bahsi geçen rasgelelik için güvenliği maksimum düzeyde sağlamak adına kuantum mekaniğine dayanan rasgelelik yöntemleri kullanılmaktadır.

Bahsi geçen yöntemlerden yola çıkılarak, gizli bir anahtarın aralarında belirli bir mesafe bulunan gönderici ve alıcı arasında güvenli olarak nasıl dağıtılacağı problemi için bazı fikirler ileri sürülmektedir. Diğer bir deyişle, iki uç birim arasında güvenli gizli anahtar dağıtımı için kuantum mekaniği kullanılabilir. Bu durum halihazırda kuantum kriptografi konularında incelenmiştir ve yukarıda anlatıldığı gibi fotonların iletilebileceği herkese açık bir kanal üzerinden polarize edilmiş fotonlarla sağlanabilmektedir. KAD problemi olarak da geçen bu problemde gizli anahtarın dağıtımı amacıyla ikili bitlerden oluşan rasgele bir mesajdaki her bir bit için gönderici tarafından belirli bir polarizasyon açısına sahip fotonlar üretilir ve alıcıya bu fotonlar gönderilir. Tüm bitler için bunlara karşılık gelen tüm fotonlar alıcıya gönderildikten sonra, alıcı tarafında foton dedektörleri ile bit dizileri elde edilir ve gizli anahtar bazı yöntemlerle, örn. CASCADE protokolü, bu bitlerden oluşturulur. Alıcı tarafta dedektör ile ölçülen fotonlara tekabül eden bitlere karar vermek için, bu işlem öncesinde, gönderici ve alıcının aynı kodlama kurallarında anlaşmış olması gerekmektedir. Bu nedenle, gönderici kodlama kuralını klasik kanal üzerinden alıcıya iletir. Böylece, alıcı ölçtüğü foton için 0 veya 1 kararını verebilecektir. Şekil 3.10'da da gösterilen bu kodlama kuralı herkese açık olabilir.



Şekil 3.10. Bitleri foton polarizasyon durumları ile kodlama kuralları

Bahsi geçen durum bir örnek üzerinden açıklanabilir. Örneğin, gönderici ve alıcı yapacakları haberleşmeyi güvenli kılmak adına sadece kendilerinin sahip olacağı gizli bir anahtar elde etmek istiyor olsun. Öyle bir yöntem bulunmalı ki, örn. KAD yöntemi gibi, bu yöntemle doğrudan orijinal bitleri alıcıya göndermek yerine üretilen rasgele bir bit dizisini göndererek hem gönderici hem de alıcı tarafında belirtilen bu gizli anahtara ulaşılabilmesi sağlanabilsin. Bunun için ilk iş, ikili bitleri foton

polarizasyonları olarak alıcıya göndermek amacıyla kodlama kurallarının tanımlanmasıdır. Şekil 3.10'da kodlama kuralına bir örnek gösterilmiştir. Bu kodlama kuralı, kuantum iletişime başlamadan önce klasik bir kanal üzerinden göndericiden alıcıya iletilir. Bu sayede, alıcı rasgele üreteceği polarizasyon tabanlarını hangi taban listesinden seçeceğini bilebilecektir. Şekil 3.11'de verilen örnek için, gönderici ve alıcı $\{+, X\}$ gibi bir polarizasyon taban listesi üzerinde anlaşır ve iki taraf da kullanacakları polarizasyon tabanlarını bu listeden rasgele olarak seçer.

KAD yönteminde, kodlama kuralları tanımlandıktan sonra ilk olarak gönderici kuantum rasgele sayı üreteci ile rasgele bir bit dizisi üretir. Ancak, alıcının bu bit dizisinden hiçbir şekilde haberi yoktur. Ardından, gönderici bitleri Şekil 3.10'da gösterilen kodlama kuralına göre kodlar (uygun polarizasyonda fotonları üretir) ve bunları alıcıya gönderir. Alıcı gelen fotonun polarizasyonunu ölçer, ancak göndericinin fotonu gönderirken kullandığı polarizasyon tabanını bilmez. Bu sebeple, alıcı taraf da kodlama kuralı uyarınca rasgele polarizasyon tabanları üretir. Kuantum mekaniğinde rasgelelik gerektiren noktalarda güvenliği en üst noktaya çekmek adına kuantum rasgele sayı üreteci kullanılabilir. Gönderici ve alıcı tarafta aynı polarizasyon tabanına sahip bitler kesinlikle aynı olur ve bu bitler gizli anahtar olarak kullanılabilir. Bu nedenle foton iletişiminden sonra, gönderici ve alıcının gizli anahtar üzerinde anlaşabilmesi için aynı polarizasyon tabanına sahip bitleri seçmeleri gerekmektedir. Bundan dolayı, gönderici ve alıcı birbirlerine rasgele ürettikleri polarizasyon tabanlarını da klasik kanaldan gönderirler. Gönderici ve alıcı polarizasyon tabanları üzerinde anlaşmak için farklı yollar da deneyebilir, örn. telefon üzerinden. Şekil 3.11'de de gösterildiği gibi aynı polarizasyon tabanına sahip bitler gizli anahtar olarak seçilir. Bu örnekte, kuantum kanalın gürültüsüz olduğu ve hattı dinleyen herhangi bir saldırgan olmadığı ideal bir durum ele alınmıştır. Diğer bir deyişle, foton iletişimi sırasında fotonu bozabilecek herhangi bir etkinin olmadığı kabul edilmiştir. Ancak, pratikte durum böyle olmamaktadır. Kuantum kanal gürültülüdür ve fotonlarda % 4'e kadar bir bozulmaya neden olmaktadır. Ayrıca, hattı dinleyen saldırganlardan da kaynaklı foton bozulmaları olabilmektedir ve hata oranı %4'ün üzerine çıkmakta ve daha önce de ifade edildiği gibi % 27,6'lara kadar çıkabilmektedir [8]. Bu durumda, gizli anahtarın elde edilmesi bu adımda sonlandırılamaz. Bu işlem sonucunda elde edilen bit dizisi, gizli anahtar adayı, üzerinde HSD teknikleri uygulanarak hatalardan

arındırılmış gizli anahtar elde edilir. Bu senaryo, bir sonraki bölümde Şekil 3.12'de incelenmiştir.

Göndericinin Bitleri	0	1	1	0	1	0	0	1
Göndericinin Tabanları	+	+	X	+	X	X	X	+
Göndericinin Polarizasyonları	↑	→	↖	↑	↖	↗	↗	→
Alıcının Tabanları	+	X	X	X	+	X	+	+
Alıcının ölçümleri	↑	↗	↖	↗	→	↗	→	→
Gizli Anahtar	0	0	1	0	1	0	1	1

Şekil 3.11. Tarafların ortak bir gizli anahtar üzerinde anlaşması (İdeal durum)

Şekil 3.11'de gönderici ve alıcının gizli bir anahtar üzerinde anlaşması basit bir örnek üzerinden anlatılmaktadır. Tarafların Şekil 3.10'daki kodlama kuralı üzerinde anlaştığı ve kuantum kanalın gürültüsüz/saldırgansız olduğu kabul edilmektedir. Bu örnek aşağıdaki gibi bir akıştan oluşmaktadır:

- Gönderici rasgele bir bit dizisi üretir,
- Gönderici bu bit dizisindeki her bir bit için yine rasgele bir polarizasyon tabanı üretir,
- Gönderici her taban için kodlama kuralına göre uygun polarizasyonda bir foton üretip alıcıya kuantum kanaldan gönderir,
- Alıcı her foton için rasgele bir polarizasyon tabanı üretir,
- Alıcı kendisine ulaşan fotonları polarizasyon tabanları vasıtasıyla ölçer,
- Alıcı kodlama kuralına göre ölçmüş olduğu fotonlara karşılık gelen bitlere karar verir,
- Taraflar rasgele ürettikleri polarizasyon tabanlarını klasik kanal üzerinden birbirlerine söylerler,
- İki tarafın da aynı polarizasyon tabanını kullanmış olduğu bitler gizli anahtarı oluşturur.

Şekil 3.11'de bazı bitler kırmızı kutular içerisinde gösterilmiştir. Bu bitlerden bazıları doğru ölçülmüş olmasına rağmen gizli anahtar seçiminde kullanılmamıştır. Bu durum polarizasyon tabanlarının rasgele üretilmesinden kaynaklanmaktadır ve gizli anahtar

seçiminde kullanılamaz. Daha önce de bahsedildiği gibi, gönderici ve alıcı rasgele ürettikleri polarizasyon tabanlarını birbirleriyle paylaşırlar ve aynı polarizasyon tabanlarına karşılık düşen bitleri gizli anahtar olarak seçerler, farklı tabanların kullanıldığı bitler dikkate alınmazlar. Kuantum kanalda herhangi bir bozucu etki, örn. kanal gürültüsü ve/veya hattı dinleyen saldırgan, olmadığı durumda bu bitlerin tamamen eşit olduğu gözlemlenmektedir. Ancak, pratik uygulamalarda kuantum kanal gürültülüdür ve hattı dinleyen saldırgandan kaynaklanan etkiler de olabilmektedir. Bu etkilerin tamamı kuantum kanal gürültüsü olarak kabul edilip ε kanal hata olasılığı ile temsil edilmektedir.

Bahsi geçen örnekte gönderici belirli bir uzunlukta rasgele bir bit dizisi üretmiştir ve gönderici ve alıcı da neredeyse bunun yarısı kadar uzunluğa sahip bir bit dizisini gizli anahtar olarak kabul etmiştir. Olasılıksal olarak, gönderilen herhangi bir foton için alıcının doğru tabanı (+ veya X tabanından birini) seçmesi olasılığı 0,5 olacaktır. Diğer bir deyişle, alıcı tarafında rasgele olarak üretilen polarizasyon tabanları % 50 ihtimalle göndericinin polarizasyon tabanlarıyla aynı olacaktır [91].

3.4. Hattı Dinleyen Saldırganların Tespit Edilmesi

Bu bölümde, yukarıda bahsi geçen örnek hattı dinleyen bir saldırganın olması durumu için yeniden ele alınacaktır. Yukarıda bahsi geçen kurallar hattı dinleyen biri(leri)nin olduğu durumda gönderici, saldırgan ve alıcıdaki bitlerin durumunun nasıl olacağı incelenecektir.

Göndericiden çıkan fotonlar alıcıya ulaşmadan hattı dinleyen saldırgan bu fotonlar üzerinde ölçümler yapar. Yapmış olduğu ölçümler sonucunda bir bit dizisi elde eder. Ancak, Bölüm 3.2'de de anlatıldığı gibi foton üzerinde yapılan bu ölçümler sonrasında fotonlar yok olur. Diğer bir deyişle, alıcıya hiçbir foton ulaşmaz. Bunu önlemek için saldırgan göndericiden gelen fotonları aynı şekilde oluşturmaya çalışacaktır ve üreteceği bu yeni foton dizisini alıcıya gönderecektir. Ancak, saldırgan da fotonlarda ölçüm yapmak için polarizasyon tabanlarını Şekil 3.11'deki örnekte de anlatıldığı gibi rasgele olarak üretecektir. Rasgelelikten ötürü bazı bitlerde hatalar oluşacaktır. Hatalı bitler için üretilen yeni foton polarizasyonları da hatalı olacaktır. Alıcı kendisine gelen bu fotonlara saldırganın da uyguladığı işlemleri uygular; polarizasyon tabanlarını

rasgele oluşturur ve bit dizisini elde eder. Mesajdaki bir bitin saldırgandan kaynaklanan sebeplerle bozulması durumu Şekil 3.12'deki gibi örneklendirilmiştir.

Göndericinin Bitleri	0	1	1	0	1	0	0	1
Göndericinin Tabanları	+	+	X	+	X	X	X	+
Göndericinin Polarizasyonları	↑	→	↘	↑	↘	↗	↗	→
Saldırganın Ürettiği Tabanlar	+	X	+	+	X	+	X	+
Saldırganın Polarizasyonları	↑	↗	→	↑	↘	→	↗	→
Alıcının Ürettiği Tabanlar	+	X	X	X	+	X	+	+
Alıcının Polarizasyonları	↑	↗	↗	↘	→	↗	↑	→
Gizli anahtar	0	0	0	1	1	0	0	1
Anahtardaki Hatalar	✓		✘			✓		✓

Şekil 3.12. Hattı dinleyen saldırgan(lar)ın tespit edilmesi (İdeal olmayan durum)

Şekil 3.12'de de gösterildiği gibi, arada saldırganın olması durumunda iletişimin nasıl olduğu aşağıdaki gibi açıklanabilir:

- Gönderici rasgele bir bit dizisi üretir,
- Gönderici bu bit dizisindeki her bir bit için yine rasgele bir polarizasyon tabanı üretir,
- Gönderici her taban için kodlama kuralına göre uygun polarizasyonda bir foton üretip alıcıya kuantum kanaldan gönderir,
- Saldırgan hattı dinlemektedir,
- Saldırgan her foton için rasgele bir polarizasyon tabanı üretir,
- Saldırgan kendisine ulaşan fotonları polarizasyon tabanları vasıtasıyla ölçer. Ölçülen fotonlar doğası gereği bir daha ölçülemez duruma gelir ve yok olur,
- Saldırgan kodlama kuralına göre ölçmüş olduğu fotonlara karşılık gelen bitlere karar verir. Polarizasyon tabanlarını rasgele ürettiği için göndericinin ürettiği orijinal bit dizisine sahip olamayacaktır (Ayrıca, kuantum kanaldaki gürültüden ötürü zaten bazı fotonlar da saldırgana hatalı olarak gelecektir. Ancak, basitlik için örnekte bu durum dikkate alınmamıştır.). Diğer bir deyişle, saldırganın hattı dinliyor olması, bit dizisini elde etmesi için yeterli olmamaktadır,

- Saldırgan ölçtüğü fotonların yok olduğunu bildiği için ve alıcı kendisinin farkına varmaması için yeni foton dizisi üretir,
- Saldırgan elindeki her bir bit için rasgele bir polarizasyon tabanı üretir,
- Saldırgan kodlama kuralına uygun fotonu üretir ve alıcıya gönderir,
- Alıcı her foton için rasgele bir polarizasyon tabanı üretir,
- Alıcı kendisine ulaşan fotonları polarizasyon tabanları vasıtasıyla ölçer,
- Alıcı kodlama kuralına göre ölçmüş olduğu fotonlara karşılık gelen bitlere karar verir,
- Gönderici ve alıcı rasgele ürettikleri polarizasyon tabanlarını klasik kanal üzerinden birbirlerine açıklarlar,
- İki tarafın da aynı polarizasyon tabanını kullanmış olduğu bitler kesinlikle aynı olmalıdır (burada da benzer şekilde, kırmızı kutuda gösterilen bitler, kendileri için rasgele üretilen polarizasyon tabanları farklı olduğu için dikkate alınmazlar),
- Saldırgan tespiti için, alıcı ürettiği bit dizisinden gönderici ile aynı polarizasyon tabanlarını kullandığı bitlerin bir alt kümesini rasgele olarak seçer,
- Alıcı bu bitlerin değerlerini göndericideki değerlerle karşılaştırır,
- Alıcı hatalı bitin olup olmadığını kontrol eder ve varsa, hatalı bit oranını hesaplar.

Aynı polarizasyon tabanı için ölçülen bit değeri farklı ise bu fotonun kanalda bozulduğu kesin olarak söylenebilmektedir. Eğer kuantum kanal gürültüsüz olsaydı, bu bozulmanın hattı dinleyen saldırgandan kaynaklandığı kesin olarak söylenebilirdi. Ancak, pratikte kuantum kanal da gürültülüdür. Bu nedenle, fotondaki bozulmalar saldırgan ya da kanalın kendisinden kaynaklanmış olabilir. Bu nedenle, pratik uygulamalarda, gönderici ve alıcı hatalı bitlerin olmasına rağmen hatalı bit sayısının belirli bir hata oranının altında olması durumunda iletişime devam da edebilmektedirler. Bölüm 1.1'de de bahsedildiği gibi sadece kuantum kanaldan kaynaklanan hata miktarı % 1-4 arasında değişmektedir. Hattı dinleyen saldırgan olması durumunda ise, bu oran artacaktır. Eğer alıcı seçmiş olduğu bitler içerisinde %4'ten daha fazla hatalı bit tespit ederse, bu durum hattı dinleyen saldırganın varlığına işaret eder. Bu durumda, gönderici ve alıcı isterlerse iletişimi sonlandırıp, daha sonraki bir zamanda yeniden deneyebilirler.

Ancak, KAD uygulamalarında, hatalı bit olsa bile belirli sınırlar içerisinde kaldığı sürece iletişime devam edilir. Modern BU protokolleri, en son güvenlik ispatlarına göre % 30'lara kadar olan hata miktarları için başarılı çalışmaktadır [26]. Ancak, bu orandan daha yüksek hata oranlarında saldırganın gizli anahtarı elde etme ihtimali çok yükseleceği için iletişimin yapılmaması önerilmektedir. Alıcının rasgele ürettiği bit dizisinden saldırganı tespit etmek maksadıyla seçmiş olduğu bitlerin değerleri de klasik kanaldan gönderildiği için bu bitler gizli anahtar olarak kullanılmaz. Gizli anahtar için geriye kalan bitler kullanılır. Kuantum kanal da gürültülü olduğu için gizli anahtar olarak kullanılacak bu kalan bitler üzerinde de mutlaka HSD işlemleri, örn. CASCADE, uygulanmalıdır.

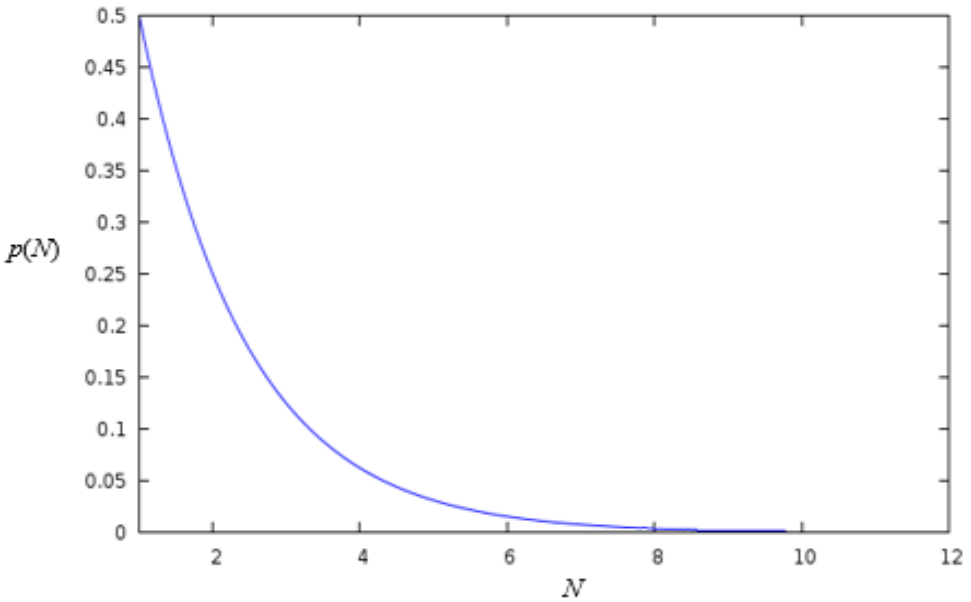
Hattı dinleyen bir saldırganın yeniden ürettiği (orijinaliyle birebir aynı değil) fotonların alıcıya gelmesi sonucu alıcıda oluşan gizli anahtar Şekil 3.13'te gösterilmiştir. Bu şekil, Şekil 3.12'deki senaryonun sonucunu kısaca göstermek amacıyla oluşturulmuştur. Şekilde kırmızı kutu içinde gönderici ve alıcıdaki polarizasyon tabanları gösterilmiştir. Koyu kırmızı ile işaretlenen bitler ise, aynı polarizasyon tabanı için elde edilen bitlerin farklı olduğunu göstermektedir. Göndericinin gönderdiği fotonlarla alıcıya gelen fotonlar farklıdır. Bunun sebebi hattı dinleyen ve hattaki fotonlara müdahale eden saldırganlardır. Örneğin, göndericinin soldan sağa doğru üçüncü biti olan 1 değeri için gönderdiği diagonal polarizasyonlu foton alıcıya yatay polarizasyon olarak gelmiştir. Alıcı bu foton için diagonal bir polarizasyon tabanı ürettiğinden ötürü (rasgele) 0 bitini elde etmiştir. Eğer göndericinin gönderdiği diagonal polarizasyonlu foton saldırganın müdahalesi olmadan alıcıya gelmiş olsa ve alıcı da yine aynı polarizasyon tabanını üretmiş olsa kesinlikle doğru olan 1 biti elde edilecekti. Göndericinin gönderdiği altıncı bit için de benzer durum söz konusudur. Ancak, alıcı tarafında üretilen polarizasyon tabanındaki rasgelelik yüzünden şans eseri bu bit doğru bir şekilde elde edilmiştir. Diğer bitlerde oluşan hatalar da yine polarizasyon tabanları rasgele üretildiği için polarizasyon tabanlarındaki farklılıklardan kaynaklanmaktadır. Bölüm 3.3'te de anlatıldığı gibi bu işlemlerde % 50 ihtimalle yanlış bitler oluşuyordu. Özetle, aynı polarizasyon tabanına sahip bitlerin tamamının aynı olması gerekmektedir. Aksi halde, fotonların polarizasyonlarında farklılıklar oluştuğu aşikar olmaktadır. Diğer bir deyişle, kuantum

kanaldan kaynaklanan bir bozulma yoksa, arada saldırgan(lar)ın olduğu kesin olarak söylenebilmektedir.

Göndericinin Bitleri	0	1	1	0	1	0	0	1
Göndericinin Tabanları	+	+	X	+	X	X	X	+
Göndericinin Polarizasyonları	↑	→	↘	↑	↘	↗	↗	→
Alıcıya Gelen Polarizasyonlar (Saldırgandan)	↑	↗	→	↑	↘	→	↗	→
Alıcının Ürettiği Tabanlar	+	X	X	X	+	X	+	+
Alıcının Polarizasyonları	↑	↗	↗	↘	→	↗	↑	→
Gizli anahtar	0		0			0		1
Anahtardaki Hatalar	✓		✗			✓		✓

Şekil 3.13. Hattı dinleyen saldırgan(lar)ın tespit edilmesi

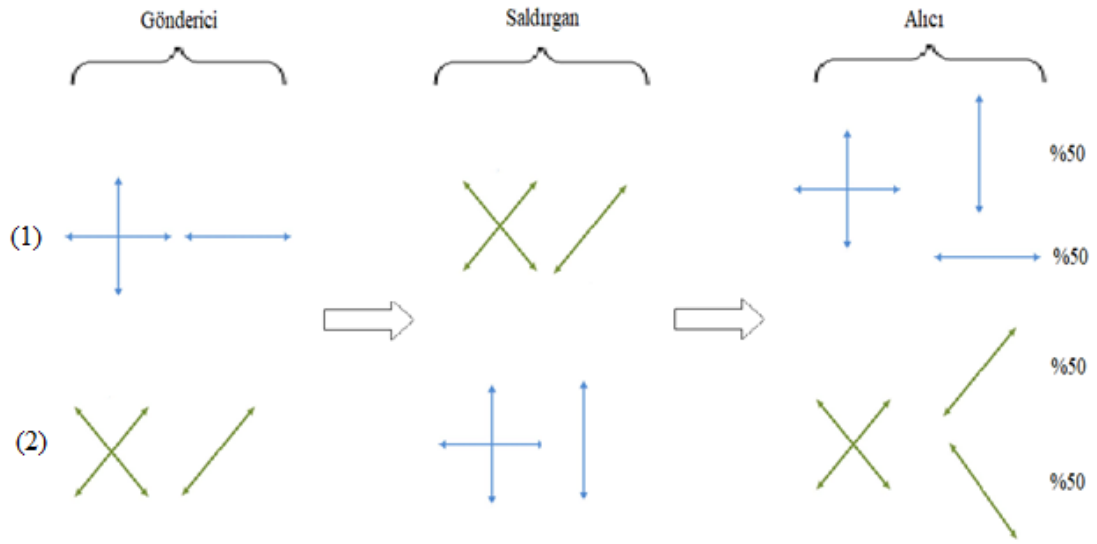
Hattı dinleyen saldırganın kendini gizleyebilmesi için kendisine gelen fotonları aynı şekilde alıcıya göndermesi gerekmektedir. Bunu yapabilmesi için de göndericinin rasgele olarak ürettiği polarizasyon tabanlarını bilmesi gerekmektedir. Bunu da bilemeyeceği için tahmin etmeye çalışacaktır. Tek bir taban için doğru tahmini yapabilmesinin olasılığı 0,5'tir. Rasgele üretilen polarizasyon tabanlarının sayısı N olmak üzere saldırganın tahmin etme olasılığı ise $(0,5)^N$ olacaktır. N ile tahmin etme olasılığı arasındaki ilişki Şekil 3.14'te gösterildiği gibidir.



Şekil 3.14. Saldırganın gönderici ile aynı tabanları seçme olasılığı

Şekil 3.14'ten de görüldüğü gibi, N ≥ 10'dan itibaren saldırganın göndericinin polarizasyon tabanlarıyla aynı polarizasyon tabanlarını tahmin etme ihtimali neredeyse sıfırdır. Ancak, daha küçük N'ler için saldırganla göndericinin ürettiği polarizasyon tabanlarının eşit olma ihtimali gerçekten de olabilir. Bu durumda göndericiden gelen mesajı saldırgan ele geçirmiş olur. Saldırgan, alıcıya da aynı fotonları gönderdiği için gönderici ve alıcı aradaki saldırgandan haberdar olamaz.

Saldırgan farklı polarizasyon tabanları seçmesine rağmen, şans eseri alıcı ile gönderici aynı polarizasyon tabanlarını üretebilir ve bunlara karşı düşen bitler de aynı olabilir (bütün süreçteki rasgelelik, saldırganın bozduğu bilgileri düzeltebilir). Şekil 3.15'te saldırganın göndericiden farklı polarizasyon tabanları seçtiği durumda neler olabileceği örneklenmektedir.



Şekil 3.15. Saldırganın tespit edilememe durumları

Şekil 3.15'te hattı dinleyen saldırganın tespit edilemediği durumlar gösterilmektedir. Şekil 3.15'te gösterilen birinci örnekte, göndericinin kenarsal (+), saldırganın diagonal (X) ve alıcının da kenarsal (+) polarizasyon tabanlarını üretmesi durumunda alıcıda ölçülen fotonların durumları gösterilmiştir. Bu örnekte, gönderici kenarsal bir polarizasyon tabanı kullanarak yatay polarizasyona sahip bir foton üretmiştir. Saldırgan ise, diagonal bir polarizasyon tabanı için 45^{o} açıdaki polarizasyona sahip bir foton üretmiştir. Alıcı ise, yine kenarsal bir polarizasyon tabanını rasgele olarak seçmiş ve buna karşılık kendisine gelen 45^{o} açıdaki polarizasyona sahip bir foton için % 50 ihtimalle yatay ve yine % 50 ihtimalle dikey polarizasyona sahip bir foton elde eder.

İkinci örnekte ise, göndericinin diagonal (X), saldırganın kenarsal (+) ve alıcının da diagonal (X) polarizasyon tabanlarını üretmesi durumunda alıcıda ölçülen fotonların durumları gösterilmiştir. Bu örnekte, gönderici, diagonal bir polarizasyon tabanı kullanarak 45^o açıdaki polarizasyona sahip bir foton üretmiştir. Saldırgan ise, kenarsal bir polarizasyon tabanı için dikey polarizasyona sahip bir foton üretmiştir. Alıcı ise, yine diagonal bir polarizasyon tabanını rasgele olarak seçmiş ve buna karşılık kendisine gelen dikey polarizasyona sahip bir foton için % 50 ihtimalle 45^o ve yine %50 ihtimalle 135^o açıdaki polarizasyona sahip bir foton elde eder. Şekil 3.15'ten de görülebileceği gibi arada saldırgan olması durumunda dahi % 50 ihtimalle göndericinin fotonları alıcıya ulaşabilmektedir. Bu durum saldırgan açısından şanslı bir durumdur ve saldırganın varlığı farkedilemez [80, 81]. Yine şekilden görülebileceği üzere % 50 ihtimalle göndericinin bitleri alıcıdaki bitlerden farklı olacaktır. Bu durumda hattı dinleyen bir saldırganın olduğu söylenebilecektir [80, 81].

Buradan çıkarılabilecek sonuç; gönderici ve alıcı gizli anahtarı belirlerken ne kadar fazla biti kontrol ederse hattı dinleyen bir saldırganın varlığının anlaşılması da o kadar kolay olacaktır. Bir foton için saldırganın hatalı bir polarizasyon tabanı seçme ihtimali 0,5 olur. Alıcının da yanlış biti elde etme olasılığı 0,5 olacaktır. Bir bit için saldırganın varlığının anlaşılması olasılığı da aşağıdaki gibi hesaplanır,

$$p_s = \frac{1}{2}.\frac{1}{2} = \frac{1}{4} \tag{3.1}$$

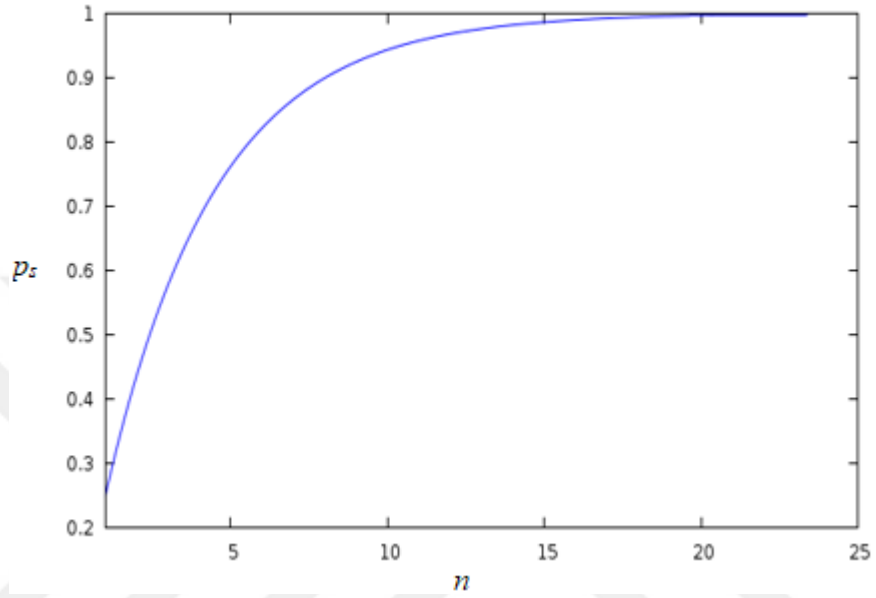
Gönderilen bir bitin saldırganı açığa çıkarmama olasılığı ise p_s'nin tümleyeni yani $3/4$ olmaktadır.

$$p'_s = 1 - p_s = \frac{3}{4} \tag{3.2}$$

Gönderici ve alıcı saldırganı tespit etmek amacıyla n adet test biti kullandığı için bu olasılık $p'_s = \left(\frac{3}{4}\right)^n$ olacaktır. Bu durumda, n adet test biti kullanılarak hattı dinleyen saldırganın ortaya çıkarılma olasılığı ise aşağıdaki gibi hesaplanabilir,

$$p_s = 1 - \left(\frac{3}{4}\right)^n \tag{3.3}$$

Seçilen n adet test bitine karşılık hattı dinleyen saldırganın ortaya çıkarılma olasılığının değişimi Şekil 3.16'da gösterildiği gibidir.



Şekil 3.16. Saldırganın varlığının ortaya çıkarılması için test edilen bit sayısı ile tespit etme olasılığının değişimi

Şekil 3.16'dan da görülebileceği üzere n test biti sayısı ne kadar fazla olursa saldırganın tespit edilme olasılığı o kadar yüksek olmaktadır ($n \geq 20$ için neredeyse %100 tespit edilebiliyor). $n = 20$ için saldırganın ortaya çıkarılması olasılığı % 99,68 iken, $n = 30$ olduğunda bu oran % 99,98 olmaktadır. Sonuç olarak, gönderici ve alıcı sadece test ettikleri bitlerin sayısını arttırarak hattı dinleyen bir saldırganın olup olmadığını kolayca anlayabilirler.

Gönderici ve alıcı, bu işlemler sonucunda, hattı dinleyen bir saldırgan olmadığına karar verirse gizli bir anahtar üzerinde anlaşmış olacaklardır. Saldırganı tespit etmek amacıyla bazı bitler test biti olarak kullanıldığı için, ortaya çıkan gizli anahtar uzunluğu daha kısa olacaktır [91].

3.5. Pratikte Kullanım Açısından Kuantum Anahtar Dağıtımı

BB84 protokolünden sonra foton polarizasyonu yerine alternatif çözümler sunan başka çalışmalar da olmuştur. Bu çalışmalar da kuantum mekaniğindeki diğer kuantum

özelliklerden faydalanmaktadırlar. Örneğin, [63]'deki çalışmada, fotonun aynı zamanda bir dalga olduğu gerçeği üzerinden yola çıkılarak fotonların polarizasyonu yerine faz bilgileri kullanılmıştır. Başka bir çalışmada ise [64], kuantum parçacık çiftlerinin dolaşıklık diye ifade edilen bir niteliğinden faydalanılmıştır. Bu iki kuantum parçacık çifti birlikte bir kuantum sistem olarak kabul edilir. Bunlardan herhangi biri üzerine uygulanan bir etki sonucu, diğeri de buna zıt yönde bir tepki gösterir. Bu durum aradaki mesafeden bağımsızdır ve bu özellik kuantum bilgi ışınlamasının temelini oluşturmaktadır. BB84 protokolünde olduğu gibi foton polarizasyonunu kullanan başka çalışmalar da olmuştur [65-76]. Bu çalışmalar da BB84'te kullanılan dört polarizasyon durumundan farklı olarak iki ve altı arasında değişen sayılarda polarizasyon durumları kullanırlar.

Birçok yöntemde olduğu gibi KAD'ın da kendisine has avantajları ve dezavantajları vardır. KAD ya da daha genel bir ifade ile kuantum kriptografi, kuantum mekaniğine dayalı fizik kurallarını kullandığı için teorik olarak güvenli bir haberleşme imkanı sunabilmektedir. Bazı dezavantajları ise, foton iletimi için gerekli ortamdan kaynaklı kısıtlar ile kuantum mekaniğini kullanan sistemlerin henüz yeterince olgunlaşmamış olması (örn. gerçekteki foton üreteçleri, kuantum kanal vb. elektronik/optik bilşenlerin teoride anlatıldığı gibi ideal olmamaları) gösterilebilir. Foton haberleşmesi foton iletimine uygun herhangi bir ortamdan sağlanabilir. Günümüzde bunun için genellikle fiber optik kablolar kullanılmaktadır. Fiber optik kablo mesafesi limiti yaklaşık olarak 400km kadardır (teknolojinin gelişmesiyle bu limitler artabilir). Optik bileşenlerin güvenli kullanımına yönelik kabul edilmiş uluslararası bir standart üzerinde çalışmalar devam etse de henüz yoktur.

Gönderici ve alıcı KAD yöntemi ile güvenli bir şekilde gizli anahtarı elde ettikten sonra, taraflar bilgi güvenliği amacıyla bu gizli anahtarı kullanabileceklerdir. İki tarafta da gizli anahtar mevcut olduğu için güçlü bir simetrik şifreleme algoritması kullanılarak bu işlem gerçekleştirilebilir. Literatürde birçok simetrik şifreleme algoritmaları mevcuttur. Vernam algoritması, Shannon tarafından kırılamazlığı ispatlanmış tek simetrik şifreleme algoritması olduğu bilinmesine rağmen anahtar yönetimindeki zorluk nedeniyle pek tercih edilmemiştir. Bunun yerine, bir diğer popüler simetrik şifreleme algoritması olan AES (Advanced Encryption Standard) [77, 78] pratik uygulamalarda yaygın olarak kullanılmaktadır. Ancak, günümüzde AES'in

kırılamayacağının ispatının yapılmadığına dikkat edilmelidir. Bu bölümde anlatılan KAD tekniği ile tüm bu simetrik şifreleme algoritmalarının ihtiyaç duyduğu gizli anahtarlar üretilebilmektedir. Bu gizli anahtarın Vernam simetrik şifreleme tekniği ile birlikte kullanılmasıyla da bir açıdan kesinlikle kırılamaz bir şifreli haberleşme olanağı da sağlanmış olmaktadır. KAD ya da daha genel bir ifadeyle kuantum kriptografi ile ilgili konular [79-83] çalışmalarında detaylı bir şekilde incelenmiştir.

4. OPTİMUM CASCADE PROTOKOLÜ

Bu bölümde öncelikle literatürde yaygın olarak ele alınmış CASCADE protokolleri tanıtılmıştır. Her yöntemin öne sürdüğü fikirler özet olarak sunulmuş, yöntemlerin bu çalışmada önerilecek yöntemden temel farkları açıklanmıştır. Daha sonra, bu tez çalışması kapsamında CASCADE protokolünde yapılabilecek iyileştirmeler anlatılmıştır [9]. Son olarak ise, iyileştirmelerin tamamını içeren ve optimum CASCADE adı verilen protokol incelenmiştir [9]. Bu inceleme sırasında yapılan her iyileştirmenin protokole katkısı deneysel olarak incelenmiş ve sonuçları grafik ve tablolar halinde sunulmuştur.

4.1. CASCADE Protokolleri

Veri paketlerinin iletimi sırasında değişik nedenlere bağlı olarak bazı bitler bozulabilir. Bu bozulmalara hata denir. Ortama bağlı olarak hataların oluşma olasılığı değişim gösterir. Örneğin optik hatlar üzerinde meydana gelebilecek hata oranı diğer ortamlardan çok daha azdır. Yıldırım ve benzeri atmosferik olaylar kablolu ve kablosuz ortamlarda hata çoğuşmalarının oluşmasına yol açarlar. Ortamdaki hatalarla ilgili olarak yapılması gereken ilk iş bu hataların sezilmesidir. Gerekiyorsa hatalı verinin düzeltilmesi ya da yeniden gönderilmesi üzerine çalışılabilir. Bazı ortamlarda, örneğin telsiz ortamlarda, hataların çoğuşma halinde oluştuğunu görebiliriz. Bu ortamlardaki hataların zamana göre homojen dağıldığını söylemek mümkün değildir. Çoğuşma halinde oluşmuş hatalar bir paketin içeriğini, çoğunlukla, orijinal veriyi tekrar oluşturamayacak şekilde bozarlar. Hata düzeltme teknikleri, hata sezme tekniklerinden daha karmaşıktır. Bu konular detaylı olarak [15]'te anlatılmaktadır. Bu tez çalışmasında, CASCADE protokolü kapsamında ele alınan hata sezme ve düzeltme teknikleri incelenecektir.

CASCADE protokolünün verimlilik performansının iyileştirilmesi maksadıyla önerilen literatürdeki yeni hata sezme ve düzeltme yöntemleri, ya orijinal CASCADE protokolünün dışına çıkmış ya da protokolün koyduğu sınırlar içinde kalarak protokolün kendisinin iyileştirilmesi gibi yöntemlere başvurmuşlardır.

Bu tez çalışmasında, orijinal CASCADE protokolünün dışına çıkmadan protokolün ana adımlarına önemli yenilikler katarak verimlilik performansı arttırılmaya çalışılmıştır. Bu maksatla, literatürdeki CASCADE teknikleri incelenmiş ve bu çalışmalarda önerilenlerden daha farklı özgün yaklaşımlar önerilmiştir. Literatürde orijinal CASCADE protokolünün verimlilik performansını iyileştirmek için genellikle şu konularda yoğunlaşılmıştır:

- Daha uygun parametre kümesi seçimi,
- Daha iyi karıştırıcıların kullanımı,
- Gereksiz/fazlalık parite kontrolünü azaltma,
- BINARY işlemini daha küçük bloklarda çalıştırma (geriye iz sürme aşamasında).

Bu tez çalışmasında, bunların hemen hemen hepsi dikkate alınarak daha verimli bir CASCADE protokolü sunulmaktadır. Bu ana kadar olan açıklamalardan da görülebileceği üzere CASCADE protokolünün çalışması sırasında yoğun olarak kullanılan iki temel adım vardır. Bunlar Parite kontrolü ve BINARY algoritmalarıdır. Parite kontrolü hata sezme maksadıyla kullanılır. BINARY ise bir blokta hata sezildiyse hatalı biti bulma ve düzeltme maksadıyla kullanılır.

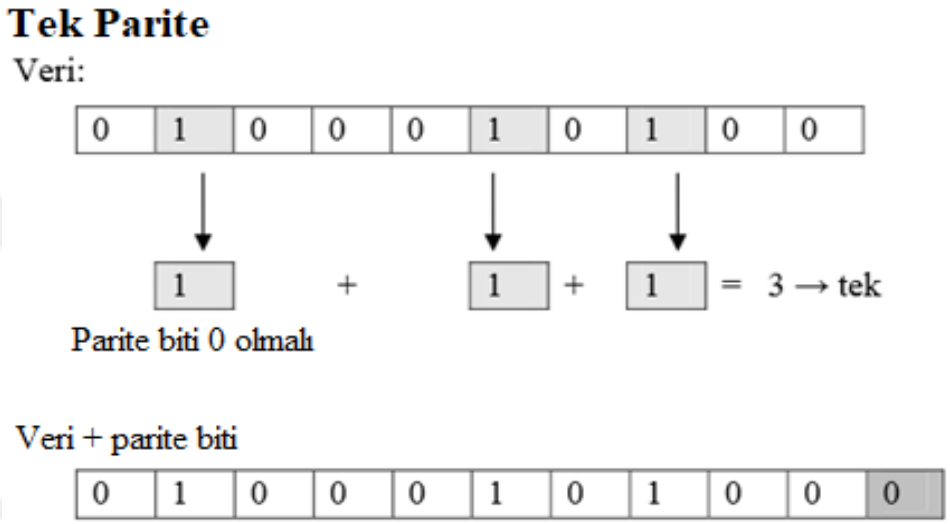
4.1.1. Parite kontrolü

Parite kontrolü, iletilen veride oluşan *tek* sayıda hatayı sezmek için kullanılır. Bir bit dizisinin parite bilgisi bitlerin birbirleriyle XOR'lanması ya da modülo iki tabanında toplanması vasıtasıyla hesaplanır.

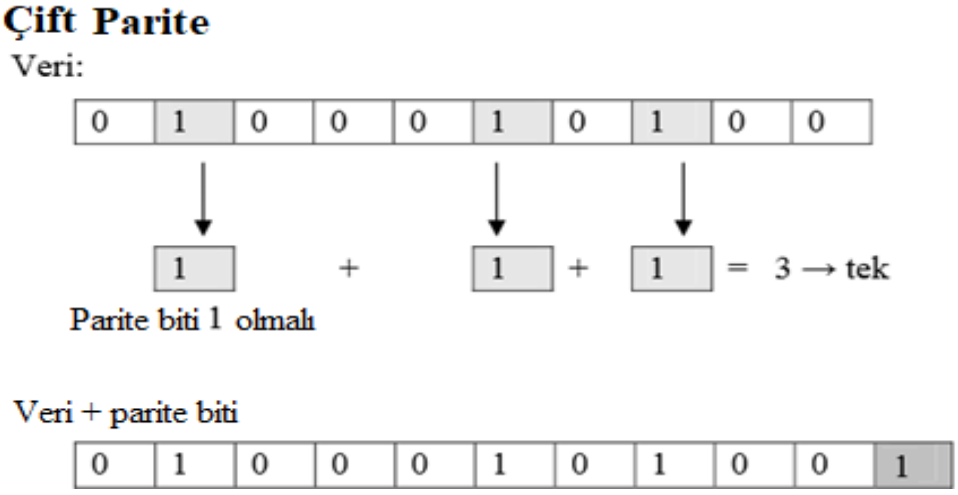
Klasik ağ haberleşmesinde amaç, verideki birlerin sayısını tek ya da çift olacak şekilde düzenlemektir. Bu amaçla veriye bir parite biti eklenir. Bit dizisine parite biti eklenerek bit dizisinin içindeki birlerin sayısının çift yada tek olması sağlanmaktadır. Birlerin sayısının çift olmasına çift parite, tek olmasına da tek parite durumu denilmektedir. Şekil 4.1 ve Şekil 4.2'de tek parite ve çift parite durumu için 0100010100 verisine eklenmesi gereken parite bitleri gösterilmiştir.

CASCADE protokolü sırasında protokolün çalışma prensibi gereği veri birçok bloğa ayrıştırılır. Bu blokların her biri için hata sezme işlemi gerçekleştirilir. CASCADE tabanlı yöntemler bu bit dizilerinde hata sezme maksadıyla çok basit ama etkili bir yöntem olan parite tabanlı kontroller yaparlar. Gönderici ve alıcı sahip oldukları bit

dizilerinin parite bilgilerini hesaplarlar ve bu bilgileri değiş tokuş ederek paritelerin uyuşup uyuşmadığını kontrol ederler. Daha önce de belirtildiği gibi parite kontrolü matematiksel doğası gereği tek sayıda hatalı biti tespit etmek için kullanılır. Parite değerinin sıfır olması o bit dizisinde hata olup olmadığı konusunda kesin bilgi vermez. Eğer bit dizisinde çift sayıda hatalı bit varsa, bu sezilemeyecektir. CASCADE protokolü o an için bu hatayı sezememiş olsa bile, protokolün ilerleyen adımlarında bu hatalı bit muhtemelen sezilebilecektir.



Şekil 4.1. Tek parite kodlanmış veri

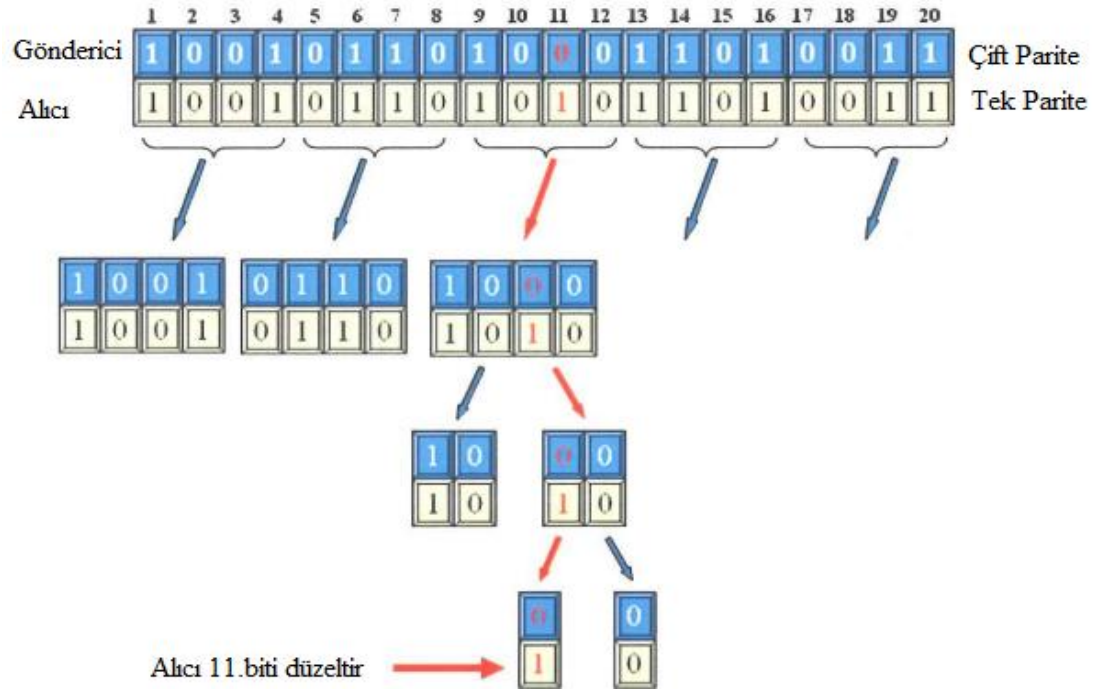


Şekil 4.2. Çift parite kodlanmış veri

4.1.2. BINARY tekniği

Göndericinin alıcıya gönderdiği orijinal mesaj X ile gösterilmek üzere, kuantum kanaldan, hattı dinleyen saldırganlardan ve diğer muhtemel etkilerden dolayı bazı bitlerde bozulmalar olur. Bu etkilerden kaynaklanan hatalar e ile gösterilmek üzere alıcıya gelen bit dizisi de Y ile gösterilmek üzere $Y = X + e$ olarak ifade edilir. Alıcı tarafında hatayı temsil eden e, CASCADE protokolü ile giderilerek $Y = X$ haline getirilmeye çalışılmaktadır. Bunun için alıcı tarafında, hata sezilen her blokta BINARY tekniği çalıştırılır ve hata düzeltilmiş olur.

BINARY tekniği, CASCADE protokolünün en önemli adımlarından biri olup hata düzeltme algoritması olarak bilinir. CASCADE protokolü çalıştırılırken, bir blokta hata sezildiğinde süreç o bloktaki hatanın bulunması ve düzeltilmesi akışına geçer. Bunun için, ilgili blok ikiye bölünür ve her parça için göndericiden parite bilgisi istenir. Paritelerin uyuşmadığı blok hatalı bit seviyesine inene kadar tekrarlı olarak ikiye bölünür. Hatalı bit bulunduğunda, bu biti düzeltmek için bitin değeri değiştirilir. Eğer protokolün ikinci ve daha sonraki raundlarında düzeltilen bir hata ise, geriye iz sürme süreci başlatılarak yeni hatalı bitler de düzeltilir. BINARY algoritmasının çalışma şeklini göstermek üzere Şekil 4.3'teki gibi bir örnek verilebilir.



Şekil 4.3. Hata düzeltme işlemi için BINARY'nin çalışma şekli

4.1.3. Orijinal CASCADE protokolü

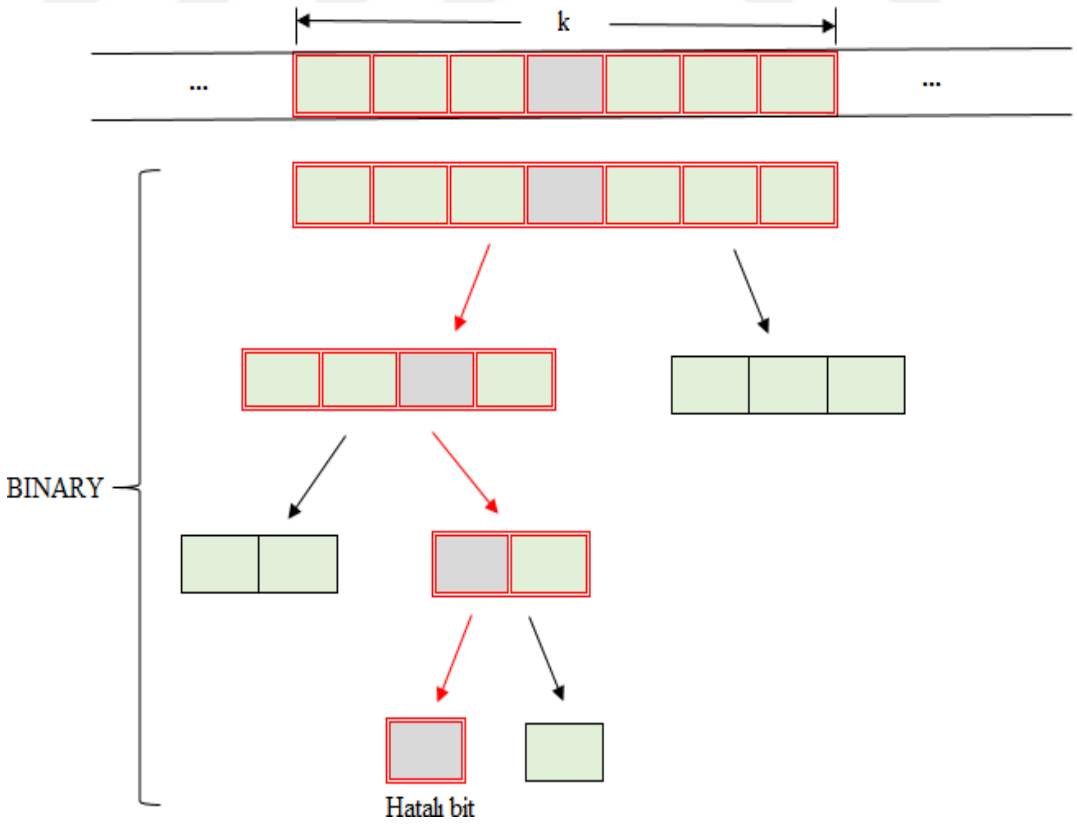
[16]'daki orijinal CASCADE protokolü, ilk olarak KAD sistemlerinin BU fazında kullanılması için tasarlanmış olan tamamen parite kontrolü tabanlı bir HSD yöntemidir. Daha önce de bahsedildiği gibi, bir bit dizisinin paritesi, dizideki tüm bitlerin XOR edilmesi ile elde edilen ikili bir değerdir, 0 ya da 1. Eğer iki dizinin pariteleri farklı ise, bu iki dizi kesinlikle birbirinden farklı demektir ve aralarında tek sayıda, kesinlikle en az 1 tane, farklı bitler vardır. Böylece, CASCADE protokollerinde olduğu gibi, sadece paritelerin kontrolü iki bit dizisi arasındaki hataları sezmekte ve düzeltmekte de kullanılabilir.

Şekil 4.4'te hatalı bir bit dizinde hata bulma işlemi gösterilmektedir. Bu şekilde, kırmızı bloklar paritesi uyuşmayan (tek sayıda hatalı bit içeren) blokları, yeşil bloklar ise paritesi uyuşan (0 ya da çift sayıda hatalı bit içeren) blokları göstermektedir. Bu şekilde hatalı bir blok sezildiğinde, hatalı biti bulmak için gönderici ve alıcı bit dizisini bloklara böler ve her bir bloğun paritesini değiş tokuş ederler. Bit dizisinde hata sezilmiş olduğuna göre bu bloklardan biri için pariteler uyuşmayacaktır. Gönderici ve alıcı paritesi uyuşmayan blokları hatalı bit bulunana kadar ikiye böler ve her bölme işlemi için parite değiş tokuşu yapar. Bu şekilde hatalı bit bulunduktan sonra bitin değeri değiştirilerek hata düzeltilmiş olur. k uzunluklu bir bloktaki hatalı bitin bulunması için yinelemeli BINARY işleminden kaynaklanan parite değiş tokuş sayısı $\log_2 k$ olmaktadır. Diğer bir deyişle, k uzunluklu bir bloktaki hatanın düzeltilmesi için ilave $\lceil \log_2 k \rceil$ kadar bilgi değiş tokuşu yapılmalıdır.

Protokol birkaç raunddan oluşur. Her bir raundda tüm hata sezmeler ve düzeltmeler sadece basit parite kontrolleri ile gerçekleştirilir. Raund 1'de, gönderici ve alıcı, raund 1 blok uzunluğu k_1 parametresini seçer ve dizilerini k_1 bit uzunluklu bloklara böler. Bu şekilde elde edilen toplam $\left\lceil N/k_1 \right\rceil$ adet blok gelecekte de kullanım için kaydedilir. Bir hata bulmak için, gönderici ve alıcı her bir bloğun paritelerini değiş-tokuş eder. Eğer bir blok için pariteler uyuşmazsa, BINARY süreci kullanılarak o blok içindeki sadece bir bit hata alıcı tarafında düzeltilir. BINARY içinde üretilen daha küçük bloklar gelecekte kullanım için saklanmazlar. Bulunan hata sadece alıcı tarafında sadece bitin değeri değiştirilerek düzeltilir. Raund 1'in sonunda, pariteleri uyuşmayan

tüm bloklar sezildiği ve düzeltildiği için, alıcının bloklarının tümü çift sayıda (belki de 0) hataya sahip olurlar.

Her bir raund i > 1'de, gönderici ve alıcı önce bit dizilerini karıştırır ve raund i blok uzunluğu k_i parametresini seçer. Daha sonra, raund 1 için bahsedilen süreci bu raundların her birinin toplam $\left\lceil N/k_i \right\rceil$ bloğu için tekrarlarlar. Bununla birlikte, bu raundlarda sürece bir geriye-iz-sürme kısmı da eklenir: Bu raundların birinde bir blokta yeni bir hata düzeltildiğinde, bu düzeltme o biti içeren tüm önceki hesaplanan bloklarda ve paritelerinde bir değişikliğe neden olur. Bu şekilde etkilenen bir blok, o zaman çift sayıda, en az iki, hataya sahip olduğundan, önceden farkedilmemiş bir hata içerir. Gönderici ve alıcı, en küçük etkilenen blok üzerinde BINARY'yi çalıştırır ve hatayı düzeltir. Bu düzeltme de o biti içeren tüm önceki bloklarda bir değişikliğe yol açar. Gönderici ve alıcı etkilenen en küçük blokları bulmaya, üzerinde BINARY'yi çalıştırmaya ve farkedilmemiş hataları düzeltmeye devam ederler. Bu geriye-iz-sürme yaklaşımı kullanılarak, daha fazla hata daha az sayıda raundda düzeltilir. Süreç her bir raund için geriye hiç parite uyuşmayan blok kalmayana kadar sürer.



Şekil 4.4. Tek sayıda hatalı bit içeren k uzunluklu bir blokta hata sezme ve düzeltme

Orijinal CASCADE protokolü için sözde kod Tablo 4.1'de verilmiştir.

Tablo 4.1. Orijinal CASCADE protokolünün algoritması

Orjinal Cascade protokolünün algoritması

```
1: CASCADE(BİTDİZİSİ BITDIZISIOKUYUCU)
2: i. İlklendirme:
3:     k_1 ← 0.73/ε
4:     k_i ← 2k_{i−1}   2 ≤ i ≤ Raund
5:     Raund ← 4
6:
7: ii. Raund i = 1:
8:     for l = 0;   l < ⌈N/k_1⌉;   l++   do
9:         gondericininParitesi ← pariteGetir(gondericiBlogu[l]);
10:        if pariteKontrol(aliciBlogu[l]) != gondericininParitesi then
11:            Binary(aliciBlogu[l]);
12:
13: iii. Raund i > 1:
14:    karıştır(bitDizisiOkuyucu)
15:    for l = 0;   l < ⌈N/k_i⌉;   l++   do
16:        gondericininParitesi ← pariteGetir(gondericiBlogu[l]);
17:        if pariteKontrol(aliciBlogu[l]) != gondericininParitesi then
18:            duzeltilenBit ← Binary(aliciBlogu[[l]);
19:            geriyeIzSur(duzeltilenBit);
```

Orijinal CASCADE protokolünün çalışma parametreleri aşağıdaki gibidir [16],

$$k_1 \cong \frac{0{,}73}{\varepsilon} \tag{4.1}$$

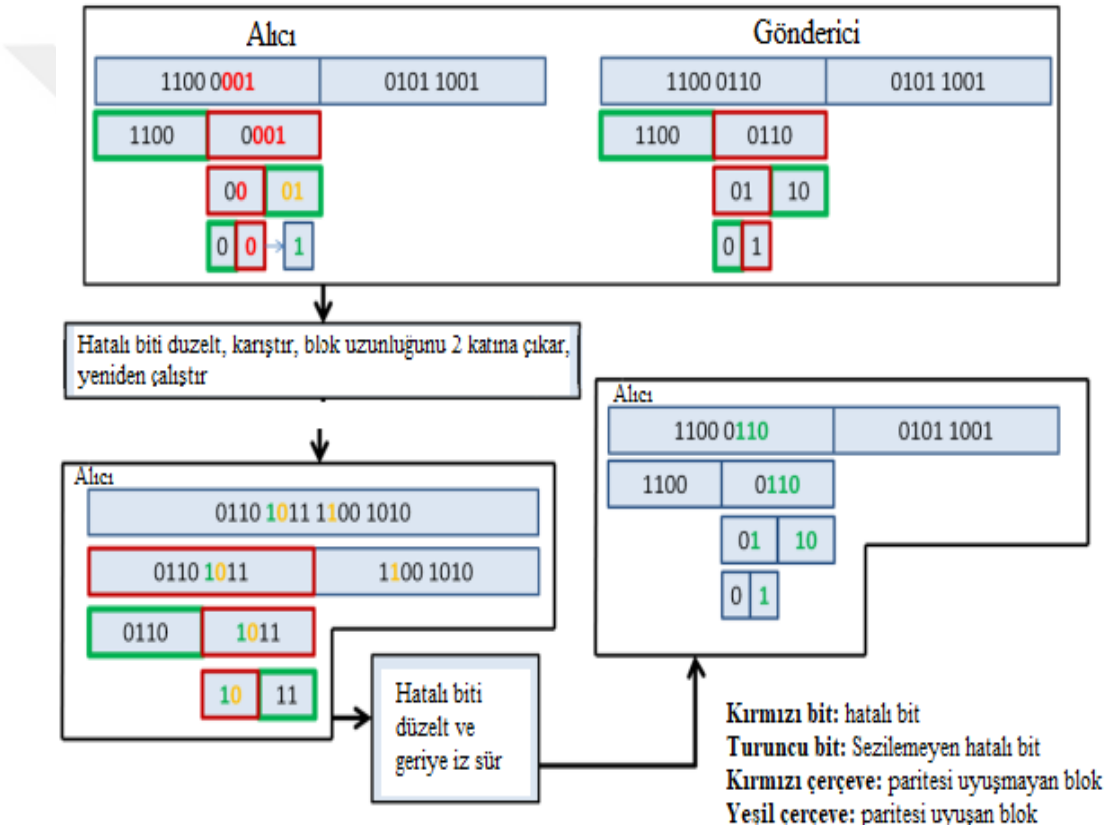
$$k_i = 2k_{i-1}\ , \quad 2 \le i \le \text{Raund} \tag{4.2}$$

$$\text{Raund} = 4 \tag{4.3}$$

[16]'daki orijinal CASCADE protokolünün yazarları blok uzunluklarının nasıl seçildiğine dair teorik bir bilgi vermemişlerdir (deneysel olarak seçmişlerdir). [16]'da sundukları çalışmada herhangi bir açıklama yapmaksızın round sayısını da 4 olarak seçmişlerdir.

Gönderici ile alıcı arasında CASCADE protokolü ile hata sezme ve düzeltme işleminin basit anlamda nasıl gerçekleştirildiğini gösteren bir örnek Şekil 4.5'te sunulmuştur. Şekil 4.5'ten de görülebileceği gibi gönderici alıcıya 1100 0110 0101 1001 bit dizisini gönderiyor. Ancak, bu bit dizisi kuantum kanaldan ve muhtemel hattı dinleyen saldırganlardan kaynaklı olarak alıcıya 1100 0001 0101 1001 olarak ulaşıyor. Şekilde

hatalı bitler kırmızı ile gösterilmiştir. Hatalı bitleri içeren bloklar da kırmızı çerçeve içerisine alınmıştır. Alıcı bit dizisini ikiye bölüyor ve bu blokları hafızaya kaydediyor. Alıcı, ilk parçada hatayı tespit ediyor ve BINARY hata düzeltme işlemine başlıyor. Alıcı, ilk raundda hatalı bir biti düzeltiyor. Ardından, bit dizisini karıştırıyor ve tekrar ikiye bölüyor. Yine ilk blokta hatayı seziyor ve hatalı biti düzeltiyor. Ancak, ilk raunddan farklı olarak bu raundda geriye iz sürerek hafızadaki bloklarda bu hatalı bitin geçtiği en küçük blok bulunur ve BINARY o blok üzerinde de çalıştırılır. Böylece, tüm hatalı bitler düzeltilmiş olur. Protokol yine de sonlandırılmaz ve yukarıda da tanımlandığı gibi 4 raund devam edilir.



Şekil 4.5. CASCADE protokolünün çalışma prensibi

4.1.4. CASCADE protokolü üzerinde yapılan değişiklikler

[16]'daki orijinal CASCADE protokolünü iyileştirmek için yapılan çalışmaların bir çoğu protokoldeki ilk raundun blok uzunluğu olan k_1'in seçilmesine odaklanmıştır [22-24]. Bu çalışmalardan bazıları da parametre iyileştirmesinin yanında BINARY yöntemi yerine daha verimli olduğunu iddia ettikleri yeni bir yöntem önermişlerdir [22, 23]. Bazıları ise protokolde mevcut olan ancak kapsamlı bir analiz sonucu tespit

edilebilen bazı içsel bilgileri kullanarak verimlilik performansını arttırmaya çalışmıştır [9, 24].

Sugimoto ve arkadaşlarının yapmış olduğu çalışmada [22], yazarlar öncelikle CASCADE protokolünün bilgisayar benzetimleri ile elde ettikleri bazı sayısal sonuçlarını vermişlerdir. Sonuçların analizinden sonra, herkese açık klasik kanal üzerinden değiş-tokuş edilen parite bitlerinin sayısını azaltmak için, buldukları yöntemleri kullanarak CASCADE protokolünü değiştirmeyi önermişlerdir. CASCADE protokolünü iki raunddan fazla tekrarlamaya gerek olmadığını görmüşlerdir. Bununla birlikte, bit dizilerinde birkaç hata kalmış olabileceğinden protokolü hemen ikinci raundun bitmesiyle sonlandırmamışlardır. Protokolün başarısız olmasını önlemek için (yani, protokol sonlandığında bit dizilerinde bir ya da birkaç hata kalması), ikinci raunddan sonra $BICONF^s$ (s defa tekrar edilen Binary Confirmation) sürecinin çalıştırılmasını önerdiler.

İki bit dizisinin eşit olduğunu onaylamak için parite-kontrolü tabanlı interaktif bir yöntem olan $BICONF^s$'i önermişlerdir. Bu yöntem aşağıdaki gibi çalışmaktadır:

1. Gönderici ve alıcı, bit dizilerinden rasgele bir alt küme, örneğin, bit dizilerinin yarısını, seçerler.
2. Gönderici, alt kümesinin paritesini alıcıya söyler.
3. Alıcı, kendi alt kümesinin aynı pariteye sahip olup olmadığını kontrol eder.
4. Sadece bu pariteler farklı olduğunda, gönderici ve alıcı, BINARY'i iki defa çalıştırır, biri alt küme için ve diğeri alt kümenin tümleyeni için.
5. Peş peşe s tekrar boyunca alıcı hata bulamayana kadar, adım 1 ile adım 4 arası tekrarlanır.

$BICONF^s$ yaklaşımının uygulanmasıyla, protokolün başarısız olma olasılığı kabaca 2^{-s}'den daha azdır.

Yazarlar çalışmalarında, orijinal CASCADE protokolünün ilk iki rounddaki blok uzunlukları için de yeni bir parametre kümesi önermişlerdir [22]. Değiş tokuş edilen bitlerin sayısını minimize etmek için raund 1 ve 2'nin blok uzunlukları aşağıdaki gibi belirlenmiştir,

$$k_1 = \left\lfloor \frac{4\ln 2}{3\varepsilon} \right\rfloor \tag{4.4}$$

$$k_2 = \left\lfloor \frac{4\ln 2}{\varepsilon} \right\rfloor \tag{4.5}$$

$$s = 10 \tag{4.6}$$

Burada, orijinal yöntem için $k_2 = 2k_1$ iken, önerilen yöntem için $k_2 \cong 3k_1$ olmuştur ve orijinal CASCADE protokolünün diğer iki raundunu yapmak yerine BICONF[10] çalıştırılmıştır.

Bu tez çalışmasında, [22]'de anlatılan yöntem gerçeklenmiş ve kapsamlı deney ve simülasyonlar yapılmıştır. Elde edilen sonuçlar hem orijinal [22] ile hem de bu çalışmada önerilen CASCADE tekniği ile karşılaştırılmış ve sonuçlar tablo ve şekillerle sunulmuştur. [22]'de önerilen yöntemin bu çalışmada geliştirilen versiyonunda ise, her iterasyonun başında bit dizileri karıştırılarak rasgelelik sağlanmıştır ve bit dizisinin alt kümesi olarak ilk yarısı seçilmiştir.

Yan ve arkadaşlarının yapmış olduğu çalışmada [23], yazarlar CASCADE'de sadece her bir raundun başında bölünen blokların kaydedildiğini farketmişlerdir. Ancak, BINARY boyunca birçok daha küçük blok üretilir. Eğer tüm bu daha küçük bloklar da kaydedilirse, o zaman geriye-iz-sürme kısmı daha küçük bloklarda arama yapabilir, değiş-tokuş edilen bilgi daha az olacaktır ve böylece protokol daha verimli olacaktır. Bu nedenle, [23]'te yazarlar stratejiyi değiştirmişlerdir ve protokolün parametrelerini de optimize etmişlerdir.

[23]'teki protokol yine birkaç raunddan oluşmaktadır. Ancak, CASCADE protokolü bu defa değiştirilmiş BINARY ile çalıştırılmıştır. i. raundda paritelerin uyuşmadığı bir j bloğu olduğunu varsayalım. Bu blok $K^i_{(j-1)k_i+1,\, jk_i}$ ile temsil edilebilir. Bu ifadede, alt simgeler bir bloğun başlangıç ve bitiş konumlarını temsil etmek için kullanılmaktadır. BINARY'nin ilk adımı şu şekilde değiştirilmiştir: $K^i_{(j-1)k_i+1,\, jk_i}$ bloğu iki daha küçük bloğa bölünmüştür: $K^i_{(j-1)k_i+1,\, (j-1)k_i+\left\lceil \frac{k_i}{2} \right\rceil}$ ve $K^i_{(j-1)k_i+\left\lceil \frac{k_i}{2} \right\rceil+1,\, jk_i}$. İlk olarak daha küçük olan bloğun paritesi değiş-tokuş edilmiştir. Daha sonra, büyük olan $K^i_{(j-1)k_i+1,\, jk_i}$ bloğu

hafızadan silinir ve yerine daha küçük olan ve bu bloğun parçaları olan bu iki blok eklenir: $K^{i}_{(j-1)k_i+1,\,(j-1)k_i+\left\lceil \frac{k_i}{2} \right\rceil}$ ve $K^{i}_{(j-1)k_i+\left\lceil \frac{k_i}{2} \right\rceil+1,\,jk_i}$. Bu küçük değişiklikle, CASCADE protokolü çalıştıkça blok kayıtları güncellenmeye devam edecektir ve değiştirilmiş BINARY sürecinden dolayı kaydedilen blokların uzunlukları gittikçe daha küçük olacaktır.

Yazarlar bu iyileştirmeleri gerçekledikten sonra, herkese açık olan klasik kanaldan değiş-tokuş edilen fazlalık bitlerin sayısını minimize etmek için protokol parametrelerini aşağıdaki gibi optimize etmişlerdir,

$$k_1 = \frac{0{,}80}{\varepsilon} \tag{4.7}$$

$$k_2 = 5k_1 \tag{4.8}$$

$$k_i = \frac{N}{2}, \quad 3 \le i \le \text{Raund} \tag{4.9}$$

$$\text{Raund} = 10 \tag{4.10}$$

Raund 3'ten itibaren, blok uzunluğunu $N/2$ olarak almışlardır ve kalan hatalardan dolayı protokolün başarısız olma olasılığını düşürmek için, raundların sayısını orijinal protokolün önerdiği değer olan 4'ten 10'a yükseltmişlerdir.

[24]'teki çalışmada Jesus ve arkadaşları CASCADE protokolü için optimum blok uzunluğu parametrelerini bulmaya çalışmışlardır. Yazarlar çalışmalarında sadece ilk raund blok uzunluğu olan k_1'in iyi seçilmesinin bile protokolün verimliliğini arttırmak için yeterli olduğunu ifade etmişlerdir. Yazarlar çalışmalarında, orijinal CASCADE protokolü ve onun en önemli türevleriyle ilgili kapsamlı bir analiz çalışması da sunmuşlardır. Bu yöntemlerde CASCADE protokolü çeşitli yönlerden iyileştirilmiştir: optimum blok uzunlukları, BINARY'de oluşturulan küçük blokların hafızaya alınması, daha iyi karıştırıcıların kullanımı, tekil (bir bitlik) blokların çıkarılması. Yazarlar her iyileştirmeyi orijinal CASCADE protokolüne uygulamışlardır ve dört farklı CASCADE protokolü elde etmişlerdir. Bu protokollerle kapsamlı simülasyonlar çalıştırılmış ve her biri için sonuçlar incelenmiştir. Yazarlar elde edilen sonuçların

tutarlılığı ile ilgili olarak da bir çalışma yapmıştır. Yazarlar [24]'teki çalışmalarında FER ve BER gibi kavramları da hesaba katmışlardır. Yazarlar, yüksek FER oranlarında çok fazla sayıda bit dizisi dışlandığı için elde edilen yüksek verimlilik değerlerinin gerçekçi olmadığını ifade etmişlerdir. Bu nedenle, verimlilik hesaplarında FER ve BER değerlerinin de hesaba katılmasıyla daha gerçekçi verimlilik değerlerinin hesaplanabileceğini ifade etmişlerdir.

[24]'teki çalışmada orijinal CASCADE protokolü üzerinde yapılan en temel değişiklik protokol parametrelerinin seçimi olmuştur. Yazarlar, CASCADE protokolü için en iyi verimlilik değerlerini verecek optimum parametre kümesini bulmak için iki boyutlu arama tekniği olan compass (pusula) arama algoritmasını kullanmışlardır. Bu teknikle elde ettikleri parametre kümesi şöyle verilmiştir,

$$\alpha=\log_2 \frac{1}{\varepsilon} - \frac{1}{2} \tag{4.11}$$

$$k_1 = 2^{\lceil \alpha \rceil} \tag{4.12}$$

$$k_2 = 2^{\left\lceil \frac{\alpha+12}{2} \right\rceil} \tag{4.13}$$

$$k_3 = 4096 \tag{4.14}$$

$$k_i = \left\lceil \frac{N}{2} \right\rceil, \quad 3 < i \leq \text{Raund} \tag{4.15}$$

$$\text{Raund} = 14 \tag{4.16}$$

$$N = 2^{14} \tag{4.17}$$

[75]'teki çalışmada da belirtildiği gibi, bit ve blok uzunlukları ikinin üstel katı olarak seçildiğinde CASCADE protokolünün verimlilik değerleri optimuma yaklaşmaktadır. Bunun sebebi ise, BINARY tekniğindeki rekürsif olarak ikiye bölme işleminden kaynaklanmaktadır. Bit dizisinin ve blokların uzunlukları ikinin üstel katı seçildiğinde BINARY tekniğinde oluşan bloklar da iki ve ikinin katlarında olacaktır. Jesus ve arkadaşları kendi çalışmalarında bu bilgiyi de kullanarak verimlilik değerlerini arttırmaya çalışmışlardır. Yazarlar nihai verimlilik değerlerini [23]'te belirtilen CASCADE protokolü ile önerdikleri optimum parametre kümesinin birlikte

kullanımıyla elde etmişlerdir ve bu sonuçların literatürdeki en başarılı CASCADE verimlilik değerleri olduğunu iddia etmişlerdir.

Bu tez çalışmasında, [16, 22-24]'te verilen CASCADE protokolü versiyonları ayrı ayrı gerçeklenmiştir. Bu bölümde bu yöntemler kapsamlı deneyler ve simülasyonlarla desteklenerek birbirleriyle karşılaştırılmıştır. Elde edilen sonuçlar Tablo 4.2'de ve Şekil 4.6'da verilmiştir. Bu simülasyonlarda, gizli anahtarın uzunluğu N = 10000 bit olarak seçilmiştir ve verimlilik değerleri μ de 100 başarılı simülasyonun (FER = BER = 0) ortalama değeri olarak hesaplanmıştır. Literatürdeki diğer çalışmalardan farklı olarak elde edilen sonuçlar daha geniş bir kanal hata olasılık değerleri için verilmiştir, $\%0{,}1 \le \varepsilon \le \%50$.

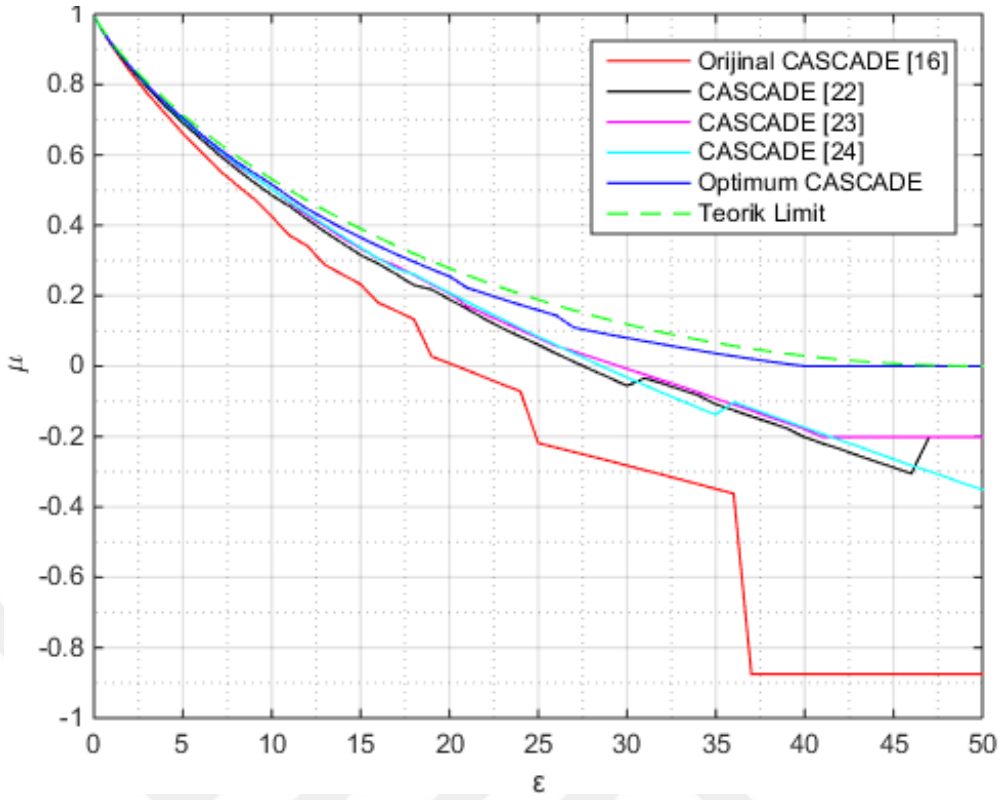
Tablo 4.2'de ve Şekil 4.6'da her yöntem kendi parametre kümesi ile çalıştırılmıştır. Bu tez çalışmasında önerilen optimum CASCADE protokolü Bölüm 4.2 ve 4.3'te incelenmiştir, ancak, bu bölümde bahsedilen yöntemlerle karşılaştırmak maksadıyla tablo ve şekillerde yer verilmiştir. Optimum CASCADE için elde edilen verimlilik değerleri [23]'te önerilen parametre kümesi ile çalıştırılmıştır. Tablo ve şekilden de görüldüğü gibi bu çalışmada önerilen optimum CASCADE protokolü bahsi geçen tüm yöntemlerden daha verimlidir, teorik limite daha yakındır.

Tablo 4.2. Optimum CASCADE yöntemi ile diğer yöntemlerin karşılaştırılması

ε (%)	Orijinal CASCADE [16]	CASCADE [22]	CASCADE [23]	CASCADE [24]	Optimum CASCADE	Teorik Limit
0.1	0.987502	0.985962	0.986680	0.985797	0.987584	0.988592
0.2	0.977050	0.975992	0.976969	0.976207	0.977882	0.979186
0.3	0.967459	0.966923	0.968148	0.967266	0.969065	0.970536
0.4	0.958288	0.958218	0.959600	0.959034	0.960512	0.962378
0.5	0.949492	0.949977	0.951442	0.950871	0.952376	0.954585
0.6	0.941242	0.942192	0.943895	0.943243	0.944874	0.947085
0.7	0.932543	0.934708	0.936403	0.935784	0.937386	0.939828
0.8	0.924588	0.927290	0.929007	0.928435	0.929974	0.932778
0.9	0.916774	0.919652	0.921653	0.921256	0.922668	0.925912
1.0	0.909178	0.912491	0.914741	0.914422	0.915781	0.919207
2.0	0.838756	0.847671	0.851357	0.851453	0.852787	0.858559
3.0	0.774571	0.791532	0.796456	0.796740	0.798424	0.805608
4.0	0.717668	0.738232	0.744691	0.745986	0.747887	0.757708
5.0	0.661736	0.691012	0.700615	0.700111	0.704362	0.713603
6.0	0.609952	0.646097	0.654559	0.656322	0.659508	0.672555
7.0	0.558847	0.601144	0.611517	0.614234	0.618784	0.634076
8.0	0.515883	0.560464	0.571386	0.575012	0.581320	0.597821
9.0	0.475340	0.522212	0.538127	0.538766	0.548644	0.563530
10.0	0.424563	0.485452	0.502952	0.502850	0.516040	0.531004

Tablo 4.2. (Devam) Optimum CASCADE yöntemi ile diğer yöntemlerin karşılaştırılması

ε (%)	Orijinal CASCADE [16]	CASCADE [22]	CASCADE [23]	CASCADE [24]	Optimum CASCADE	Teorik Limit
11.0	0.371622	0.454197	0.464864	0.467587	0.480516	0.500084
12.0	0.341292	0.416961	0.428023	0.433862	0.446700	0.470639
13.0	0.287354	0.381306	0.396600	0.400530	0.418489	0.442562
14.0	0.260032	0.348480	0.363401	0.368626	0.390483	0.415761
15.0	0.232394	0.315701	0.334283	0.337038	0.365737	0.390160
16.0	0.180380	0.290978	0.305174	0.306313	0.341198	0.365690
17.0	0.156786	0.261262	0.284665	0.276797	0.317859	0.342295
18.0	0.133234	0.230045	0.258984	0.261269	0.296671	0.319923
19.0	0.027366	0.218027	0.232501	0.234662	0.275089	0.298529
20.0	0.007989	0.189652	0.207149	0.208511	0.255107	0.278072
21.0	-0.011727	0.162377	0.170361	0.182701	0.222495	0.258517
22.0	-0.032346	0.133590	0.147775	0.157289	0.206229	0.239832
23.0	-0.052460	0.107388	0.124677	0.132734	0.189780	0.221989
24.0	-0.071619	0.083487	0.102169	0.107825	0.173944	0.204960
25.0	-0.218871	0.060869	0.079999	0.083654	0.159041	0.188722
26.0	-0.231334	0.035816	0.057169	0.059831	0.144071	0.173254
27.0	-0.243931	0.012663	0.042302	0.036622	0.109806	0.158535
28.0	-0.256786	-0.010864	0.025700	0.012804	0.099532	0.144549
29.0	-0.269431	-0.032658	0.009474	-0.009704	0.089699	0.131279
30.0	-0.282474	-0.056078	-0.007620	-0.032295	0.080059	0.118709
31.0	-0.295678	-0.034152	-0.023989	-0.054134	0.070824	0.106827
32.0	-0.308735	-0.049775	-0.041069	-0.076083	0.061889	0.095619
33.0	-0.322063	-0.066311	-0.056930	-0.097019	0.052956	0.085074
34.0	-0.335422	-0.082715	-0.074462	-0.118348	0.044571	0.075181
35.0	-0.348755	-0.108559	-0.092470	-0.138092	0.035780	0.065932
36.0	-0.362221	-0.125509	-0.109719	-0.102226	0.028225	0.057317
37.0	-0.875000	-0.142981	-0.126533	-0.120010	0.020547	0.049328
38.0	-0.875000	-0.159736	-0.144898	-0.138543	0.013284	0.041958
39.0	-0.875000	-0.176846	-0.161820	-0.156648	0.006448	0.035200
40.0	-0.875000	-0.202259	-0.179584	-0.174756	-0.00010	0.029049
41.0	-0.875000	-0.219748	-0.201600	-0.192774	0.000000	0.023500
42.0	-0.875000	-0.235962	-0.201600	-0.210599	0.000000	0.018546
43.0	-0.875000	-0.253903	-0.201600	-0.227653	0.000000	0.014185
44.0	-0.875000	-0.271153	-0.201600	-0.246453	0.000000	0.010412
45.0	-0.875000	-0.287700	-0.201600	-0.264181	0.000000	0.007226
46.0	-0.875000	-0.305524	-0.201600	-0.282157	0.000000	0.004622
47.0	-0.875000	-0.202000	-0.201600	-0.298709	0.000000	0.002598
48.0	-0.875000	-0.202000	-0.201600	-0.316138	0.000000	0.001154
49.0	-0.875000	-0.202000	-0.201600	-0.334723	0.000000	0.000289
50.0	-0.875000	-0.202000	-0.201600	-0.350668	0.000000	0.000000



Şekil 4.6. Optimum CASCADE yöntemi ile diğer yöntemlerin karşılaştırılması

4.2. CASCADE Protokolü Üzerine Yapılan İyileştirmeler

Gönderici ve alıcı tarafından değiş-tokuş edilen her bir parite kanalı dinleyen saldırgana bir bit bilgi açığa çıkarır, o paritenin önceki değiş-tokuş edilmiş paritelerden hesaplanabilmesi hariç, böylece BU'nun verimlilik performansını düşürür.

Sızma miktarının minimize edilmesi, ve verimliliğin arttırılması için, [23]'te yapıldığı gibi hataların mümkün olduğu kadar küçük bloklarda aranması gerekmektedir. Bununla birlikte, [23]'te verilene ek olarak bunu başarmak için halen başka yollar da vardır. Protokol içinde kullanılabilecek halihazırda bazı başka içsel bilgiler de mevcuttur. Böylece, hataları aramak için daha küçük bloklar kullanarak değiş-tokuş edilen parite bitlerinin sayısını azaltmak yoluyla CASCADE protokolünün verimliliği daha fazla iyileştirilebilmektedir. BINARY yöntemini daha büyükler yerine bu küçük bloklar üzerinde çalıştırmakla değiş-tokuş edilen fazlalık bilgi daha az ve böylece CASCADE protokolü daha verimli olacaktır. Üzerlerinde BINARY'yi çalıştırmadan önce blokları daha küçük yapmak üzere, bu tez çalışmasında bizim önerdiğimiz yöntemler bu bölümde anlatılmaktadır.

CASCADE protokolünde parite değiş tokuş işlemleri her raundun başında bit dizisinde hatalı bit olup olmadığını anlamak için ve hatalı bit içeren blok üzerinde BINARY yöntemi çalıştırılırken kullanılır. Bir bloktaki hatayı sezmek için blok uzunluğundan bağımsız olmak üzere bir parite biti değiş tokuş edilir. Eğer bu blokta hata sezildiyse, BINARY yönteminden kaynaklanan ekstra $\lceil \log_2 k \rceil$ bit kadar ilave bilgi değiş tokuşu yapılır. Bu işlem sonunda ise sadece bir bit düzeltilir.

i raund sayısını göstermek üzere ve CASCADE protokolü boyunca j adet hata sezilen (paritesi uyuşmayan) blok tespit edildiği durumda protokol boyunca açığa çıkan (sızan) toplam bilgi miktarı (parite değiş tokuşu) E_i, Denklem (4.18)'te verilmiştir:

$$E_i = j(1+\lceil \log_2 k_i \rceil) + (\left\lceil N/k_i \right\rceil - j) + T_i \qquad (4.18)$$

Bu ifadede paritesi uyuşmayan her blok için $1 + \lceil \log_2 k_i \rceil$ kadar parite değiş tokuşu yapılacaktır. Paritesi uyuşan her blok için bir bitlik parite değiş tokuşu yapılır, yani toplamda $\left\lceil N/k_i \right\rceil - j$ kadar parite değiş tokuşu olacaktır. T_i ise geriye iz sürme adımlarında çalıştırılan BINARY'lerden kaynaklanan açığa çıkmış parite bitlerinin toplam sayısını göstermektedir. İlk round için $T_1 = 0$ olacaktır. Buradan da görülebileceği gibi CASCADE protokolünde verimlilik kaybının en fazla olduğu yer BINARY tekniğidir. Bu nedenle ya çalıştırılan BINARY sayısı azaltılmalı ya da BINARY'nin çalıştırıldığı blok uzunluğu azaltılmalıdır. Hatalı bit varsa BINARY çalıştırmaktan kurtulamayacağımız için BINARY'nin üzerinde çalıştığı blok uzunluğunu azaltmanın yollarını aramak daha gerçekçidir.

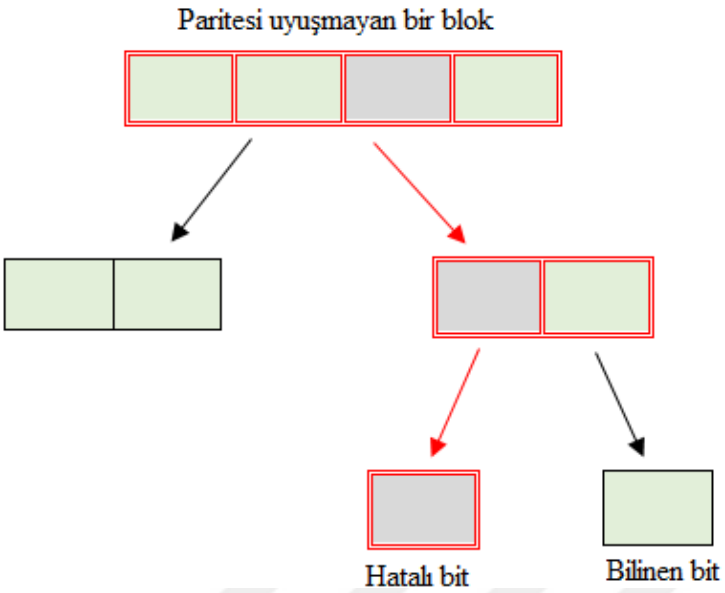
Değiş tokuş edilen parite bitlerinin sayısını azaltmak için BINARY işlemi olabildiğinde en küçük bloklarda çalıştırılmalıdır [23]. Böylece, CASCADE protokolünün verimliliği artacaktır. Bu çalışmadaki yaklaşım doğru olmakla beraber verimliliği arttırmanın başka yolları da vardır. Bu amaçla bu tez çalışmasında, protokol içinde verimliliği arttırmak amacıyla kullanılabilecek bilgilerin olup olmadığı incelenmiştir ve tamamen bilinen bitler ve paritesi bilinen bloklar fikirleri önerilmiştir.

4.2.1. Tamamen bilinen bitler

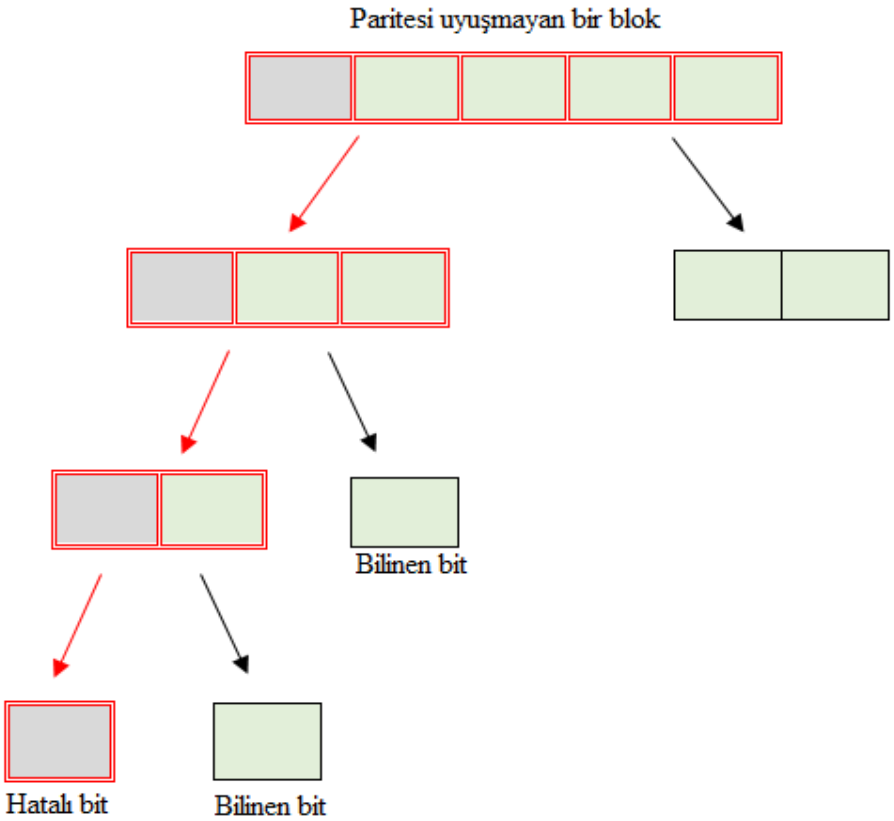
Protokol boyunca birçok hatalı bit düzeltilir. Hatası düzeltilen her bir bite Tamamen Bilinen Bit (TBB) adı verilir. Bununla birlikte, bir bit düzeltildiğinde, artık hatalı bit olamayacağı için, o biti gelecek BINARY uygulamalarında dahil etmeye gerek yoktur. Düzeltilen bitlerin çıkarılması hata aradığımız bloğun uzunluğunu düşürecektir ve böylece hatayı bulmak için değiş-tokuş ettiğimiz parite bitlerinin sayısı azalacaktır. Denklem (4.18)'te de gösterildiği gibi, CASCADE protokolünün verimlilik performansını etkileyen en önemli unsur BINARY tekniğidir. BINARY tekniğinin üzerinde çalıştığı blok uzunluğu ne kadar az olursa değiş tokuş edilen fazlalık parite bit miktarı da o kadar az olacaktır. k uzunluklu bir blok üzerinde BINARY tekniği çalıştırıldığında $\lceil \log_2 k \rceil$ kadar parite biti değiş tokuş edilecektir. Parite kontrolü durumunda ise, kontrolü yapılan blok için bir bitlik parite değiş tokuşu yapılmaktadır. Görüldüğü gibi, verimlilik performansını daha çok etkileyen BINARY işlemleridir. Bu nedenle blok uzunluğunu azaltmak için yapılan her iyileştirme protokolün verimliliğini göreceli olarak daha fazla arttıracaktır.

Şimdi, bir hatanın nasıl bulunduğunu düşünelim: gönderici ve alıcı iki bit uzunluklu bir bloğu bölüyor olsun. Şekil 4.7'de gösterildiği gibi, iki uzunluklu bloğun paritesinin ve düzeltilmiş bitin değerinin bilinmesi, bize diğer bitin değerini de söyler. Gelecekte hata ararken düzeltilen bu biti dahil etmeye de herhangi bir gerek yoktur. [19]'da da yapıldığı gibi, değerlerini tam olarak bildiğimiz hatalı olmadığı kesin olan bu iki biti TBB olarak adlandırmaktayız. Bu bitleri tespit edebilmek için gereken bilginin klasik kanal üzerinden değiş-tokuş edilmesi nedeniyle, değerlerini hattı dinleyen saldırgan da bilmektedir.

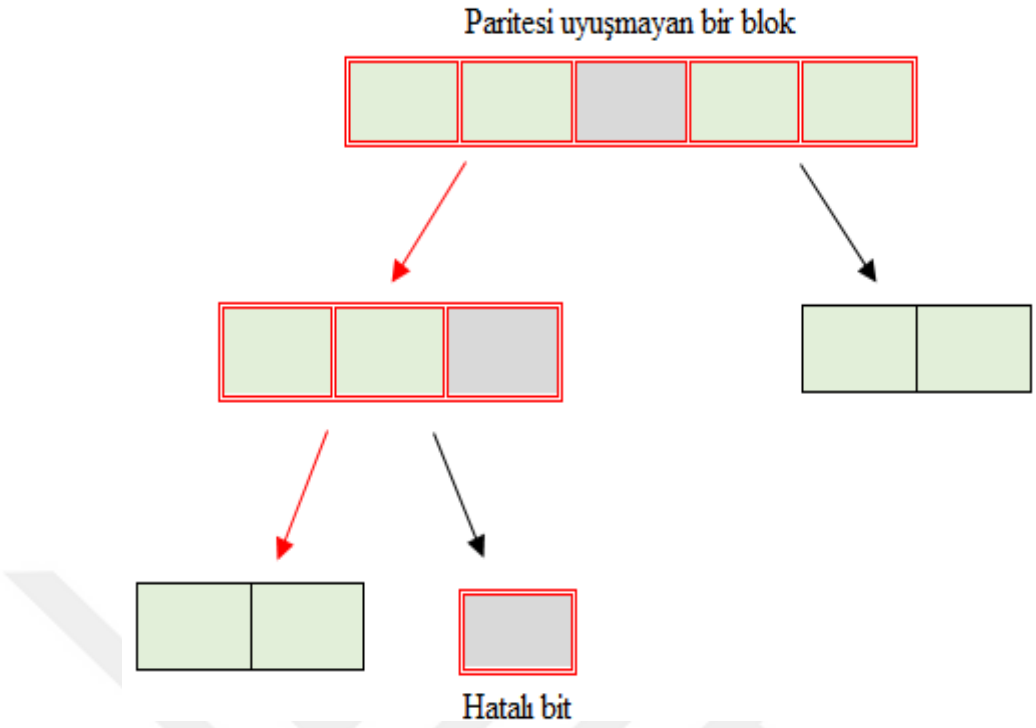
Fikri üç uzunluklu bloklarla devam ettirmek de mümkündür. Şekil 4.8'de görüldüğü gibi, üç uzunluklu bir bloktaki üç bitin tamamının değerini elde edebiliriz. Eğer hata üç uzunluklu bir bloğun sol dalında yer alıyorsa, üç bitin tamamı tam olarak bilinebilir, biz ve hattı dinleyen saldırgan tarafından. Bununla birlikte, eğer hata sağ dalda ise, bizim gerçeklemelerimizde sadece bir bit tam olarak bilinen olacaktır. Bu durum Şekil 4.9'da da gösterilmiştir. Bu çalışmada, eğer bloğun uzunluğu birden büyük tek sayı ise, bloğun küçük parçası sağ taraf olarak seçilmiştir.



Şekil 4.7. İki bit uzunluklu bloktaki durum



Şekil 4.8. Üç bit uzunluklu bloktaki hatalı bitin sol dalda olma durumu



Şekil 4.9. Üç bit uzunluklu bloktaki hatalı bitin sağ dalda olma durumu

Bu yöntemlerle elde edilen tamamen bilinen bitler, BINARY uygulamadan hemen önce bloklardan çıkarılarak bloklar daha küçük yapılabilir. Ayrıca, parite kontrolü yapmadan önce, bir blok tamamen bilinen bitlerden oluşturulabiliyorsa parite kontrolü yapmaya da gerek kalmayacaktır. Bu nedenle, protokolün her aşamasında hafızaya alınan bu tamamen bilinen bitler protokolün ileriki aşamalarında kullanılarak gereksiz parite kontrolü olabildiğince azaltılmaya çalışılacaktır. Bu da protokolün verimliliğini arttıracaktır.

Buradan da anlaşılabileceği gibi, orijinal protokolün iç adımlarında gizli ve iyileştirilmeye açık olan noktalar olduğu anlaşılmaktadır. Orijinal CASCADE protokolü aslında geriye iz sürme adımında BINARY işlemini en küçük blokta çalıştırmak isterken de aynı amacı hedeflemişti; BINARY işleminde değiş tokuş edilen fazlalık bilgi miktarını en az seviyeye indirmek. TBB fikri ilk defa 2000 yılında ortaya atılmıştır [93, 94]. Yazar bu çalışmasında, BINARY sırasında düzeltilen bitlerin protokolün ileriki aşamalarında kullanılarak, bloklardan çıkarılarak, BINARY'nin çalışacağı bloğun uzunluğunun azaltılabileceğini ifade etmiştir. Ancak, bu çalışmada sadece hatası düzeltilen bit ile birlikte tamamen bilinebilir olan tüm bitler de hafızaya alınmaktadır. Yani, [93, 94]'teki bir bitlik kazanç bu tez çalışmasında önerilen

yöntemlerle iki veya üç bite kadar çıkmaktadır. Bu değişiklik CASCADE protokolünün genel verimlilik performansını önemli oranda arttırmaktadır. Ayrıca, bu iyileştirme parite kontrolü adımlarında da kullanılmıştır. Parite kontrolü yapmadan hemen önce blok üzerinde arama işlemleri yapılarak bloğun tamamen bilinen bitlerden oluşturulabilmesi durumunda, parite kontrolüne ihtiyaç olmayacağı değerlendirilmiş ve parite kontrolü yapılmamıştır. Bu da protokole gerçekleşen her durum için bir bitlik kazanç sağlamıştır. Bu iyileştirmenin detayları Bölüm 4.3'te açıklanmıştır. Bu tez çalışmasında bu durum kapsamlı deneylerle ispatlanmış ve deney sonuçları Bölüm 4.3'ün sonunda verilmiştir.

4.2.2. Paritesi bilinen bloklar

Orijinal CASCADE protokolünde, her raundun başında bit dizisi bloklara bölünür ve bu bloklar geriye iz sürme aşamasında kullanılmak üzere hafızaya alınır. Paritesi bilinen ve hafızaya alınan bu bloklara Paritesi Bilinen Bloklar (PBB) adı verilir. BINARY işlemi dikkatle incelenirse, bu işlem sırasında da birçok ve daha küçük blokların oluşturulduğu görülebilir. Ancak, bu küçük bloklar orijinal CASCADE protokolünde hafızaya alınmamaktadır. [23]'teki çalışmada yazarlar CASCADE protokolünün geriye iz sürme sırasında hafızadaki en küçük blokta arama yapma fikrinden esinlenmiş olacaklar ki, hafızaya raundun başındaki büyük blokları almak yerine BINARY sırasında oluşturulan küçük blokları almayı tercih etmişlerdir. Böylece, geriye iz sürme işlemi daha küçük bloklar üzerinde çalışmaktadır. Bu açıdan, orijinal CASCADE protokolündeki fikri biraz daha geliştirmişlerdir ve protokolün verimliliğini arttırmayı başarmışlardır. Ayrıca, bir bakıma raundun başındaki blok da hafızaya alınmış olmaktadır. Çünkü, BINARY sırasında oluşan küçük bloklar büyük bloğun parçaları olduğu için, hafızaya büyük blok yerine küçük parçalarının alınması büyük bloğun da alındığı anlamına gelmektedir. Bunun yanında, BINARY'nin üzerinde çalışacağı blok uzunluğunu da düşüreceği için verimliliğe de olumlu katkı sağlamaktadır. Ancak, hafızada tutulan blok sayısı arttığı için bu yöntemde ilave bellek ihtiyacı olmaktadır. Yine, blok sayısı arttığı için geriye iz sürme adımında ekstra arama işlemleri de gerekeceğinden ötürü işlemci kaynağında da artışa ihtiyaç olmaktadır. Bu ihtiyaçlar marjinal seviyede kalmakta ve mevcut bilgisayar teknolojileri ile fazlasıyla, kolayca ve çok düşük maliyetlerle karşılanabilmektedir.

TBB'deki duruma benzer olarak, buradaki temel prensip de BINARY'nin üzerinde çalıştığı blok uzunluğunu olabildiğince azaltmaktır. Blok uzunluğu ne kadar az olursa değiş tokuş edilmesi gereken parite biti sayısı da o kadar az olacaktır, böylece protokolün verimliliği de o kadar fazla olacaktır. Bu durum Denklem (4.18) ile açıklanmıştır.

Bu tez çalışmasında ise, hafızaya alınan bu bloklar sadece geriye iz sürme aşamasında değil, aynı zamanda parite kontrolü ve BINARY işlemlerinde de kullanılmıştır. Böylece, hem BINARY işlemine sokulan bloğun uzunluğu hem de değiş tokuş edilen parite biti miktarı çok daha fazla azaltılmış ve protokolün verimliliği de önemli oranda arttırılmıştır. Ayrıca, bu iyileştirme parite kontrolü adımlarında da kullanılmıştır. Parite kontrolü yapmadan hemen önce blok üzerinde arama işlemleri yapılarak bloğun hafızadaki paritesi bilinen küçük bloklardan oluşturulabilmesi durumunda, parite kontrolüne ihtiyaç olmayacağı görülmüştür ve parite kontrolü yapılmamıştır. Bu da protokole gerçekleşen her durum için bir bitlik kazanç sağlamıştır. Bu iyileştirmenin detayları Bölüm 4.3'te açıklanmıştır.

Protokol boyunca birçok küçük blok oluşturulur, bu bloklar hafızaya alınır. PBB iyileştirmesinin temel amacı parite değiş tokuşu gerektiren her işlem için değiş tokuş yapmadan önce hafızadaki blokların kullanılıp kullanılmayacağına karar vermektir. Bu maksatla, bir blok üzerinde BINARY'e başlamadan önce, gönderici ve alıcı, Şekil 4.10'da görüldüğü gibi blok içinde yer alan ve paritelerinde uzlaştıkları önceden hesaplanmış daha küçük blokları çıkarabilir. Böylece, bu da bloğu daha küçük yapacaktır.

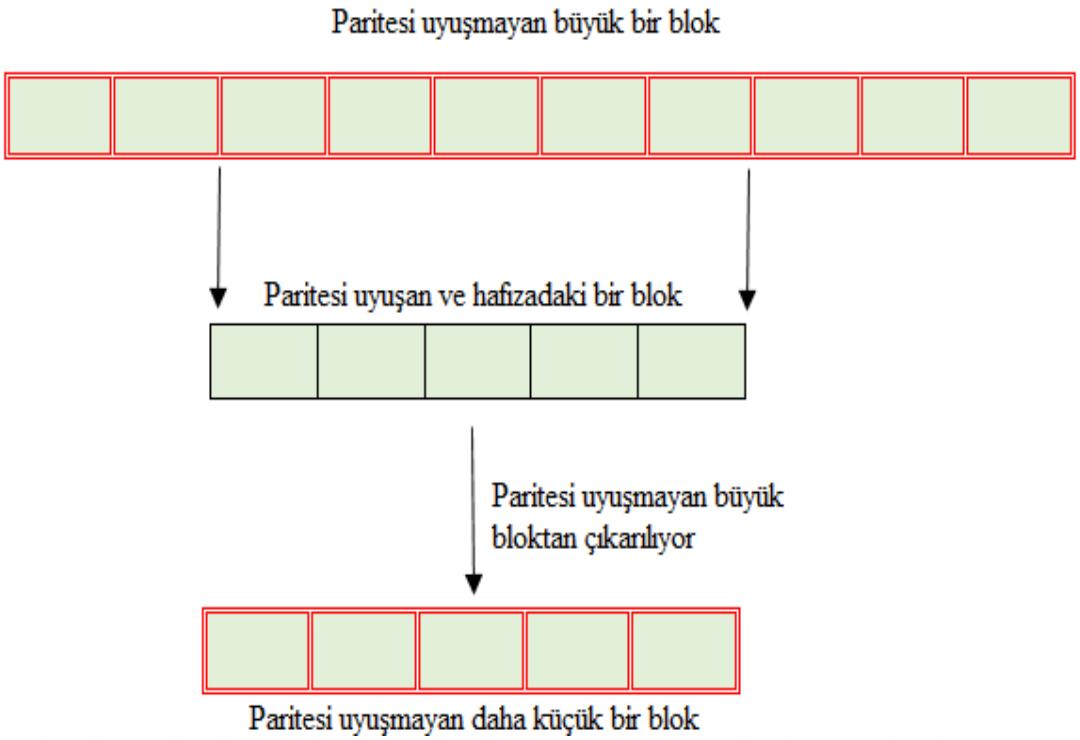
Özetle, BINARY uygulamadan hemen önce, bir blok içindeki

- tam olarak bilinen bitler ve
- önceki pariteleri uyuşan daha küçük bloklar

çıkarılarak, hata aramak için daha küçük bir blok elde edilebilir.

Daha küçük bloklar açığa çıkan bilgi miktarının daha az olmasını ve daha yüksek bir verimliliği sağlayacaktır. Bununla birlikte, bahsedilen fikirleri gerçeklemek için, BINARY'den hemen önce, paritesi uyuşmayan bloğun tam olarak bilinen bitlerin ve önceden hafızaya alınmış paritesi uyuşan daha küçük blokların herhangi birini içerip

içermediğine yönelik, ilave arama işlemlerine ihtiyaç vardır. Bu da Bölüm 4.2.1'de de açıklandığı gibi bellek ve işlemci kaynaklarının kullanımının az da olsa artmasına neden olmaktadır.



Şekil 4.10. Paritesi bilinen blokların büyük bloklardan çıkarılması

Şekil 4.10'dan da görülebileceği gibi ve TBB iyileştirmesinde olduğu gibi bu iyileştirme de bloğun uzunluğunu küçültmektedir ve BINARY tekniği olabildiğince en küçük blok üzerinde çalışmaktadır. Bu da CASCADE protokolünün verimliliğini daha fazla arttırmaktadır. Burada dikkat edilmesi gereken bir nokta da paritesi bilinen bloğun uzunluğunun bir olduğu durumda TBB iyileştirmesinin PBB iyileştirmesinin özel bir hali olmasıdır. Yapılan literatür araştırmalarının sonucuna göre, bu bölümde önerilen PBB iyileştirmesi ilk defa bu tez çalışmasında ortaya atılmıştır.

Bu iyileştirmenin detayları Bölüm 4.3'te açıklanmıştır. Bu tez çalışmasında, bu iyileştirme için kapsamlı deneyler yapılmış ve deney sonuçları Bölüm 4.3'ün sonunda verilmiştir.

4.3. Önerilen Optimum CASCADE Protokolü

Bu tez çalışmasında önerilen TBB ve PBB iyileştirmeleri kullanılarak CASCADE protokolünün parite kontrolü ve BINARY teknikleri iyileştirilmiştir ve böylece protokolün verimlilik performansı arttırılmıştır. Yapılan değişiklikler CASCADE protokolünün temel yapısını değiştirmemiş, sadece protokolün kullandığı yöntemleri tespit edilen iyileştirmeleri uygulayarak daha verimli hale getirmiştir.

Bu çalışmada önerilen bu yeni hata sezme ve düzeltme stratejisi Tablo 4.3 ve Tablo 4.4'te verilmiştir. Algoritmalarda parite kontrolü ve BINARY adımları sırasıyla Yeni Parite Kontrolü ve Yeni BINARY olarak adlandırılmıştır. Yeni ifadesi önerilen yöntemin eski yöntemin iyileştirilmiş bir versiyonu olduğunu göstermektedir.

Algoritmalarda geliştirilen uygulamadaki yazılımsal adımlar gösterilmiştir. Algoritmanın sol tarafında verilen rakamlar satır numaralarını göstermektedir. Algoritmanın ilk satırında geliştirilen metodun/fonksiyonun ismi bulunmaktadır. Fonksiyonla birlikte parantez içinde parametre olarak blok verilmiştir. Ayrıca, algoritmanın her alt akışında neler yapıldığı yorum cümleleriyle açıklanmaya çalışılmıştır.

Önerilen bu yeni yöntemde iyileştirmeler fonksiyona verilen bu blok parametresine uygulamaktadır. Eğer bu çalışmada önerilen TBB ve PBB iyileştirmeleri sonucu hafızaya alınan bilgiler blok içinde varsa, bu bilgiler bloktan çıkarılır. Böylece, gereksiz parite kontrolü yapılmaz ya da blok uzunluğu azaltılmış olur. Her iki iyileştirmede de değiş tokuş edilen parite miktarı azaltılmış olur. Diğer bir deyişle, yöntemin verimliliği arttırılmış olmaktadır.

Algoritmanın çalışmasıyla ilgili diğer detaylı bilgiler ise algoritmaların hemen altında açıklanmıştır.

Tablo 4.3. Yeni Parite Kontrolü algoritması

Yeni Parite Kontrolü

```
1: YeniPariteKontrolu(BLOK blok)
2: i. Parite Kontrolünden Once:
3:     ▷ Eğer blok tamamen bilinen bitlerden oluşturulabiliyorsa, parite kontrolü yapılmaz.
4:     if blok_tamamen_bilinen_bitlerden_olusturulabiliyormu(blok) then
5:         return true
6:
7:     ▷ Eğer blok hafızadaki paritesi bilinen bloklardan oluşturulabiliyorsa, parite kontrolü
   yapılmaz.
8:     if blok_paritesi_bilinen_bloklardan_olusturulabiliyormu(blok) then
9:         return true
10:
11:
12: ii. Parite Kontrolünü Yap:
13:     ▷ Eğer blok hafızadaki tamamen bilinen bitler ve paritesi bilinen bloklardan
   oluşturulamıyorsa, parite kontrolü yapılır.
14:     gondericininParitesi ← pariteGetir(gondericininBlogu);
15:
16:     ▷ Bu durumda, standart Parite Kontrol yöntemi çalıştırılır. Eğer pariteler
   uyuşmuyorsa, bu yöntem geriye olumsuz yanıt döner.
17:     if pariteKontrolu(blok) ! = gondericininParitesi then
18:         return false
19:
20:
21: iii. Parite Kontrolünden Sonra:
22:     ▷ Eğer blok uzunluğu 1 ise, bu durumda bu bit tamamen bilinen bitler listesine eklenir
23:     blok_uzunlugu ← blok_uzunlugu(blok)
24:     if blok_uzunlugu == 1 then
25:         tamamen_bilinen_bitlere_ekle(block[0]);
26:
27:     ▷ Eğer diğer tüm bitler tamamen biliniyorsa, kalan 1 biti tamamen bilinen bitler
   listesine ekle
28:     kucuk_blok ← tamamen_bilinen_bitleri_cikar(blok)
29:     kucuk_blok_uzunlugu ← blok_uzunlugu(kucuk_blok)
30:     if kucuk_blok_uzunlugu == 1 then
31:         tamamen_bilinen_bitlere_ekle(smaller_block[0]);
32:
33:     ▷ Eğer diğer tüm bitler paritesi bilinen bloklardan elde edilebiliyosa, kalan 1 biti tama-
   men bilinen bitler listesine ekle
34:     kucuk_blok ← paritesi_bilinen_bloklari_cikar(block)
35:     kucuk_blok_uzunlugu ← blok_uzunlugu(kucuk_blok)
36:     if kucuk_blok_uzunlugu == 1 then
37:         tamamen_bilinen_bitlere_ekle(smaller_block[0]);
38:
39:
40:     ▷ Parite Kontrolü başarılı olmuştur. Geriye olumlu yanıt dönülür.
41:     return true
```

Tablo 4.3'te Yeni Parite Kontrolü tekniğinin akışı anlatılmaktadır. Bu yöntemin temel amacı eski yöntemde yapılan fazlalık parite kontrollerinin yapılmaması ve böylece verimlilik performansının arttırılmasıdır. Bu maksatla TBB ve PBB iyileştirmeleri kullanılmaktadır. Algoritma şöyle çalışmaktadır:

- Parite kontrolü yapılacak olan blok, eğer tamamen bilinen bitlerden oluşturulabiliyorsa ya da parite bilgisi daha önceden hafızaya alınmış olan paritesi bilinen bloklardan elde edilebiliyorsa, parite kontrolüne gerek kalmaz ve yöntem paritesinin uyuştuğuna (hata sezilmediğine) dair geriye *true* bilgisi döner. Gönderici ve alıcı arasında parite değiş tokuşu yapılmaz.
- Yukarıdaki koşullar sağlanmıyorsa, blok üzerinde parite kontrolü (değiş tokuşu) yapılır.
- Parite kontrolü sonucunda paritelerin uyuşmadığı (blokta tek sayıda hatalı bit olduğu) gözlemlenirse, yöntem blokta hata olduğunu ifade edecek şekilde geriye false değeri döner (bu durumda protokol hata düzeltme adımına geçer, bizim çalışmamızda Yeni BINARY tekniği çalıştırılır).
- Parite kontrolü sonucu paritelerin uyuştuğu (blokta hatalı bit olmadığı ya da çift sayıda hatalı bit olduğu) gözlemlenirse, bu durumda tamamen bilinen bitler olarak kaydedilebilecek bit olup olmadığı kontrol edilir:

- Eğer bloğun uzunluğu bir ise, bu bit tamamen bilinen bir bit olarak işaretlenir ve hafızaya kaydedilir.
- Bloğun uzunluğu birden büyükse ve bloktaki bir bit dışındaki diğer tüm bitler tamamen bilinen bitlerden oluşturulabiliyorsa ya da pariteleri paritesi bilinen bloklardan elde edilebiliyorsa, bu durumda o bir bit de tamamen bilinen bit olarak işaretlenip hafızaya alınır.

- Yeni Parite Kontrolü yöntemi geriye paritenin uyuştuğunu (blokta hatalı bit sezilmediğini) ifade eden true değerini döner.

Tablo 4.4. Yeni Binary algoritması

Yeni Binary
1: YeniBinary(BLOK blok)
2: *i. Binary'ye Başlamadan Önce*:
3: ▷ Parite değiş tokuş miktarını minimize etmek için hafızadaki bilgiler kullanılarak blok uzunluğu olabildiğince azaltılmaya çalışılır.
4: *kucuk_blok ← paritesi_bilinen_bloklari_cikar(block)*
5: *daha_kucuk_blok ← tamamen_bilinen_bitleri_cikar(kucuk_blok)*
6:
7: *ii. Binary'yi Çalıştır*:
8: ▷ Blok uzunluğu olabildiğince azaltıldıktan sonra, standart Binary tekniği çalıştırılır ve yeni tamamen bilinen bitler hafızaya kaydedilir.
9: *duzeltilen_bit_indisi ← Binary(daha_kucuk_blok)*
10:
11: ▷ Orjinal Binary tekniğinde sadece bir bit düzeltilir ve onun indisi dönülür.
12: **return** *duzeltilen_bit_indisi*

Tablo 4.4'te Yeni BINARY tekniği verilmiştir. Bu tekniğin temel prensibi hata düzeltme tekniği olan BINARY'nin üzerinde çalışacağı blok uzunluğunu olabildiğince azaltmaktır ve böylece protokolün verimlilik performansını arttırmaktır. Algoritma şöyle çalışmaktadır:

- Öncelikle, blok üzerinde arama yapılır ve daha önce hafızaya alınmış olan paritesi bilinen blokların blok içinde olup olmadığı kontrol edilir ve olanlar varsa bloktan çıkarılır.
- Ardından, kalan küçük blok üzerinde arama yapılır ve daha önce hafızaya alınmış olan tamamen bilinen bitlerin blok içinde olup olmadığı kontrol edilir ve olanlar varsa bloktan çıkarılır.
- Daha sonra, elde edilen bu daha küçük blok üzerinde standart BINARY tekniği çalıştırılır, hatalı bit düzeltilir ve tespit edilen yeni tamamen bilinen bitler hafızaya kaydedilir.
- Son olarak yöntem standart BINARY tekniğinde olduğu gibi düzeltilmiş bitin indis bilgisini geriye döner.

Bu tez çalışmasında, [23]'te ele alındığı gibi BINARY sonucu oluşan küçük bloklar hafızaya alınmaktadır. Ancak, bu çalışmada [23]'ten farklı olarak BINARY'nin uygulanmasında farklılıklar vardır:

- BINARY'nin çalışacağı blok uzunluğunun hafızaya daha önceden kaydedilmiş olan bilgileri kullanarak azaltılabilir,
- BINARY sırasında hatası düzeltilmiş bitlerin sayısı da farklıdır.

Aslında gerçekte yine bir bit düzeltilir, ancak tamamen bilinen bitler olarak işaretlenen bitlerin sayısı bloğun uzunluğu ve/veya bitin blok içindeki konumuna göre iki veya üç olabilmektedir. [23]'teki çalışmada ve literatürde şu ana kadar bahsi geçen tekniklerde böyle bir yaklaşım görülmemiştir.

Sonuç olarak, bu çalışmada önerilen Yeni Parite Kontrolü ve Yeni BINARY teknikleri daha az parite kontrolü gerçekleştirir ve böylece yeni protokolün verimlilik performansını önemli oranda iyileştirir/arttırır [9]. Yeni Parite Kontrolü bazı parite kontrollerinin yapılmamasını sağlarken, Yeni BINARY tekniği ise hem bazı parite kontrollerinin yapılmamasını sağlar hem de daha az sayıda parite kontrolü yapar. Orijinal CASCADE protokolündeki bu iki temel adım da iyileştirilmiş ve iyileştirilmeleri içeren geliştirilmiş yeni protokol Tablo 4.5'teki gibi önerilmiştir. Bu yeni protokole optimum CASCADE adı verilmiştir [9].

Tablo 4.5. Bu tez çalışmasında önerilen optimum CASCADE protokolünün algoritması

Optimum Cascade protokolünün algoritması

```
1: OPTIMUM_CASCADE(BİTDİZİSİ BITDIZISIOKUYUCU)
2: i. Ilklendirme:
3:     k_1 ← 0.73/ε
4:     k_i ← 2k_{i-1}   2 ≤ i ≤ Raund
5:     Raund ← 4
6:
7: ii. Raund i = 1:
8:     for l = 0;  l < ⌈N/k_1⌉;  l++  do
9:         gondericininParitesi ← pariteGetir(gondericiBlogu[l]);
10:        if pariteKontrol(aliciBlogu[l]) != gondericininParitesi then
11:            Binary(aliciBlogu[l]);
12:
13: iii. Raund i > 1:
14:    karıştır(bitDizisiOkuyucu)
15:    for l = 0;  l < ⌈N/k_i⌉;  l++  do
16:        ▷ Parite kontrolü ve değiş tokuşu Yeni Parite Kontrolü yöntemi içinde sadece gerekli
   ise yapılır.
17:        if yeniPariteKontrolu(alicininBlogu[l]) == false then
18:            duzeltilenBit ← Binary(aliciBlogu[[l]);
19:            geriyeIzSur(duzeltilenBit);
```

İlk raundda hafızada kayıtlı bilgiler bulunmamaktadır. Diğer bir deyişle, ilk raundda hata sezme ve düzeltme işlemleri sonucu tamamen bilinen bitler ve paritesi bilinen bloklar hafızaya kaydedilmeye başlanmıştır. İkinci raunddan itibaren Yeni Parite Kontrolü ve Yeni BINARY teknikleri uygulanmaya başlamıştır. Ayrıca, Yeni

BINARY tekniği geriye iz sürme adımında da uygulanmaktadır. İkinci raunddan itibaren algoritmada değişiklikler görülmektedir. Tablo 4.1 ve Tablo 4.5 karşılaştırıldığında satır 16'da değişiklik olduğu görülmektedir. Orijinal protokolde gönderici ve alıcı arasında peşinen bir parite kontrolü yapılırken, yeni protokolde bu adımın yerini Yeni Parite Kontrolü adımı almıştır. Diğer bir deyişle, peşinen parite kontrolü yapmak yerine gerekiyorsa parite kontrolü yapılacaktır ve hatta gerekmiyorsa yapılmayacaktır. Ayrıca, yine ikinci raunddan itibaren BINARY tekniği yerine de Yeni BINARY tekniği kullanılmaktadır.

Bu bölümde bahsi geçen tüm iyileştirmelerin verimlilik performansına katkısını ölçmek amacıyla kapsamlı simülasyonlar yapılmıştır. Bu tez çalışmasında, önerilen iyileştirmelerin sonuçlarını daha iyi görmek için, her bir iyileştirme öncelikle tek tek ele alınmış ve ardından bütün iyileştirmeler birlikte de incelenmiştir.

Öncelikle, [23]'te anlatılan CASCADE protokolü gerçeklenmiştir ve simülasyonlar bu referans protokol ile çalıştırılmıştır. Referans protokol, [23]'te olandan halihazırda daha verimlidir.

Daha sonra TBB ve PBB iyileştirmeleri sırasıyla referans CASCADE protokolüne eklenerek verimlilik performansının değişimi izlenmiştir. Her bir iyileştirme protokole ayrı ayrı eklenmiş ve her biri için simülasyonlar çalıştırılmıştır.

Son olarak, bütün iyileştirmeler referans protokole uygulanmış ve nihai hali de simülasyonlara sokulmuştur. Tüm simülasyon sonuçları Tablo 4.6, Şekil 4.11 ve Şekil 4.12'de verilmiştir. Simülasyonlarda gizli anahtarın uzunluğu N = 10000 olarak seçilmiştir ve elde edilen μ verimlilik değerleri 100 başarılı denemenin ortalaması olarak verilmiştir.

Tablo 4.6. Bu çalışmada önerilen TBB, PBB ve optimum CASCADE yeniliklerin verimlilik performansına etkisi

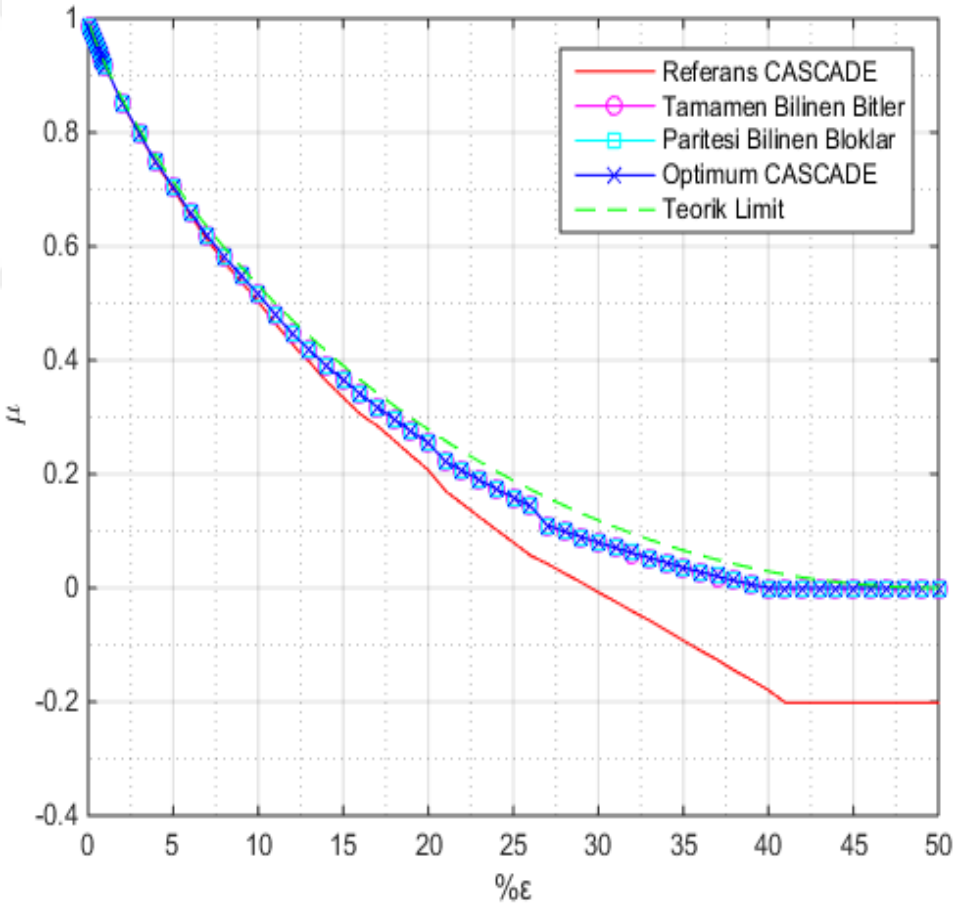
ε (%)	**Referans Protokol**	**TBB**	**PBB**	**Optimum CASCADE**	**Teorik Limit**
0.1	0.986680	0.986683	0.987586	0.987584	0.988592
0.2	0.976969	0.976971	0.977887	0.977882	0.979186
0.3	0.968148	0.968169	0.969059	0.969065	0.970536
0.4	0.959600	0.959616	0.960510	0.960512	0.962378

Tablo 4.6. (Devam) Bu çalışmada önerilen TBB, PBB ve optimum CASCADE yeniliklerin verimlilik performansına etkisi

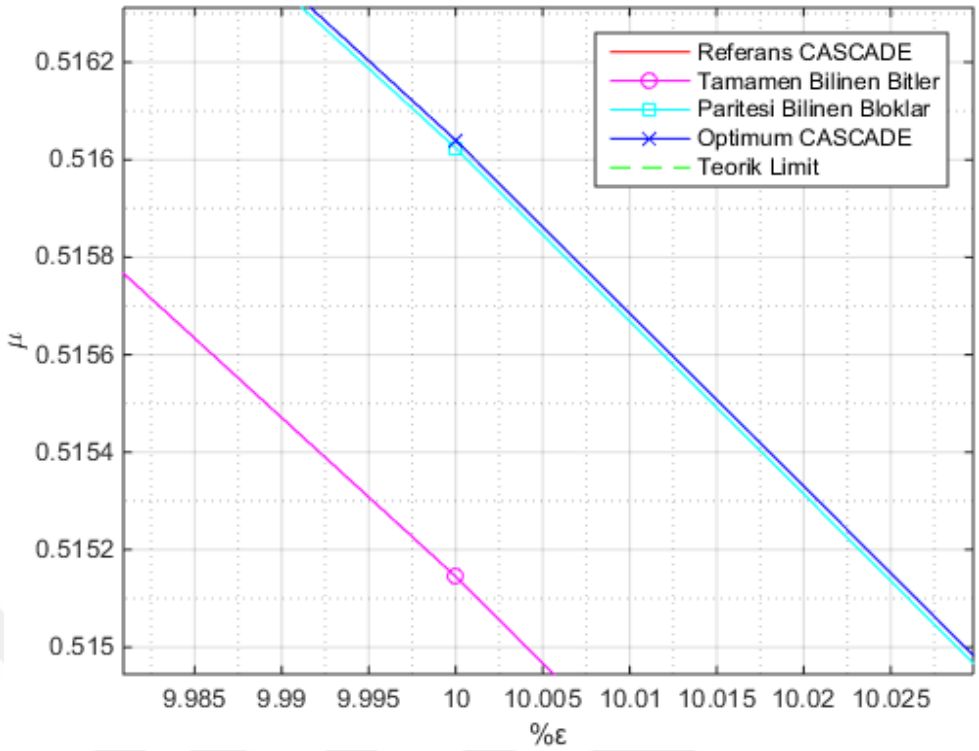
ε (%)	**Referans Protokol**	**TBB**	**PBB**	**Optimum CASCADE**	**Teorik Limit**
0.5	0.951442	0.951489	0.952372	0.952376	0.954585
0.6	0.943895	0.943976	0.944869	0.944874	0.947085
0.7	0.936403	0.936489	0.937386	0.937386	0.939828
0.8	0.929007	0.929087	0.929979	0.929974	0.932778
0.9	0.921653	0.921769	0.922668	0.922668	0.925912
1.0	0.914741	0.914867	0.915780	0.915781	0.919207
2.0	0.851357	0.851877	0.852782	0.852787	0.858559
3.0	0.796456	0.797505	0.798419	0.798424	0.805608
4.0	0.744691	0.746966	0.747874	0.747887	0.757708
5.0	0.700615	0.703436	0.704356	0.704362	0.713603
6.0	0.654559	0.658589	0.659498	0.659508	0.672555
7.0	0.611517	0.617863	0.618772	0.618784	0.634076
8.0	0.571386	0.580440	0.581306	0.581320	0.597821
9.0	0.538127	0.547720	0.548633	0.548644	0.563530
10.0	0.502952	0.515145	0.516024	0.516040	0.531004
11.0	0.464864	0.479586	0.480498	0.480516	0.500084
12.0	0.428023	0.445775	0.446682	0.446700	0.470639
13.0	0.396600	0.417551	0.418471	0.418489	0.442562
14.0	0.363401	0.389532	0.390463	0.390483	0.415761
15.0	0.334283	0.364795	0.365725	0.365737	0.390160
16.0	0.305174	0.340216	0.341166	0.341198	0.365690
17.0	0.284665	0.316941	0.317842	0.317859	0.342295
18.0	0.258984	0.295733	0.296629	0.296671	0.319923
19.0	0.232501	0.274114	0.275063	0.275089	0.298529
20.0	0.207149	0.254180	0.255078	0.255107	0.278072
21.0	0.170361	0.221583	0.222475	0.222495	0.258517
22.0	0.147775	0.205291	0.206193	0.206229	0.239832
23.0	0.124677	0.188770	0.189757	0.189780	0.221989
24.0	0.102169	0.172989	0.173926	0.173944	0.204960
25.0	0.079999	0.158109	0.159013	0.159041	0.188722
26.0	0.057169	0.143084	0.144027	0.144071	0.173254
27.0	0.042302	0.108853	0.109802	0.109806	0.158535
28.0	0.025700	0.098656	0.099526	0.099532	0.144549
29.0	0.009474	0.088714	0.089694	0.089699	0.131279
30.0	-0.007620	0.079190	0.080051	0.080059	0.118709
31.0	-0.023989	0.070002	0.070816	0.070824	0.106827
32.0	-0.041069	0.060927	0.061879	0.061889	0.095619
33.0	-0.056930	0.052179	0.052944	0.052956	0.085074
34.0	-0.074462	0.043624	0.044561	0.044571	0.075181
35.0	-0.092470	0.034850	0.035772	0.035780	0.065932
36.0	-0.109719	0.027300	0.028210	0.028225	0.057317
37.0	-0.126533	0.019654	0.020522	0.020547	0.049328
38.0	-0.144898	0.012398	0.013245	0.013284	0.041958

Tablo 4.6. (Devam) Bu çalışmada önerilen TBB, PBB ve optimum CASCADE yeniliklerin verimlilik performansına etkisi

ε (%)	Referans Protokol	TBB	PBB	Optimum CASCADE	Teorik Limit
39.0	-0.161820	0.005445	0.006409	0.006448	0.035200
40.0	-0.179584	-0.001103	-0.000160	-0.000109	0.029049
41.0	-0.201600	0.000000	0.000000	0.000000	0.023500
42.0	-0.201600	0.000000	0.000000	0.000000	0.018546
43.0	-0.201600	0.000000	0.000000	0.000000	0.014185
44.0	-0.201600	0.000000	0.000000	0.000000	0.010412
45.0	-0.201600	0.000000	0.000000	0.000000	0.007226
46.0	-0.201600	0.000000	0.000000	0.000000	0.004622
47.0	-0.201600	0.000000	0.000000	0.000000	0.002598
48.0	-0.201600	0.000000	0.000000	0.000000	0.001154
49.0	-0.201600	0.000000	0.000000	0.000000	0.000289
50.0	-0.201600	0.000000	0.000000	0.000000	0.000000



Şekil 4.11. İyileştirmelerin verimlilik performansı üzerindeki etkileri



Şekil 4.12. İyileştirmelerin verimlilik performansı üzerindeki etkileri (Şekil 4.11'deki bir bölüm büyütülmüştür)

Bu simülasyonlarda, optimum CASCADE ile ifade edilen protokol önerilen bütün iyileştirmeleri içermektedir. Bu protokol, [23]'te önerilen yöntemin üzerine Bölüm 4.2.1 ve 4.2.2'de anlatılan TBB ve PBB iyileştirmelerini eklemektedir. Tablo 4.6, Şekil 4.11 ve Şekil 4.12'deki sonuçlar da [23]'teki parametre kümesi kullanılarak elde edilen sonuçları göstermektedir.

Tablo ve şekillerden de görülebileceği üzere, Bölüm 4.2.1 ve Bölüm 4.2.2'deki iyileştirmelerin her biri referans CASCADE protokolünün verimliliğini önemli oranda arttırmıştır. Bu iyileştirmelerle teorik limite daha çok yaklaşılmıştır, ancak halen tam olarak ulaşılamamıştır. Bu da protokolde halen yapılabilecek iyileştirmelerin olduğunu göstermektedir. Öncelikle, yenilikler referans protokole sırasıyla, daha sonra ise hep birlikte uygulanmıştır. Tek başına TBB ve PBB de verimlilik performansını oldukça arttırmıştır. Şekil 4.11'den de görüldüğü gibi tüm iyileştirmelerin yaptığı katkı birbirine çok yakın olmakla birlikte hepsi referans CASCADE protokolünden oldukça fazladır. Her bir iyileştirmenin getirdiği katkı birbirine yakın olup, PBB iyileştirmesi Şekil 4.12'den de görüldüğü gibi TBB iyileştirmesinin sunduğu katkıya oranla daha fazla olmaktadır. Bunun sebebi ise, PBB iyileştirmesinin BINARY koşturulacak olan

bloktan blok seviyesinde çıkarma yapmasıdır. Oysa ki, TBB ise bit seviyesinde çıkarma yapmaktadır. Yine Şekil 4.12'den görüldüğü gibi optimum CASCADE protokolü tüm iyileştirmeleri içerdiği için PBB iyileştirmesinden gelen sonuçlara yakın değerler üretse de en başarılı verimlilik sonuçlarına sahip olmaktadır. Optimum CASCADE ile PBB iyileştirmesinin yakın değerler içermesi ikisinin de aramalar sonucunda birbirlerine yakın/ortak bilgileri (tamamen bilinen bit ve/veya paritesi bilinen blok) bloktan çıkarmalarıyla ilgilidir.

Bu tez çalışması kapsamında yapılan araştırmalarda görülebildiği kadarıyla bu çalışmada önerilen optimum CASCADE tekniği literatürdeki tüm CASCADE versiyonlarından daha başarılı verimlilik sonuçları üretmektedir [9]. Bu durum Bölüm 5'te daha detaylı ele alınmış olup, önerilen iddia kapsamlı deney ve simülasyonlarla desteklenmiştir. Deney sonuçları da geniş bir kanal hata olasılığı spektrumunda incelenmiştir ve literatürde bu seviyede bir incelemede bulunulmamıştır [9].

5. DENEY VE ANALİZLER

Bu bölümde optimum CASCADE protokolü verimlilik açısından kapsamlı bir şekilde incelenmiştir. Bu kapsamda, öncelikle literatürdeki şu an itibariyle en başarılı olan [24]'te verilen CASCADE tekniği ile, ardından diğer CASCADE teknikleri ve CASCADE olmayan tekniklerle karşılaştırılmıştır. Her karşılaştırma için kapsamlı simülasyonlar çalıştırılmıştır ve sonuçları tablo ve şekillerle sunulmaya çalışılmıştır. Bu noktada vurgulanması gereken önemli bir husus şu olabilir: bu tez çalışmasının öncelikle amacı CASCADE protokolünün verimlilik performansını arttıracak yöntemler bulmaktır. Diğer bir deyişle, önerilen yöntem öncelikle CASCADE standartlarının dışına çıkmayacaktır ve literatürdeki CASCADE versiyonlarıyla kendisini karşılaştıracaktır. Önerilen yöntem, literatürdeki tüm CASCADE versiyonlarının parametre kümesi ile çalışabilmektedir [9]. Bununla birlikte, önerilen yöntem elde ettiği verimlilik sonuçlarını CASCADE olmayan ve literatürde BU maksadıyla kullanılan diğer yöntemlerle de karşılaştırmaktadır. Bu karşılaştırmaların her biri için kapsamlı deney ve simülasyonlar çalıştırılmıştır ve sonuçları bu bölüm altında verilmiştir.

CASCADE protokolü için literatürde temel olarak iki performans ölçütü üzerinde çalışmaların yoğunlaştığı daha önce de ifade edilmişti. Bunlar: verimlilik ve hız. Bu çalışmaların her biri ayrı bir doktora çalışması olabilecek niteliktedir. Bu tez çalışmasında ise öncelikle verimlilik konusunda yoğunlaşılmıştır. Literatürdeki popüler CASCADE yöntemleri araştırılmış, karşılaştırmaların sağlıklı ve doğru bir şekilde yapılabilmesi için yazılımsal olarak gerçeklenmişlerdir. Yapılan karşılaştırmalarda her bir iyileştirmenin verimlilik performansına katkısı da kapsamlı bir şekilde ele alınmıştır. Yapılan performans iyileştirmelerinde çoğu zaman bir performans ölçütünden kazanç sağlarken bir diğerinden kaybedilebilmektedir. Bu çalışmada da verimlilik performansının iyileştirilmesinin bir diğer performans ölçütü olan hız üzerindeki etkisi de incelenmiştir. Bu maksatla, verimlilik için yapılan analizlerden sonra hız açısından da kapsamlı deneyler ve simülasyonlar yapılmıştır.

Bu sonuçlar ışığında, CASCADE protokolüne yapılan her iyileştirme verimliliği arttırırken hızdan kaybettirmektedir. Bu nedenle iyileştirmelerin hız üzerindeki etkileri kapsamlı bir şekilde incelenmiştir ve sonuçları tablo ve şekillerle bu bölüm altında sunulmuştur.

5.1. Verimlilik

Bu bölümde, Tablo 4.5'te detayı verilen optimum CASCADE protokolü literatürdeki popüler/öncü BU teknikleriyle karşılaştırılmıştır.

Mevcut durumda literatürdeki CASCADE protokolleri arasında en verimli olan teknik [24]'te verilmiştir. Bu çalışmanın yazarları [23]'te önerilmiş olan CASCADE protokolünü referans almışlardır ve sadece protokolün kullandığı parametre kümesini değiştirmişlerdir. [24]'ün yazarları çalışmalarında CASCADE protokolünü birçok açıdan analiz etmişler, literatürde önerilmiş olan birçok iyileştirmeyi de incelemişler ve sonuçları detaylı bir şekilde sunmuşlardır. Ancak, nihai protokol olarak [23]'te önerilen CASCADE protokolünü kendilerinin önerdikleri yeni parametre kümesiyle birleştirerek elde etmişlerdir ve simülasyonlarını bu nihai protokol ile çalıştırmışlardır. Yazarlar çalışmalarında, yeni parametre kümesini bulabilmek için iki boyutlu arama algoritması olan compass (pusula) arama yöntemini kullanmışlardır ve bu parametrelerin de optimum olduğunu ifade etmişlerdir. Ayrıca, yazarlar [23]'te önerilen CASCADE protokolünün verimliliğini arttırmak için [84]'te de önerildiği gibi blok uzunlukları ve gizli anahtar uzunluğunu ikinin üstel katı olarak seçmişlerdir.

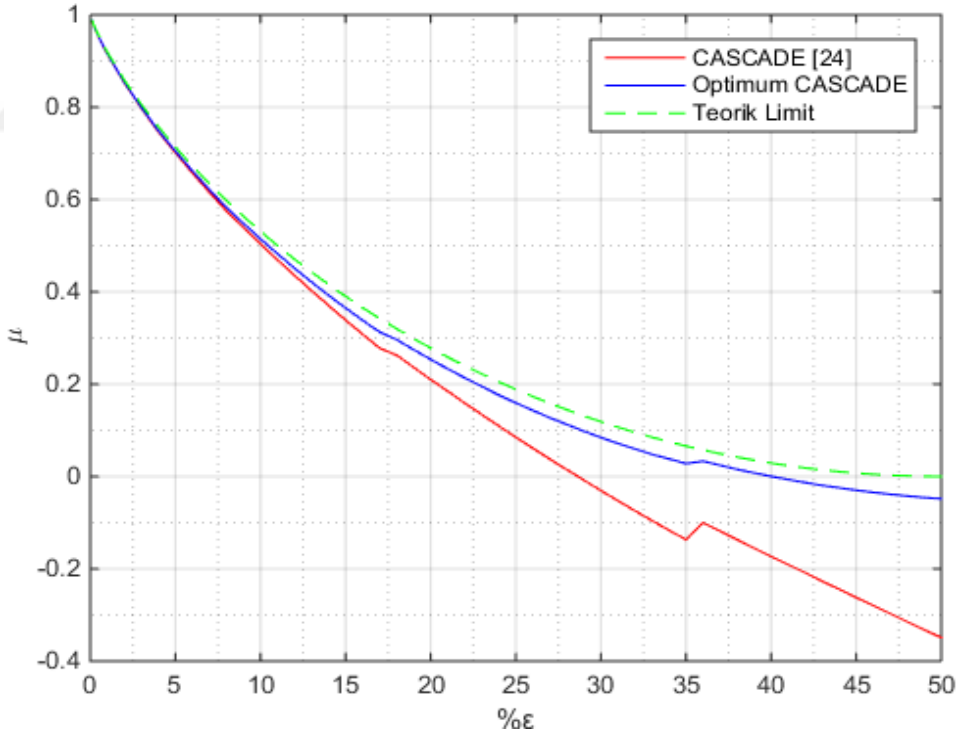
Bu çalışmada da referans protokol olarak [23]'te önerilen CASCADE protokolü seçilmiştir. Ancak, [24]'tekinin aksine protokolde önemli iyileştirmeler yapılmıştır. Bölüm 4.2.1, 4.2.2 ve 4.3'te bahsedilen iyileştirmelerin tamamı [23]'te önerilen CASCADE protokolüne eklenmiştir. Protokolün bu yeni haline ise optimum CASCADE adı verilmiştir [9]. Optimum CASCADE ile literatürde şu an en iyi sonuçlara sahip olan [24]'teki protokolü karşılaştırmak için yine [24]'te önerilen yeni parametre kümesi kullanılmıştır ve kapsamlı simülasyonlar çalıştırılmıştır. Simülasyon sonuçları ise sırasıyla Tablo 5.1'de ve Şekil 5.1'de verilmiştir. Bu simülasyonlarda gizli anahtar uzunluğu ikinin üstel katı olacak şekilde $N = 2^{14}$ olarak seçilmiştir ve μ verimlilik değerleri ise 100 başarılı denemenin ortalaması olarak ele alınmıştır (FER = BER = 0).

Tablo 5.1. Optimum CASCADE ile [24]'te önerilen CASCADE protokolünün karşılaştırılması

ε (%)	k_1	k_2	k_3	k_i i > 3	CASCADE [24]	Optimum CASCADE	Teorik Limit
0.1	1024	2048	4096	8192	0.986396	0.987191	0.988592
0.2	512	2048	4096	8192	0.977033	0.977827	0.979186
0.3	256	1024	4096	8192	0.967891	0.968693	0.970536
0.4	256	1024	4096	8192	0.959799	0.960601	0.962378
0.5	256	1024	4096	8192	0.951875	0.952684	0.954585
0.6	128	1024	4096	8192	0.943914	0.944755	0.947085
0.7	128	1024	4096	8192	0.936761	0.937601	0.939828
0.8	128	1024	4096	8192	0.929141	0.929964	0.932778
0.9	128	1024	4096	8192	0.922217	0.923032	0.925912
1.0	128	1024	4096	8192	0.915266	0.916136	0.919207
2.0	64	512	4096	8192	0.852666	0.853860	0.858559
3.0	32	512	4096	8192	0.797791	0.799321	0.805608
4.0	32	512	4096	8192	0.746494	0.748843	0.757708
5.0	16	256	4096	8192	0.701508	0.704431	0.713603
6.0	16	256	4096	8192	0.657473	0.661687	0.672555
7.0	16	256	4096	8192	0.615295	0.621019	0.634076
8.0	16	256	4096	8192	0.575132	0.582579	0.597821
9.0	8	256	4096	8192	0.539786	0.548116	0.563530
10.0	8	256	4096	8192	0.503978	0.514454	0.531004
11.0	8	256	4096	8192	0.469187	0.482297	0.500084
12.0	8	256	4096	8192	0.435433	0.451057	0.470639
13.0	8	256	4096	8192	0.402509	0.421076	0.442562
14.0	8	256	4096	8192	0.370294	0.392300	0.415761
15.0	8	256	4096	8192	0.338575	0.364851	0.390160
16.0	8	256	4096	8192	0.308132	0.338162	0.365690
17.0	8	256	4096	8192	0.278050	0.312886	0.342295
18.0	4	128	4096	8192	0.262582	0.296356	0.319923
19.0	4	128	4096	8192	0.236360	0.274636	0.298529
20.0	4	128	4096	8192	0.209905	0.253472	0.278072
21.0	4	128	4096	8192	0.184655	0.233186	0.258517
22.0	4	128	4096	8192	0.159028	0.213691	0.239832
23.0	4	128	4096	8192	0.133972	0.194992	0.221989
24.0	4	128	4096	8192	0.109268	0.176470	0.204960
25.0	4	128	4096	8192	0.085200	0.159261	0.188722
26.0	4	128	4096	8192	0.061208	0.143148	0.173254
27.0	4	128	4096	8192	0.037578	0.127156	0.158535
28.0	4	128	4096	8192	0.014684	0.112599	0.144549
29.0	4	128	4096	8192	-0.007887	0.098067	0.131279
30.0	4	128	4096	8192	-0.030673	0.084451	0.118709
31.0	4	128	4096	8192	-0.052515	0.071548	0.106827
32.0	4	128	4096	8192	-0.074318	0.059682	0.095619
33.0	4	128	4096	8192	-0.095596	0.047792	0.085074
34.0	4	128	4096	8192	-0.116468	0.038153	0.075181

Tablo 5.1. (Devam) Optimum CASCADE ile [24]'te önerilen CASCADE protokolünün karşılaştırılması

ε (%)	k_1	k_2	k_3	k_i i > 3	CASCADE [24]	Optimum CASCADE	Teorik Limit
35.0	4	128	4096	8192	-0.136808	0.028014	0.065932
36.0	2	128	4096	8192	-0.100363	0.033124	0.057317
37.0	2	128	4096	8192	-0.118495	0.024321	0.049328
38.0	2	128	4096	8192	-0.136697	0.015423	0.041958
39.0	2	128	4096	8192	-0.155322	0.007724	0.035200
40.0	2	128	4096	8192	-0.173522	0.000154	0.029049
41.0	2	128	4096	8192	-0.190926	-0.006601	0.023500
42.0	2	128	4096	8192	-0.208455	-0.013143	0.018546
43.0	2	128	4096	8192	-0.227036	-0.019379	0.014185
44.0	2	128	4096	8192	-0.244374	-0.024594	0.010412
45.0	2	128	4096	8192	-0.262417	-0.029858	0.007226
46.0	2	128	4096	8192	-0.279576	-0.034459	0.004622
47.0	2	128	4096	8192	-0.297084	-0.038535	0.002598
48.0	2	128	4096	8192	-0.314844	-0.042215	0.001154
49.0	2	128	4096	8192	-0.332173	-0.045667	0.000289
50.0	2	128	4096	8192	-0.349032	-0.048260	0.000000



Şekil 5.1. Optimum CASCADE ile [24]'te önerilen CASCADE protokolünün karşılaştırılması

Tablo 5.1 ve Şekil 5.1'deki sonuçlardan da görülebildiği üzere, bu çalışmada önerilen optimum CASCADE protokolü [24]'te önerilen CASCADE protokolüne göre verimlilik performansı olarak daha iyi sonuçlar üretmiştir [9]. Diğer bir deyişle,

verimliliğin teorik üst limitine daha fazla yaklaşılmıştır. Ancak halen tam olarak ulaşılamadığı için protokolde daha iyileştirilebilecek noktalar olduğu da anlaşılabilmektedir. Bu aşamada, optimum CASCADE protokolünün literatürdeki en iyi sonuçları veren CASCADE protokollerinden daha iyi verimlilik sonuçları ürettiği söylenebilir [9].

Tablo 5.1 ve Şekil 5.1'den de görülebildiği gibi optimum CASCADE tekniği tüm ε değerleri için daha iyi verimlilik değerleri üretmiştir. Şekil 5.1'den de görülebildiği gibi özellikle kanal hata olasılığı ε = %11'den itibaren optimum CASCADE protokolü teorik limite yakın değerler üretmektedir. Kanal hata olasılığının ε = % 29 ve/veya ε = % 30 gibi değerlerinden sonra verimlilik değerlerinin negatif değerler aldığı görülmektedir. Bu da protokolün bu hata olasılıklarında aşırı verimsiz çalıştığını ifade etmektedir. Denklem (2.17)'deki ifade üzerinden yorumlanırsa, N bitlik bir mesaj üzerinde hata sezme ve düzeltme işlemi gerçekleştirmek için E > N kadar ilave bilgi miktarı gerekmektedir. Zaten kanal hata olasılığının bu değerlere ulaştığı durumlar için teorik limit de düşük değerlere inmektedir. Optimum CASCADE protokolü her ne kadar negatif değerler üretse de teorik limitten yine çok fazla ayrışmamıştır.

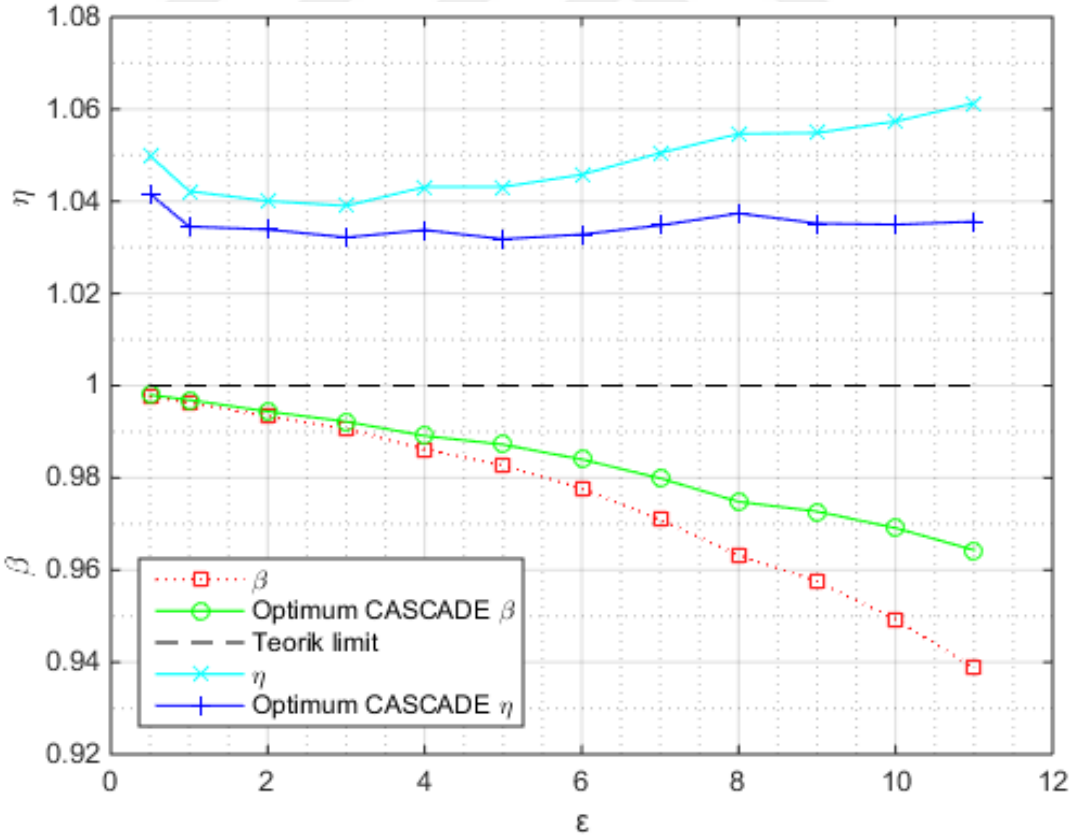
Daha önce de bahsedildiği gibi adil bir karşılaştırma olması açısından [24]'te önerilen CASCADE protokolünün parametre kümesi aynen kullanılmıştır. Optimum olduğu öne sürülen blok uzunlukları Tablo 5.1'de sunulmuştur ve [24]'te verilen formüllere göre hesaplanmıştır. Bu formüller bu tez çalışmasının 4.1.2 bölümünde de sunulmuştur.

[24]'te optimum blok uzunluklarının hesaplanması için bir formül kümesi verilmiştir, ancak nihari simülasyon sonuçlarında bu formülden elde edilen sonuçlara tamamen uyulmadığı görülmüştür. Yazarlar, en iyi verimlilik performansını elde etmek için formüllerden elde edilen bazı blok uzunluklarını değiştirmişlerdir. Bu nedenle, yazarların en iyi sonuçlarını elde ettikleri parametre kümesini aynı şekilde kullanarak simülasyonlar tekrarlanmıştır ve elde edilen sonuçlar Tablo 5.2 ve Şekil 5.2'de gösterilmiştir. Yazarlar simülasyonlarını 10^5 kez tekrar etmişler ve verimlilik değerlerini bu simülasyonların ortalama değerleri olarak vermişlerdir. Yazarlar, bu tez çalışmasında şu ana kadar tercih edilen μ verimlilik formülü yerine β ve η formüllerini kullanmayı tercih etmişlerdir. Ayrıca, yazarlar gizli anahtar uzunluğu olarak ikinin üstel katı olacak şekilde $N = 2^{14}$ seçmişlerdir. Adil bir karşılaştırma olması açısından

yazarların yaptığı tüm tercihler olduğu gibi kabul edilmiştir ve simülasyonlar çalıştırılmıştır. Sonuçlar Tablo 5.2 ve Şekil 5.2'de gösterilmiştir.

Tablo 5.2. Optimum CASCADE ile [24]'te önerilen CASCADE protokolünün β ve η verimlilik performanslarının karşılaştırılması (η, [24]'te f_{EC} olarak gösterilmiştir)

ε (%)	k_1	k_2	k_3	β	Optimum CASCADE β	η	Optimum CASCADE η
0.5	256	1024	4096	0.9976	0.9980	1.04989	1.04158
1.0	128	512	4096	0.9963	0.9969	1.04219	1.03452
2.0	64	512	4096	0.9934	0.9944	1.04006	1.03389
3.0	32	512	4096	0.9906	0.9922	1.03902	1.03217
4.0	32	256	4096	0.9862	0.9891	1.04313	1.03378
5.0	16	256	4096	0.9827	0.9872	1.04313	1.03180
6.0	16	256	4096	0.9777	0.9840	1.04580	1.03282
7.0	16	256	4096	0.9709	0.9799	1.05050	1.03478
8.0	8	256	4096	0.9632	0.9748	1.05465	1.03740
9.0	8	256	4096	0.9575	0.9727	1.05486	1.03520
10.0	8	256	4096	0.9493	0.9691	1.05736	1.03496
11.0	8	256	4096	0.9387	0.9643	1.06130	1.03561

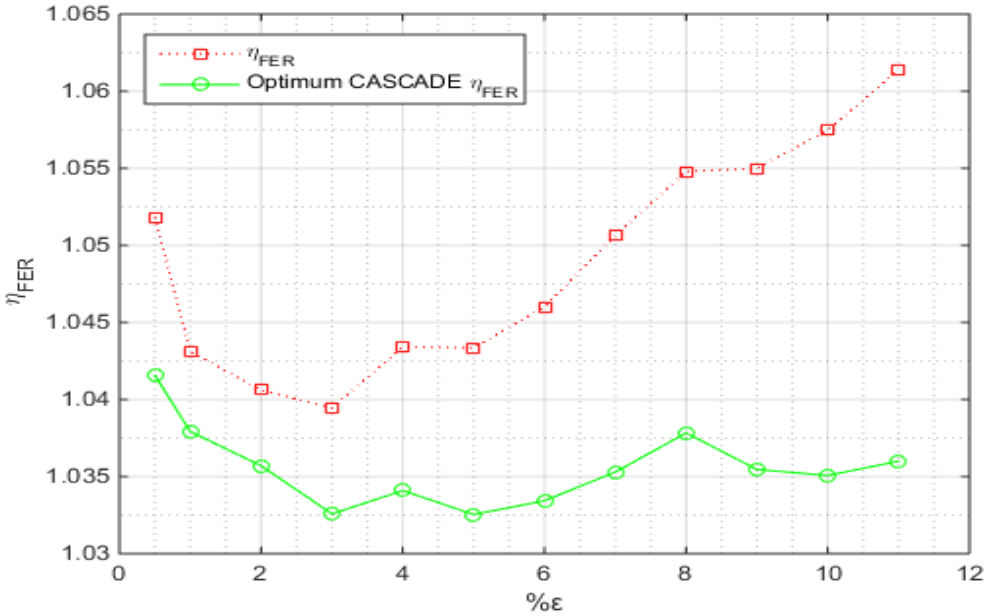


Şekil 5.2. Optimum CASCADE ile [24]'te önerilen CASCADE protokolünün β ve η verimlilik performanslarının karşılaştırılması (η, [24]'te f_{EC} olarak gösterilmiştir)

[24]'te, verimlilik değerlerinin doğruluğu da ele alınmış ve yüksek FER değerleri altında çok sayıda mesaj dikkate alınmadığı için elde edilen yüksek verimlilik değerlerinin çok gerçekçi olmadığı öne sürülmüştür. Bu bağlamda, elde edilen verimlilik değerleri Bölüm 2.3'te verilen ifadeler kullanılarak ve FER değerleri de hesaba katılarak normalize edilmiştir. Sonuçlar sırasıyla Tablo 5.3 ve Şekil 5.3'te verilmiştir.

Tablo 5.3. Optimum CASCADE ile [24]'te önerilen CASCADE protokolünün FER ve η_{FER} değerlerinin karşılaştırılması (η_{FER} , [24]'te η_{EC} olarak gösterilmiştir)

ε (%)	k_1	k_2	k_3	*FER*	Optimum CASCADE *FER*	η_{FER}	Optimum CASCADE η_{FER}
0.5	256	1024	4096	$9.2\text{x}10^{-5}$	0	1.05182	1.04158
1.0	128	512	4096	$8.0\text{x}10^{-5}$	$3\text{x}10^{-4}$	1.04310	1.03792
2.0	64	512	4096	$9.3\text{x}10^{-5}$	$3\text{x}10^{-4}$	1.04062	1.03570
3.0	32	512	4096	$1.1\text{x}10^{-4}$	$1\text{x}10^{-4}$	1.03945	1.03258
4.0	32	256	4096	$9.4\text{x}10^{-5}$	$1\text{x}10^{-4}$	1.04342	1.03409
5.0	16	256	4096	$8.9\text{x}10^{-5}$	$3\text{x}10^{-4}$	1.04335	1.03254
6.0	16	256	4096	$1.1\text{x}10^{-4}$	$3\text{x}10^{-4}$	1.04601	1.03343
7.0	16	256	4096	$8.7\text{x}10^{-5}$	$3\text{x}10^{-4}$	1.05065	1.03529
8.0	8	256	4096	$9.7\text{x}10^{-5}$	$3\text{x}10^{-4}$	1.05479	1.03783
9.0	8	256	4096	$1.0\text{x}10^{-4}$	$2\text{x}10^{-4}$	1.05499	1.03546
10.0	8	256	4096	$1.0\text{x}10^{-4}$	$1\text{x}10^{-4}$	1.05747	1.03507
11.0	8	256	4096	$1.0\text{x}10^{-4}$	$4\text{x}10^{-4}$	1.06139	1.03600



Şekil 5.3. Optimum CASCADE ile [24]'te önerilen CASCADE protokolünün FER ve η_{FER} değerlerinin karşılaştırılması (η_{FER} , [24]'te η_{EC} olarak gösterilmiştir)

Tablo 5.2'deki ve Tablo 5.3'teki optimum CASCADE ile başlayan sütunlar haricindeki tüm değerler [24]'teki çalışmanın Tablo 3'ünden alınmıştır. Yazarların nihai simülasyon sonuçlarında formülden elde ettikleri blok uzunluklarını değiştirdikleri daha önce de ifade edilmişti. Tablo 5.1'deki değerler ise [24]'teki çalışmada önerilen formüllere göre hesaplanmıştır. Tablolar karşılaştırıldığında, ε = %1 için $k_2\,(1024 \rightarrow 512)$, ε = %4 için $k_2\,(512 \rightarrow 256)$ ve ε = %8 için $k_1\,(16 \rightarrow 8)$ farklılık göstermektedir.

Şekil 5.2'den de görülebileceği gibi tüm ε değerleri için optimum CASCADE protokolü [24]'e göre daha iyi sonuçlar üretmiştir. Yine Şekil 5.3'den görülebileceği gibi özellikle kanal hata olasılığının ε = % 4'ten büyük olduğu durumlarda optimum CASCADE protokolü [24]'ten ayrışarak teorik limite daha yakın sonuçlar vermektedir. ε = % 8'den itibaren aradaki fark daha da açılmaktadır.

Yazarlar [24]'teki çalışmalarında daha önce de bahsedildiği gibi elde ettikleri verimlilik değerlerinin doğruluğunu analiz etmişler ve daha gerçekçi olması açısından hatalı sonlanmış simülasyonları verimlilik hesaplarından çıkarmışlardır. Yine adil bir karşılaştırma olması açısından bu çalışmada da benzer değerler hesaplanmıştır ve sonuçlar Şekil 5.3'te sunulmuştur. Şekil 5.3'ten de görülebileceği tüm ε değerleri için optimum CASCADE protokolü [24]'e göre daha iyi sonuçlar üretmiştir. ε = % 2'den itibaren de [24]'ten verimlilik olarak olumlu yönde ayrışarak teorik limite [24]'e göre daha yakın sonuçlar elde etmiştir.

Tüm bu simülasyon sonuçlarından da görüldüğü üzere bu çalışmada önerilen optimum CASCADE protokolü, [24]'teki çalışmada önerilen ve şu an literatürde en verimli CASCADE olan protokole göre ve literatürdeki diğer CASCADE protokollerine göre daha verimlidir, teorik limitlere daha yakındır.

2015 yılında, [27]'deki çalışmada yazarlar KAD'da kullanılmak üzere CASCADE protokolü olmayan, ancak CASCADE gibi interaktif çalışan yeni bir BU protokolü önermişlerdir. Bu çalışmada, yazarlar algoritmanın birbirini takip eden iterasyonlarında bitleri sonsal hata olasılıklarına göre gruplamışlardır ve optimum blok uzunluklarının hesabında bu bit gruplarını kullanmışlardır. Ayrıca, yazarlar orijinal CASCADE protokolünün fazla parite değiş tokuşu yaptığı temel noktaları tespit etmeye çalışmışlar ve bunun önüne geçmek için buldukları yöntemleri anlatmaya

çalışmışlardır. Yazarlar yine bu çalışmada blok uzunluğu ve gizli anahtarın uzunluğunun neden ikinin üstel katı olması gerektiğini teorik olarak açıklamaya çalışmışlardır. [24]'teki çalışmada bu bilgi olduğu gibi kullanılmış ancak herhangi bir teorik ispat verilmemişti. Her iki çalışmada da referans verilmese dahi bu bilgi literatürde daha önce kullanılmış ve teorik olarak ispatı da sunulmuştur [84].

Hem [27]'deki çalışma ile hem de literatürde yaygın olarak kullanılan ve kabul görmüş CASCADE olmayan popüler diğer BU teknikleriyle bu çalışmada önerilen optimum CASCADE protokolünü karşılaştırmak için kapsamlı deneyler ve simülasyonlar çalıştırılmıştır. Elde edilen sonuçlar Tablo 5.4'te verilmiştir. Tablo 5.4'te optimum CASCADE protokolünün parametre kümesi olarak [24]'teki çalışmada önerilen parametre kümesi kullanılmıştır. Bu parametre kümesi aynı zamanda bu tez çalışmasının Bölüm 4.1.4'ünde de verilmiştir. Bu yöntemlerle ilgili detaylı bilgi için [26], [27], [32] ve [41] çalışmaları incelenebilir.

Tablo 5.4'teki [26], [27], [37] ve [41] için verilen verimlilik değerleri [85]'deki Tablo 2'den alınmıştır. [27]'de sunulan IR için verilen verimlilik değerleri ise [27]'deki Tablo 1'den alınmıştır. Tablo 5.4'ten de görüldüğü gibi optimum CASCADE protokolü [27]'de verilen BU tekniği dışında diğer tüm tekniklerden daha iyi verimlilik performansı göstermektedir.

Tablo 5.4. Optimum CASCADE protokolü ile literatürdeki popüler BU tekniklerinin verimlilik performanslarının karşılaştırılması

BU Protokolleri	**N**	**ε (%)**	**β**	**FER**
Polar [37]	2^{16}	2	0.94	0.09
Polar [37]	2^{24}	2	0.98	0.08
LDPC [37]	2^{17}	2	0.93	0.01
LDPC [37]	2^{17}	2	0.93	0.03
LDPC [41]	1944	2	0.87	< 0.01
LDPC [41]	1944	2	0.90	0.06
IR [27]	2^{10}	3	0.9747	0.00016
IR [27]	2^{14}	3	0.9940	0.00014
IR [27]	2^{16}	3	0.9955	0.0001
CASCADE [26]	$1Mbit$	3.8	≈ 0.97	≈ 0
Optimum CASCADE	2^{10}	3	0.9780	0.0023
Optimum CASCADE	2^{14}	3	0.9922	0.00014
Optimum CASCADE	2^{16}	3	0.9937	0

Bu tez çalışmasında yapılan kapsamlı simülasyonlara göre, kanal hata olasılığı % 10'u ve/veya gizli anahtarın uzunluğu 2^{10}'u geçtiğinde optimum CASCADE protokolü daha iyi verimlilik performansı göstermektedir. Ancak, bu bölümün girişinde de ifade edildiği gibi bir performans ölçütünde yapılan iyileştirme diğer performans ölçütü üzerinde olumsuz bir etkiye sebep olabilir. Şu ana kadar [23]'ten referans olarak alınan CASCADE protokolü üzerine yapılan değişikliklerin hız performansı üzerindeki etkisi incelenmemiştir. Bölüm 5.2'de protokole verimliliği iyileştirmek adına eklenen her iyileştirmenin hız üzerindeki etkisi incelenmiştir. Bu maksatla kapsamlı deney ve simülasyonlar çalıştırılmıştır ve hız performansları tablo ve şekillerle sunulmuştur.

5.2. Hız

KAD, pratik uygulamalarda kullanılmak üzere zaman geçtikçe daha uygun hale gelmektedir. Pratik olarak ilk deney 1989 yılında laboratuvar ortamında yapılmıştır. İlk gizli anahtarın iletimi uçlar arasındaki mesafenin 30 cm olduğu ve iletimin açık hava üzerinden sağlandığı bir ortamda yapılmıştır. Bu deneyde, veri iletim hızı sadece bir kaç bit/saniye olmuştur. Bugünlerde ise, ticari uygulamalar için uygun bir hale gelmiştir ve kilometreler mertebesinde veri iletimi yapılabilmektedir. Hız olarak ise Mbps hızlarına ulaşılmıştır [86]. Mesafe ve hızlarda iyileştirmeler sağlanmasına karşın hata sezme ve düzeltme maksadıyla kullanılan BU tekniklerinde de hem verimli hem de hızlı tekniklere ihtiyaç vardır.

KAD'da BU maksadıyla kullanılan CASCADE protokolü için literatürdeki çalışmalar iki performans ölçütü üzerinde yoğunlaşmıştır. Bunlar verimlilik ve hız olarak ifade edilmektedir. Bu çalışmada, protokolün verimlilik performansı üzerinde yoğunlaşılmıştır ve verimliliği arttırmak için protokolde gizli olan bazı içsel bilgiler tespit edilmiştir. Bu içsel bilgiler kullanılarak protokoldeki bazı fazlalık parite değiş tokuş işlemleri azaltılmış ve böylece orijinal protokolün verimlilik performansı oldukça iyileştirilmiştir. Birbiriyle ilintili birçok performans ölçütlerinde olduğu gibi, bir performans ölçütünde yapılan iyileştirme diğer performans ölçütü üzerinde olumsuz bir etkiye neden olabilir. Bu tez çalışmasından örnek verilecek olursa, verimlilik performansı odaklı yapılan iyileştirmeler protokolün hız performansını düşürmüş olabilir. Daha önce de belirtildiği gibi, bu tez çalışmasında [23]'teki çalışmada önerilen protokol gerçeklenmiş (referans protokol), üzerine Bölüm 4'teki

iyileştirmeler uygulanmış ve protokolün verimlilik performansı literatürdeki en iyi sonuçları verecek şekilde iyileştirilmiştir. Literatürde hız performans ölçütü ile ilgili yapılmış olan çalışmalar da mevcuttur. Bu konuda daha fazla bilgi edinmek isteyen araştırmacılar [26] nolu çalışmayı inceleyebilir.

Bu bölümde, referans protokol ve bunun üzerine eklediğimiz her bir iyileştirmenin ve son tahlilde optimum CASCADE protokolünün hızında olan değişimler kapsamlı simülasyonlarla ölçülmektedir. Bu bağlamda, Bölüm 4.3'te yapılan deneydekine benzer şekilde kapsamlı simülasyonlar burada da çalıştırılmıştır. Tablo 4.6 incelendiğinde öncelikle referans protokol, referans protokol üzerine Bölüm 4.2.1'de önerilen TBB iyileştirmesi uygulandığında, referans protokol üzerine Bölüm 4.2.2'de önerilen PBB iyileştirmesi uygulandığında ve son olarak optimum CASCADE protokolü uygulandığında elde edilen verimlilik performansları ele alınmıştı. Aynı yaklaşım burada da uygulanmaktadır, ancak bu bölümde Tablo 4.6'den farklı olarak protokol hızındaki değişimler ölçülmektedir. Bu bağlamda, yapılan simülasyonlarda gizli anahtar uzunluğu N=10000 olarak seçilmiş ve simülasyonlar 100 başarılı deneme için çalıştırılmıştır. [23]'teki parametre kümesi hiç değiştirilmeden burada da aynı şekilde kullanılmıştır. Her iyileştirmenin hız performansı üzerindeki etkileri Tablo 5.5 ve Şekil 5.4'te verilmektedir.

Tablo 5.5. Verimlilik iyileştirmelerinin protokolün hızı (bit/saniye) üzerindeki etkisi

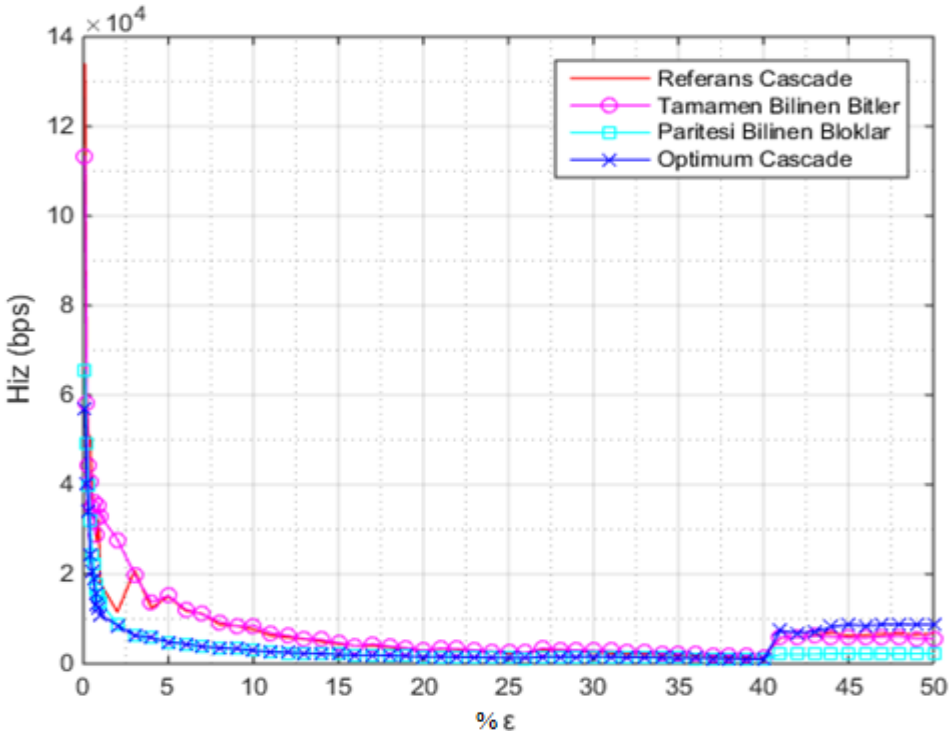
ε **(%)**	**Referans CASCADE**	**TBB**	**PBB**	**Optimum CASCADE**
0.1	133886,73	113263,11	65505,04	56989,79
0.2	61304,56	58339,65	49229,55	40303,07
0.3	51036,03	44320,34	39652,64	34097,10
0.4	50671,39	40430,17	32242,46	24316,10
0.5	38599,60	34801,97	24105,67	20473,34
0.6	35184,01	36280,52	22152,81	18873,26
0.7	30032,73	28763,73	17606,56	15651,65
0.8	27210,88	32504,46	16528,92	13179,57
0.9	32323,75	35457,22	15122,87	12350,40
1.0	18048,58	33057,85	13657,65	10983,45
2.0	11514,90	27753,87	8853,23	8571,03
3.0	20819,88	19993,60	6209,36	6311,01
4.0	12461,05	13745,70	5802,34	5798,17
5.0	15012,76	15158,86	4845,92	4818,18

Tablo 5.5. (Devam) Verimlilik iyileştirmelerinin protokolün hızı (bit/saniye) üzerindeki etkisi

ε (%)	Referans CASCADE	Tamamen Bilinen Bitler	Paritesi Bilinen Bloklar	Optimum CASCADE
6.0	11845,25	12188,28	4465,84	4425,67
7.0	11167,31	11021,22	3807,02	3803,64
8.0	8731,26	9384,12	3541,77	3530,64
9.0	8471,20	8393,48	3301,52	3309,19
10.0	7634,52	8544,74	2957,89	2926,78
11.0	6295,48	6856,68	2662,22	2638,52
12.0	5860,25	6269,00	2428,56	2514,14
13.0	5496,89	5571,03	2321,19	2353,63
14.0	4952,01	5366,87	2257,01	2222,14
15.0	4397,71	4668,46	2070,45	2077,92
16.0	3996,99	3985,55	1869,81	1876,33
17.0	4060,40	4402,86	1937,63	1934,12
18.0	3684,93	3895,30	1816,37	1822,01
19.0	3364,88	3562,28	1642,45	1660,48
20.0	2914,12	2994,43	1493,61	1496,72
21.0	3353,58	3450,50	1581,73	1560,70
22.0	3141,26	3304,93	1508,43	1482,20
23.0	2951,75	3070,28	1436,13	1406,96
24.0	2670,78	2855,10	1357,28	1352,34
25.0	2354,93	2654,50	1286,50	1286,06
26.0	2319,76	2489,01	1217,03	1245,81
27.0	3151,77	3423,36	1517,98	1537,31
28.0	2905,00	3210,10	1503,01	1533,22
29.0	2724,68	3052,27	1513,07	1526,68
30.0	2543,11	2931,33	1545,03	1567,32
31.0	2310,96	2869,95	1533,30	1489,58
32.0	2168,90	2688,85	1466,51	1476,04
33.0	2116,33	2552,95	1424,12	1415,71
34.0	2026,85	2337,21	1343,21	1336,70
35.0	1874,41	2211,47	1271,96	1269,33
36.0	1749,11	2134,54	1220,58	1224,69
37.0	1677,54	1948,78	1199,60	1197,89
38.0	1529,38	1863,00	1129,34	1131,10
39.0	1504,92	1770,35	1078,34	1070,53
40.0	1469,40	1588,52	990,804	1002,44
41.0	5823,46	5866,16	2084,96	7380,94
42.0	5811,31	5805,51	2066,85	6860,31
43.0	6334,12	6197,93	2101,66	7143,62
44.0	6924,15	6040,58	2220,60	8204,05
45.0	6241,49	5774,07	2236,74	8763,47
46.0	6407,05	5737,33	2213,38	8444,37
47.0	6594,52	6033,76	2237,37	8755,11

Tablo 5.5. (Devam) Verimlilik iyileştirmelerinin protokolün hızı (bit/saniye) üzerindeki etkisi

ε (%)	Referans CASCADE	Tamamen Bilinen Bitler	Paritesi Bilinen Bloklar	Optimum CASCADE
48.0	6756,75	6119,12	2209,02	8798,40
49.0	6492,83	5605,66	2255,26	8715,73
50.0	6743,22	5623,91	2223,91	8727,67



Şekil 5.4. Verimlilik iyileştirmelerinin protokolün hızı (bit/saniye) üzerindeki etkisi

Tablo 5.5 ve Şekil 5.4'ten de görüldüğü gibi, önerilen iyileştirmeler verimlilik performansını arttırken hız performansını azaltmıştır. TBB iyileştirmesi, [23]'te önerilen ve referans olarak alınan CASCADE protokolünü hız açısından çok olumsuz etkilememiş, aksine bazı gereksiz parite kontrollerini yapmadığı için hız performansı açısından az da olsa iyileştirmiştir (bkz. Tablo 5.9). Bölüm 4.3'te, PBB iyileştirmesinin blok uzunluğunu daha fazla azalttığı için verimlilik performansı olarak tamamen bilinen bitlerden biraz daha iyi sonuçlar ürettiği ifade edilmişti. Bunun sebebi ise, blok seviyesinde çıkarma yaptığı için arama yapılan blokların boyutunu TBB'ye göre daha fazla küçültmesinden kaynaklanmaktadır. Ancak, yoğun olarak yapılan blok arama işlemleri protokolün TBB iyileştirmesine göre daha yavaş çalışmasına neden olmaktadır. Bu yoğun blok arama işlemi, yöntemi en yavaş protokol versiyonu olarak

karşımıza çıkarmaktadır (bkz. Tablo 5.9). Optimum CASCADE protokolünün hız performansı ise, her iki iyileştirmeyi birlikte kullandığı için, referans protokol veya sadece TBB iyileştirmesini içeren protokol kadar hızlı değil, ancak yoğun arama işlemi sonucu yavaş çalışan PBB iyileştirmesini içeren protokol kadar da yavaş çalışmamaktadır. Daha doğrusu, ε < % 40 değerleri için paritesi bilinen bloklar ile çok benzer hız performansına sahiptir. Ancak, ε > % 40 için yoğun arama yapan PBB iyileştirmesini içeren protokolden daha iyi sonuçlar üretmektedir.

Şekil 5.4'ten görüldüğü gibi tüm protokollerin hızları kanal hata olasılığı ε arttıkça azalmaktadır. Ancak dikkat edilirse, ölçülen hız miktarı ε = % 40'tan itibaren tekrar yükselmeye başlamıştır. Bunun sebebi, Bölüm 4.1.4'te de verildiği gibi $k_1 = 0{,}80/\varepsilon$ olmasından kaynaklanmaktadır. ε arttıkça blok uzunluğu 1'e yaklaşmaktadır. Bu da bir bit uzunluğunda blokların oluşması anlamına gelmektedir. Diğer bir deyişle, bit seviyesinde parite değiş tokuşu, yani bitin kendisinin değiş tokuşu anlamına gelecektir. Bu sebeple, her bit bire bir şekilde değiş tokuş edilecektir. Bu durum çok verimsiz ve de çok güvensizdir. Zira, hattı dinleyen saldırgan gizli anahtarın tüm bitlerini ele geçirmiş olur. Bu şekilde, bir ya da bire yakın uzunluklu bloklar için BINARY protokolü çok az zaman alacaktır ve hatta bir bit uzunluğundaki bloklar için hiç çalıştırılmayacaktır. Bu nedenle, yüksek ε değerleri için protokol çok çabuk sonlanır, ortalama sonlanma süresi çok kısalır. Bu durum, Tablo 5.6'da, Tablo 5.7'de ve Şekil 5.5'te de gösterilmiştir.

Protokolde yapılan her bir iyileştirmenin protokolün hız performansına etkisi olmaktadır. Tamamen bilinen bitler ve optimum CASCADE iyileştirmeleri, bazı ε değerleri için hız performansına olumlu katkıda bulunurken, bazı ε değerleri için de hızda olumsuz etkide bulunmuştur. Bunun aksine, paritesi bilinen bloklar iyileştirmesi ise, protokolün hız performansı üzerinde bütün ε değerleri için olumsuz etkide bulunmuştur. Ayrı ayrı bütün ε değerleri için iyileştirmelerin protokolün hız performansı üzerindeki etkileri Tablo 5.6'da gösterilmiştir. Tablonun kolay analizi açısından her bir iyileştirmenin referans CASCADE protokolü üzerindeki etkisi ayrı ayrı ele alınmıştır. Tablodaki referans CASCADE kolonundaki değerler referans CASCADE protokolünün hız değerlerini içermektedir. İyileştirmelerin olduğu kolonlarda ise her bir iyileştirmenin referans CASCADE kolonundaki hız

performansına etkisi yüzde olarak verilmiştir. Örneğin, tamamen bilinen bitler kolonundaki değerler bu iyileştirmenin referans CASCADE protokolünün hız değerlerini hangi oranda arttırdığı veya azalttığını ifade etmektedir. Tablo 5.6'da, her bir iyileştirme için sunulan değişim oranlarındaki pozitif değerler hız performansının arttığını, diğer bir deyişle protokolün kısa sürede sonlandığını göstermektedir ve aslında değerlerdeki bu artış protokol iyileştirmesi için olumlu bir anlam taşımaktadır.

Tablo 5.6. İyileştirmelerin referans protokolünün hız performansı üzerindeki etkisi

ε (%)	Referans CASCADE	Tamamen Bilinen Bitler (%)	Paritesi Bilinen Bloklar (%)	Optimum CASCADE (%)
0.1	133886,73	-15.4038	-51.0743	-57.4343
0.2	61304,56	-4.8364	-19.6968	-34.2576
0.3	51036,03	-13.1587	-22.3046	-33.1901
0.4	50671,39	-20.2111	-36.3695	-52.0122
0.5	38599,60	-9.8385	-37.5494	-46.9597
0.6	35184,01	3.1165	-37.0373	-46.3584
0.7	30032,73	-4.2254	-41.3754	-47.8847
0.8	27210,88	19.4539	-39.2562	-51.5651
0.9	32323,75	9.6940	-53.2144	-61.7916
1.0	18048,58	83.1604	-24.3284	-39.1451
2.0	11514,90	141.0257	-23.1150	-25.5657
3.0	20819,88	-3.9687	-70.1758	-69.6876
4.0	12461,05	10.3093	-53.4362	-53.4697
5.0	15012,76	0.9732	-67.7213	-67.9061
6.0	11845,25	2.8959	-62.2985	-62.6376
7.0	11167,31	-1.3082	-65.9092	-65.9395
8.0	8731,26	7.4773	-59.4358	-59.5632
9.0	8471,20	-0.9175	-61.0265	-60.9360
10.0	7634,52	11.9224	-61.2564	-61.6639
11.0	6295,48	8.9143	-57.7122	-58.0887
12.0	5860,25	6.9750	-58.5588	-57.0984
13.0	5496,89	1.3488	-57.7727	-57.1825
14.0	4952,01	8.3776	-54.4223	-55.1265
15.0	4397,71	6.1566	-52.9198	-52.7500
16.0	3996,99	-0.2862	-53.2195	-53.0564
17.0	4060,40	8.4341	-52.2798	-52.3663
18.0	3684,93	5.7089	-50.7082	-50.5551
19.0	3364,88	5.8665	-51.1885	-50.6526
20.0	2914,12	2.7559	-48.7458	-48.6390
21.0	3353,58	2.8900	-52.8346	-53.4617
22.0	3141,26	5.2103	-51.9801	-52.8151
23.0	2951,75	4.0156	-51.3465	-52.3347
24.0	2670,78	6.9014	-49.1804	-49.3654

Tablo 5.6. (Devam) İyileştirmelerin referans protokolünün hız performansı üzerindeki etkisi

ε (%)	Referans CASCADE	Tamamen Bilinen Bitler (%)	Paritesi Bilinen Bloklar (%)	Optimum CASCADE (%)
25.0	2354,93	12.7210	-45.3699	-45.3886
26.0	2319,76	7.2960	-47.5364	-46.2957
27.0	3151,77	8.6171	-51.8372	-51.2239
28.0	2905,00	10.5026	-48.2613	-47.2213
29.0	2724,68	12.0231	-44.4680	-43.9685
30.0	2543,11	15.2656	-39.2464	-38.3699
31.0	2310,96	24.1886	-33.6510	-35.5428
32.0	2168,90	23.9730	-32.3846	-31.9452
33.0	2116,33	20.6310	-32.7080	-33.1054
34.0	2026,85	15.3124	-33.7292	-34.0504
35.0	1874,41	17.9822	-32.1408	-32.2811
36.0	1749,11	22.0358	-30.2171	-29.9821
37.0	1677,54	16.1689	-28.4905	-28.5925
38.0	1529,38	21.8141	-26.1570	-26.0419
39.0	1504,92	17.6375	-28.3457	-28.8647
40.0	1469,40	8.1067	-32.5708	-31.7790
41.0	5823,46	0.7332	-64.1972	26.7449
42.0	5811,31	-0.0998	-64.4340	18.0510
43.0	6334,12	-2.1501	-66.8200	12.7800
44.0	6924,15	-12.7607	-67.9296	18.4846
45.0	6241,49	-7.4889	-64.1634	40.4067
46.0	6407,05	-10.4529	-65.4540	31.7981
47.0	6594,52	-8.5034	-66.0723	32.7634
48.0	6756,75	-9.4369	-67.3065	30.2165
49.0	6492,83	-13.6638	-65.2654	34.2362
50.0	6743,22	-16.5990	-67.0201	29.4288

Tablo 5.7'de ise her bir iyileştirmenin protokolün sonlanma süresi üzerindeki etkisi incelenmektedir. Tabloda referans CASCADE ve her bir iyileştirmenin uygulandığı durumda protokolün sonlanma süresi sıralanmıştır.

Tablo 5.7. Verimlilik iyileştirmelerinin protokolün sonlanma süresi (milisaniye) üzerindeki etkisi

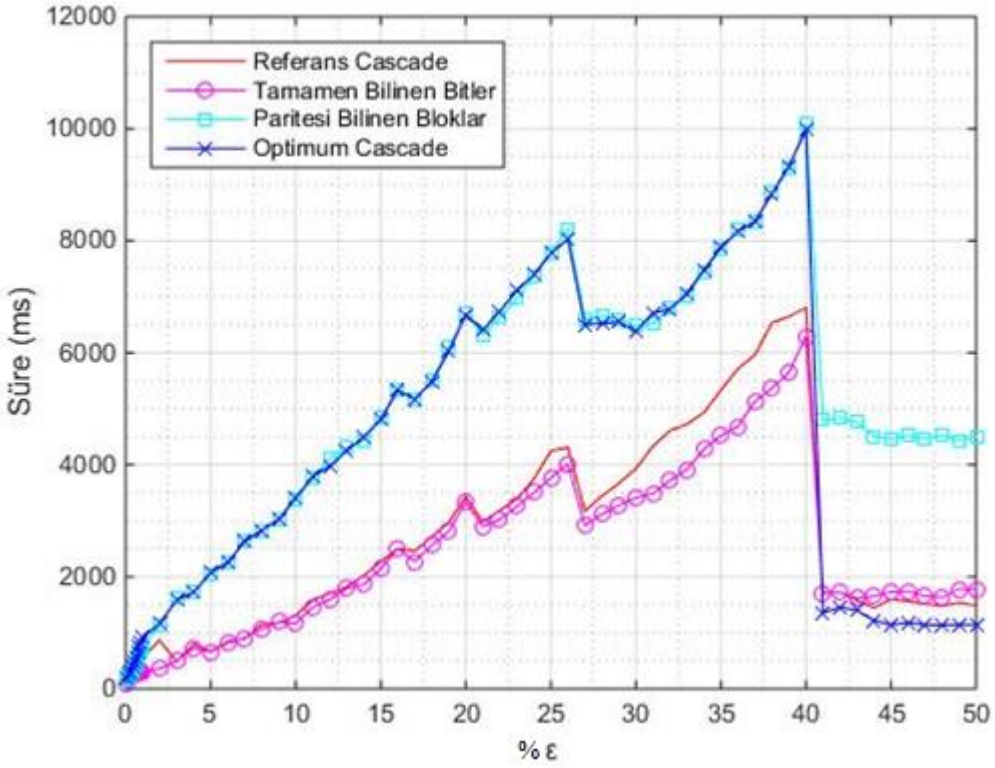
ε (%)	Referans CASCADE	Tamamen Bilinen Bitler	Paritesi Bilinen Bloklar	Optimum CASCADE
0.1	74,69	88,29	152,66	175,47
0.2	163,12	171,41	203,13	248,12
0.3	195,94	225,63	252,19	293,28
0.4	197,35	247,34	310,15	411,25

Tablo 5.7. (Devam) Verimlilik iyileştirmelerinin protokolün sonlanma süresi (milisaniye) üzerindeki etkisi

ε (%)	Referans CASCADE	Tamamen Bilinen Bitler	Paritesi Bilinen Bloklar	Optimum CASCADE
0.5	259,07	287,34	414,84	488,44
0.6	284,22	275,63	451,41	529,85
0.7	332,97	347,66	567,97	638,91
0.8	367,50	307,65	605,00	758,75
0.9	309,37	282,03	661,25	809,69
1.0	554,06	302,50	732,19	910,46
2.0	868,44	360,31	1129,53	1166,72
3.0	480,31	500,16	1610,47	1584,53
4.0	802,50	727,50	1723,44	1724,68
5.0	666,10	659,68	2063,59	2075,47
6.0	844,22	820,46	2239,22	2259,54
7.0	895,47	907,34	2626,72	2629,06
8.0	1145,31	1065,63	2823,44	2832,34
9.0	1180,47	1191,40	3028,90	3021,88
10.0	1309,84	1170,31	3380,78	3416,72
11.0	1588,44	1458,43	3756,25	3790,00
12.0	1706,41	1595,15	4117,66	3977,50
13.0	1819,21	1795,00	4308,13	4248,75
14.0	2019,38	1863,28	4430,63	4500,16
15.0	2273,91	2142,03	4829,85	4812,50
16.0	2501,88	2509,06	5348,13	5329,53
17.0	2462,81	2271,25	5160,93	5170,31
18.0	2713,75	2567,19	5505,47	5488,43
19.0	2971,87	2807,19	6088,44	6022,34
20.0	3431,56	3339,53	6695,15	6681,25
21.0	2981,88	2898,13	6322,19	6407,35
22.0	3183,43	3025,78	6629,38	6746,72
23.0	3387,81	3257,03	6963,13	7107,50
24.0	3744,22	3502,50	7367,65	7394,54
25.0	4246,41	3767,18	7772,97	7775,63
26.0	4310,78	4017,66	8216,71	8026,88
27.0	3172,82	2921,10	6587,66	6504,85
28.0	3442,34	3115,16	6653,28	6522,19
29.0	3670,15	3276,25	6609,06	6550,15
30.0	3932,19	3411,41	6472,34	6380,31
31.0	4327,19	3484,38	6521,87	6713,28
32.0	4610,62	3719,06	6818,91	6774,84
33.0	4725,15	3917,03	7021,88	7063,59
34.0	4933,75	4278,59	7444,85	7481,09
35.0	5335,00	4521,87	7861,88	7878,12
36.0	5717,18	4684,84	8192,81	8165,31
37.0	5961,09	5131,40	8336,10	8347,97

Tablo 5.7. (Devam) Verimlilik iyileştirmelerinin protokolün sonlanma süresi (milisaniye) üzerindeki etkisi

ε (%)	Referans CASCADE	Tamamen Bilinen Bitler	Paritesi Bilinen Bloklar	Optimum CASCADE
38.0	6538,59	5367,66	8854,69	8840,93
39.0	6644,84	5648,60	9273,44	9341,09
40.0	6805,47	6295,16	10092,81	9975,62
41.0	1717,19	1704,69	4796,25	1354,84
42.0	1720,78	1722,50	4838,28	1457,66
43.0	1578,75	1613,44	4758,13	1399,85
44.0	1444,22	1655,47	4503,28	1218,91
45.0	1602,18	1731,88	4470,78	1141,10
46.0	1560,78	1742,97	4517,97	1184,22
47.0	1516,41	1657,34	4469,53	1142,19
48.0	1480,00	1634,22	4526,88	1136,57
49.0	1540,16	1783,91	4434,07	1147,35
50.0	1482,97	1778,12	4496,57	1145,78



Şekil 5.5. Verimlilik iyileştirmelerinin protokolün tamamlanması süresi (milisaniye) üzerindeki etkisi

Tablo 5.7'de ve Şekil 5.5'ten de görüldüğü gibi, önerilen iyileştirmeler verimlilik performansını arttırırken simülasyonların sonlanma sürelerini de genel olarak arttırmaktadır. TBB iyileştirmesi, [23]'te önerilen ve referans olarak alınan

CASCADE protokolünü simülasyon süresi açısından çok olumsuz etkilememiş, aksine bazı gereksiz parite kontrollerini yapmadığı için onu yer yer geçmiştir. Bölüm 4.3'te PBB iyileştirmesi daha önce de bahsedildiği gibi yoğun olarak blok arama işlemleri yaptığı için protokolün yavaş çalışmasına neden olmaktadır. Optimum CASCADE protokolü ise, PBB iyileştirmesini içeren protokolden daha iyi sonuçlar üretmiştir. Optimum CASCADE protokolü, her iki iyileştirmeyi de içerdiği için TBB iyileştirmesi sayesinde PBB versiyonundaki kadar kötü simülasyon süreleri vermemiştir.

Protokolde yapılan her bir iyileştirmenin protokolün sonlanma süresine etkisi olmaktadır. Tamamen bilinen bitler ve optimum CASCADE iyileştirmeleri, bazı ε değerleri için protokolün sonlanma süresine olumlu katkıda bulunurken, bazı ε değerleri için de sürede olumsuz etkide bulunmuştur. Bunun aksine, paritesi bilinen bloklar iyileştirmesi ise, protokolün sonlanma süresi üzerinde bütün ε değerleri için olumsuz etkide bulunmuştur. Ayrı ayrı bütün ε değerleri için iyileştirmelerin protokolün sonlanma süresi üzerindeki etkileri Tablo 5.8'de gösterilmiştir. Tablonun kolay analizi açısından her bir iyileştirmenin referans CASCADE protokolü üzerindeki etkisi olarak ele alınmıştır. Tablodaki referans CASCADE kolonundaki değerler referans CASCADE protokolünün sonlanma süresi değerlerini içermektedir. İyileştirmelerin olduğu kolonlarda ise her bir iyileştirmenin referans CASCADE kolonundaki sonlanma süresine etkileri yüzde olarak verilmiştir. Örneğin, tamamen bilinen bitler kolonundaki değerler bu iyileştirmenin referans CASCADE protokolünün sonlanma süresi değerlerini hangi oranda arttırdığı ve azalttığını ifade etmektedir. Bu tabloda, her bir iyileştirme için sunulan sonlanma sürelerindeki pozitif değerler sonlanma süresinin arttığını, diğer bir deyişle protokolün hızının düştüğünü, göstermektedir ve aslında değerlerdeki bu artış protokol iyileştirmesi için olumsuz bir anlam taşımaktadır.

Tablo 5.8. İyileştirmelerin referans protokolünün sonlanma süreleri üzerindeki etkisi

ε (%)	Referans CASCADE	Tamamen Bilinen Bitler (%)	Paritesi Bilinen Bloklar (%)	Optimum CASCADE (%)
0.1	74,69	18.2086	104.3915	134.9310
0.2	163,12	5.0821	24.5280	52.1089
0.3	195,94	15.1526	28.7078	49.6785

Tablo 5.8. (Devam) İyileştirmelerin referans protokolünün sonlanma süreleri üzerindeki etkisi

ε (%)	Referans CASCADE	Tamamen Bilinen Bitler (%)	Paritesi Bilinen Bloklar (%)	Optimum CASCADE (%)
0.4	197,35	25.3306	57.1573	108.3861
0.5	259,07	10.9121	60.1266	88.5359
0.6	284,22	-3.0223	58.8242	86.4225
0.7	332,97	4.4118	70.5769	91.8822
0.8	367,50	-16.2857	64.6259	106.4626
0.9	309,37	-8.8373	113.7408	161.7222
1.0	554,06	-45.4030	32.1499	64.3252
2.0	868,44	-58.5107	30.0643	34.3466
3.0	480,31	4.1327	235.2980	229.8974
4.0	802,50	-9.3458	114.7589	114.9134
5.0	666,10	-0.9638	209.8018	211.5853
6.0	844,22	-2.8144	165.2413	167.6482
7.0	895,47	1.3256	193.3342	193.5955
8.0	1145,31	-6.9571	146.5219	147.2990
9.0	1180,47	0.9259	156.5842	155.9896
10.0	1309,84	-10.6524	158.1063	160.8502
11.0	1588,44	-8.1848	136.4742	138.5989
12.0	1706,41	-6.5201	141.3054	133.0917
13.0	1819,21	-1.3308	136.8132	133.5492
14.0	2019,38	-7.7301	119.4055	122.8486
15.0	2273,91	-5.7997	112.4029	111.6399
16.0	2501,88	0.2870	113.7644	113.0210
17.0	2462,81	-7.7781	109.5545	109.9354
18.0	2713,75	-5.4006	102.8731	102.2452
19.0	2971,87	-5.5413	104.8690	102.6448
20.0	3431,56	-2.6819	95.1051	94.7001
21.0	2981,88	-2.8086	112.0203	114.8762
22.0	3183,43	-4.9522	108.2465	111.9324
23.0	3387,81	-3.8603	105.5348	109.7963
24.0	3744,22	-6.4558	96.7740	97.4921
25.0	4246,41	-11.2855	83.0480	83.1107
26.0	4310,78	-6.7997	90.6084	86.2048
27.0	3172,82	-7.9336	107.6279	105.0179
28.0	3442,34	-9.5046	93.2778	89.4697
29.0	3670,15	-10.7325	80.0760	78.4709
30.0	3932,19	-13.2440	64.5989	62.2584
31.0	4327,19	-19.4771	50.7184	55.1418
32.0	4610,62	-19.3371	47.8957	46.9399
33.0	4725,15	-17.1025	48.6065	49.4892
34.0	4933,75	-13.2791	50.8964	51.6309
35.0	5335,00	-15.2414	47.3642	47.6686
36.0	5717,18	-18.0568	43.3016	42.8206

Tablo 5.8. (Devam) İyileştirmelerin referans protokolünün sonlanma süreleri üzerindeki etkisi

ε (%)	Referans CASCADE	Tamamen Bilinen Bitler (%)	Paritesi Bilinen Bloklar (%)	Optimum CASCADE (%)
37.0	5961,09	-13.9184	39.8419	40.0410
38.0	6538,59	-17.9080	35.4220	35.2116
39.0	6644,84	-14.9927	39.5585	40.5766
40.0	6805,47	-7.4985	48.3044	46.5824
41.0	1717,19	-0.7279	179.3081	-21.1013
42.0	1720,78	0.1000	181.1678	-15.2907
43.0	1578,75	2.1973	201.3859	-11.3317
44.0	1444,22	14.6273	211.8140	-15.6008
45.0	1602,18	8.0952	179.0436	-28.7783
46.0	1560,78	11.6730	189.4687	-24.1264
47.0	1516,41	9.2937	194.7442	-24.6780
48.0	1480,00	10.4203	205.8703	-23.2047
49.0	1540,16	15.8263	187.8967	-25.5045
50.0	1482,97	19.9026	203.2138	-22.7375

Bu tez çalışmasında, simülasyonlar aynı bilgisayar üzerinde çalıştırılmış olup elde edilen sonuçlarda ağ haberleşmelerinden kaynaklanacak süreler hesaba katılmamıştır. Diğer bir deyişle, bu çalışmada sunulan hız performansları ve simülasyon sonlanma süreleri protokolün işlemsel karmaşıklığından kaynaklanan süreleri ölçmektedir.

Şu ana kadar yapılan analizlerde her iyileştirmenin tüm ε değerleri için hız performansı ve protokol sonlanma süresi hesaplanmıştır. Protokolün detaylı analizi için bu analizler yeterli olmaktadır. Ancak, protokol iyileştirmelerinin referans protokole katkısını hızlı ve kaba bir hesapla görülebilmesi için bu değerlerin ortalamasına da bakılabilir. Bu açıdan, referans CASCADE ve önerilen her bir protokol iyileştirmesi için Tablo 5.5 ve Tablo 5.7'deki değerlerin ortalaması hesaplanmıştır ve her bir iyileştirmenin referans CASCADE protokolüne ortalama olarak nasıl bir etkide bulunduğu incelenmiştir. Sonuçlar, Tablo 5.9 ve Tablo 5.10'da sunulmuştur. Bu sayede, çeşitli amaçlar için protokol seçimi de mümkün olabilmektedir. Öncelikle, protokolün davranışı bu değerler üzerinden genel hatlarıyla ele alınır ve ardından Tablo 5.5 ve Tablo 5.7'deki değerler de daha detaylı incelenerek nihai protokol seçimine karar verilebilecektir.

Tablo 5.9 ve Tablo 5.10'dan da görüldüğü gibi TBB iyileştirmesi protokolün ortalama hızında % 0,0719 oranında iyileştirme yapmış ve simülasyonların ortalama % 8,59 kadar daha hızlı sonlanmasını sağlamıştır. PBB iyileştirmesi ise, bellekte yoğun olarak blok araması yaptığı için protokolün hızını % 44,8446 oranında azaltmış ve simülasyonların % 91,23 kadar daha geç sonlanmasına neden olmuştur. Optimum CASCADE protokolü de protokolün hızını % 43,3409 oranında düşürmüştür, bu da simülasyonların % 68,14 oranında daha geç sonlanmasına neden olmuştur.

Tablo 5.9. Verimlilik için yapılan iyileştirmelerin ortalama hız performansına (bit/saniye) etkileri

Protokol	Hız (bps)	Kazanç (%)
Referans CASCADE	12570	-
Tamamen Bilinen Bitler	12579	+ %0,0719
Paritesi Bilinen Bloklar	6933,3	**-** %44,8446
Optimum CASCADE	7122,3	**-** %43,3409

Tablo 5.10. Verimlilik için yapılan iyileştirmelerin ortalama sonlanma sürelerine (milisaniye) olan etkileri

Protokol	Süre (milisaniye)	Kazanç (%)
Referans CASCADE	2402,3	-
Tamamen Bilinen Bitler	2195,8	+ %8.59
Paritesi Bilinen Bloklar	4593,9	**-** %91,23
Optimum CASCADE	4039,4	**-** %68,14

Tablo 5.11. Verimlilik için yapılan iyileştirmelerin ortalama verimlilik değerlerine etkileri

Protokol	Ortalama Verimlilik	Kazanç (%)
Referans CASCADE	0.2790	-
Tamamen Bilinen Bitler	0.3539	+ %26,83
Paritesi Bilinen Bloklar	0.3546	+ %27,10
Optimum CASCADE	0.3546	+ %27,11

Tablo 5.11'den ve Bölüm 4'te yapılan kapsamlı verimlilik analizlerinden de görülebileceği gibi en yüksek verimlilik kazancı optimum CASCADE protokolü ile elde edilmektedir. Verimliliğin çok kritik olduğu uygulamalarda bu protokol tercih edilebilir. Hızın da çok önemli olduğu uygulamalarda ise, şekil ve tablolardan da görülebileceği gibi en fazla hız kaybı bellekte çok sayıda arama yükü getirdiği için PBB iyileştirmesinde olmaktadır. Bu nedenle, hızı arttırmak için sadece TBB iyileştirmesinin kullanıldığı bir konfigürasyon ayarlanabilir. Bu sayede, hızdan kaybetmeyecek, aksine ortalama % 0,0719 da olsa kazanacak, ve verimlilikten de ortalama % 26,83 oranında kazanç sağlanabilecektir. PBB iyileştirmesi ile birlikte kullanılmış olsa dahi verimlilik kazancı sadece % 27,11'e çıkacaktı, ancak bunun için protokol ortalama % 68,14 oranında daha yavaş çalışacaktı. Bu bağlamda, bu protokolü pratik uygulamalarda kullanacak araştırmacılar performans ölçütlerine göre uygun iyileştirmeleri ayarlayabilmektedir.

6. SONUÇLAR VE ÖNERİLER

KAD'da BU maksadıyla kullanılan protokoller için ana performans ölçütlerinden biri verimliliktir. Bu tez çalışmasında, orijinal CASCADE protokolünün verimlilik performansının arttırılması için yeni fikirler sunulmuştur. Ancak verimliliği arttırırken protokolün diğer performans ölçütü olan hızdan fedakarlıklar yapılmıştır. Önerilen her yeni fikrin protokolün hız performansı üzerinde etkileri de detaylı olarak incelenmiştir. [26]'deki çalışmada protokolün işlemsel performansı ve CASCADE protokolünün hız performansı kapsamlı olarak incelenmiştir.

KAD'daki BU problemi için literatürde şu an itibariyle üç ana çalışma kolu mevcuttur. Bunlar CASCADE [9, 24, 34, 85], LDPC **[**31-33, 37, 87**]** ve Polar kod [37, 38, 41, 88, 89] temelli çalışmalardır. Literatürde bahsi geçen tekniklerin sunduğu verimlilik performanslarına göre, bu çalışmada önerilen optimum CASCADE protokolü bütün LDPC ve Polar kod tabanlı BU tekniklerinin tamamından daha başarılı sonuçlar üretmektedir. CASCADE, BU ve diğer KAD konuları için ilgili araştırmacılar daha detaylı bilgileri [9, 28, 86, 89, 90] nolu çalışmalardan elde edebilirler.

Bu çalışmada sunulan iyileştirmelerle, verimlilik performansı teorik limite biraz daha yaklaşmıştır, ancak tam olarak ulaşamamıştır. Bu da protokolde halen yapılabilecek başka iyileştirmelerin olduğunu göstermektedir. Örneğin, geriye iz sürme adımlarında iyileştirmeler içinde hata sezilen en küçük bloğa uygulanmaktadır. Ancak bunun yerine, iyileştirmelerin tüm hatalı bloklara uygulandığı ve oluşan bloklar içinden en küçük bloğun seçilip BINARY tekniğinin bu bloğa uygulanması durumunda elde edilecek sonuçların nasıl olacağı denenmesi gereken bir husustur. Ayrıca, PBB iyileştirmesi uygulanırken, blok içinde paritesi bilinen bloğun tamamının içerilip içerilmediğine bakılmıştır. Ancak, paritesi bilinen bloğun bir kısmının içerilmiş olması durumu ele alınmamıştır. Paritesi bilinen bloğun bir kısmı içeriliyorsa bu kısım da çıkarılarak hata aranan bloğun uzunluğu azaltılabilir. Böylece BINARY işlemi daha küçük bir blok üzerinde çalışacaktır. Bu husus da denenecek konular arasında değerlendirilebilir.

Daha önce de değerlendirildiği gibi bu iyileştirmeler bellekte arama işlemleri getirdiği için protokolün hızını azaltmakta ve sistemdeki işlemci yükünü arttırmaktadır. Bu şekilde yeni aramaların eklenmesi sistemi yaklaşık olarak on kat daha yavaşlatmaktadır. Dolayısıyla, burada bahsedilen iyileştirmelerin daha hızlı gerçeklemeleri üzerinde çalışmalar yapılabilir. Diğer bir konu da, bu tez çalışmasında önerilen iyileştirmelerin BBBSS, Winnow, Liu'nun protokolü ve [27] gibi diğer interaktif BU teknikleri üzerinde uygulanması olabilir.

KAYNAKLAR

[1] Diffie W., Hellman M. E., Multiuser Cryptographic Techniques, *AFIPS National Computer Conference*, New York, USA, 7-10 June 1976.

[2] Diffie W., Hellman M. E., New Directions In Cryptography, *IEEE Transactions On Information Theory*, 1976, **22**, 644-654.

[3] Schneier B., *Applied Cryptography: Protocols, Algorithms, And Source Code In C*, 2nd Edition, John Wiley & Sons. Inc, New Jersey, 1996.

[4] Harrison K., Munro B., Spiller T., Security Through Uncertainty, *Elsevier Network Security*, 2007, **2**, 4–7.

[5] Kollmitzer C., Pivk M. (Eds.), *Applied Quantum Cryptography, Lecture Notes In Physics*, Springer, Berlin Heidelberg, 2010.

[6] Lomonaco S. J., A Quick Glance At Quantum Cryptography, *Cryptologia*, 1999, **1**, 1-41.

[7] Scarani V., Bechmann-Pasquinucci H., Cerf N. J., Dusek M., Lutkenhaus N., Peev M., The Security Of Practical Quantum Key Distribution, *Reviews Of Modern Physics*, 2009, DOI: 10.1103/RevModPhys.81.1301.

[8] Bae J., Acín A., Key Distillation From Quantum Channels Using Two-Way Communication Protocols, *Physical Review A*, 2007, **75** (1), 012334.

[9] Toyran M., Toyran M., Öztürk S., Optimized Cascade Protocol For Efficient Information Reconciliation In Quantum Key Distribution Systems, *Quantum Information & Computation (QIC)*, 2018, **7&8**, 0541-0552.

[10] Wiesner S., Conjugate Coding, *Sigact News*, 1983, **15** (1), 78–88.

[11] Bennett C.H., Brassard G., Quantum Cryptography: Public Key Distribution And Coin Tossing, *IEEE International Conference On Computers, Systems And Signal Processing*, Bangalore India, New York USA, 9-12 December 1984.

[12] Robert J. M., Détection Et Correction D'erreurs En Cryptographie, Msc Thesis, Université De Montréal, Department D'informatique Et De Recherche Operationnelle, Montreal, Canada, 1985.

[13] Bennett C. H., Brassard G., Robert J. M., Privacy Amplification By Public Discussion, *Siam J Comput*, 1988, **17**, 210–229.

[14] Bennett C. H., Bessette F., Brassard G., Salvail L., Smolin J., Experimental Quantum Cryptography, *J Cryptol*, 1992, **5**, 3–28.

[15] Richardson T., Urbanke R., *Modern Coding Theory*, 1st ed., Cambridge University Press, New York USA, 2008.

[16] Brassard G., Salvail L., Secret Key Reconciliation By Public Discussion, *EUROCRYPT Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, Lofthus Norway, 23-27 May 1993.

[17] Furukawa E., Yamazaki K., Application Of Existing Perfect Code To Secret Key Reconciliation, *ISCIT International Symposium On Communications And Information Technologies*, Chiangmai Thailand, 14-16 November 2001.

[18] Yamamura A., Ishizuka H., Error Detection And Authentication In Quantum Key Distribution, *Lect Notes Comput Sc*, 2001, **2119**, 260–273.

[19] Liu S., Information-Theoretic Secret Key Agreement, Phd Thesis, Eindhoven University Of Technology, Department Of Mathematics And Computer Science, Eindhoven, Netherlands, 2002.

[20] Buttler W. T., Lamoreaux S. K., Torgerson J. R., Nickel G. H., Donahue C. H., Peterson C. G., Fast, Efficient Error Reconciliation For Quantum Cryptography, *Phys Rev A*, 2003, **67**, 52303–52310.

[21] Liu S., Tilborg H. C. A. V., Dijk M. V., A Practical Protocol For Advantage Distillation And Information Reconciliation, *Design Code Cry*, 2003, **30**, 39–62.

[22] Sugimoto T., Yamazaki K., A Study On Secret Key Reconciliation Protocol "Cascade", *IEICE T Fund Electr*, 2000, **E83a**, 1987–1991.

[23] Yan H., Ren T., Peng X., Liu T., Guo H., Information Reconciliation Protocol In Quantum Key Distribution System, *IEEE Fourth International Conference On Natural Computation*, 2008, **3**, 637–641.

[24] Mateo J.M., Pacher C., Peev M., Ciurana A., Martin V., Demystifying The Information Reconciliation Protocol Cascade, *QIC*, 2015, **5&6**, 0453-0477.

[25] Toyran M., Pedersen T. B., More Efficient Implementations Of Cascade, *2nd Annual Conference On Quantum Cryptography (QCRYPT 2012)*, Singapore, 10-14 September 2012.

[26] Pedersen T. B., Toyran M., High Performance Information Reconciliation For QKD With Cascade, *QIC*, 2013, **5-6**, 419-434.

[27] Pacher C., Grabenweger P., Mateo J. M., Martin V., An Information Reconciliation Protocol For Secret-Key Agreement With Small Leakage, *IEEE International Symposium On Information Theory*, Hong Kong, 26 Apr – 1 May 2015.

[28] Elliott C., Pearson D., Troxel G., Quantum Cryptography In Practice, *In Proceedings Of The 2003 Conference On Applications, Technologies, Architectures, And Protocols For Computer Communications,* S*er. SIGCOMM'03*, Karlsruhe Germany, 25-29 August 2003.

[29] Pearson D., High-Speed QKD Reconciliation Using Forward Error Correction, *In 7th International Conference On Quantum Communication, Measurement And Computing*, November 2004, **1**, 299–302.

[30] Elliott C., Colvin A., Pearson D., Pikalo O., Schlafer J., Yeh H., Current Status Of The DARPA Quantum Network, 2005, *Proceedings Volume 5815, Quantum Information and Computation III*, DOI: 10.1117/12.606489.

[31] Elkouss D., Leverrier A., Alléaume R., Boutros J. J., Efficient Reconciliation Protocol For Discrete-Variable Quantum Key Distribution, *IEEE International Symposium On Information Theory*, Seoul Korea, 28 June – 3 July 2009.

[32] Sasaki M., Fujiwara M., Ishizuka H., Klaus W., Wakui K., et al., Field Test Of Quantum Key Distribution In The Tokyo QKD Network, *Opt. Express*, 2011, **19**(11), 10387–10409

[33] Mink A., Nakassis A., Ldpc For QKD Reconciliation, *The Computing Science And Technology International Journal*, 2012, **2** (2), 6-14.

[34] Arıkan E., Channel Polarization: A Method For Constructing Capacity-Achieving Code, *IEEE International Symposium On Information Theory (ISIT)*, Toronto Canada, 6-11 July 2008.

[35] Hassani S. H., Urbanke R., Polar Codes: Robustness Of The Successive Cancellation Decoder With Respect To Quantization, *2012 IEEE International Symposium On Information Theory Proceedings*, Massachusetts USA, 1-6 July 2012.

[36] Hassani S. H., Alishahi K., Urbanke R. L., Finite-Length Scaling For Polar Codes, *IEEE Transactions On Information Theory*, 2014, **60**(10), 5875-5898.

[37] Jouguet P., Kunz-Jacques S., High Performance Error Correction For Quantum Key Distribution Using Polar Codes, *Quantum Inf Comput*, 2014, **14**, 329–338.

[38] Nakassis A., Mink A., Polar Codes In A QKD Environment, *Proceedings Of SPIE: Defense Security & Sensing, Baltimore, Md*, 2014, **9123**, 1-11.

[39] Makkaveev A. P., Molotkov S. N., Pomozov D. I., Timofeev A. V., Practical Error-Correction Procedures In Quantum Cryptography, *Journal of Experimental and Theoretical Physics*, 2005, **101**, 230–252.

[40] Assche G.V., Information-Theoretic Aspects Of Quantum Key Distribution, Phd Thesis, Université Libre De Bruxelles, Brussels, Belgium, 2005.

[41] Martinez-Mateo J., Elkouss D., Martin V., Key Reconciliation For High Performance Quantum Key Distribution, 2013, *Scientic Reports*, DOI: 10.1038/srep01576.

[42] Elkouss D., Martinez-Mateo J., Martin V., Analysis Of A Rate-Adaptive Reconciliation Protocol And The Effect Of Leakage On The Secret Key Rate, *Physical Review A*, 2013, **87** (4), 23-34.

[43] Leverrier A., Alléaume R., Boutros J., Zémor G., Grangier P., Multidimensional Reconciliation For A Continuous-Variable Quantum Key Distribution, *Physical Review A*, 2008, **77** (4), 23-25.

[44] Walenta N., 1 Mbps Coherent One-Way QKD With Dense Wavelength Division Multiplexing And Hardware Key Distillation, *Presentation At 2nd Annual Conference On Quantum Cryptography (QCRYPT 2012)*, Singapore, 10-14 September 2012.

[45] Mink A., Custom Hardware To Eliminate Bottlenecks In QKD Throughput Performance, *Proc. Spıe 6780, Quantum Communications Realized*, Boston USA, 10 September 2007.

[46] C. E. Shannon, A Mathematical Theory Of Communication, *The Bell System Technical Journal*, July And October 1948, **27**, 379–423 and 623–656.

[47] -, Communication Theory Of Secrecy Systems, *The Bell System Technical Journal*, 1949, **28**, 656–715.

[48] Cover T. M., Thomas J. A., *Elements Of Information Theory 2nd Edition*, 2nd Edition, Wiley-Interscience, New Jersey USA, 2006.

[49] Dereli T., Verçin A., *Kuantum Mekaniği Temel Kavramlar Ve Uygulamaları*, 2. Basım, Genişletilmiş İkinci Basım, TÜBA, Ankara Türkiye, 2009.

[50] Trappe W., Washington L. C., *Introduction To Cryptography With Coding Theory*, 1st Edition, Prentice-Hall Inc, New Jersey USA, 2002.

[51] Zettili N., *Quantum Mechanics: Concepts And Applications*, 2nd Edition, Wiley, New Jersey USA, 2009.

[52] Heisenberg W., Über Den Anschaulichen Inhalt Der Quantentheoretischen Kinematik Und Mechanik, *Zeitschrift Für Physik*, 1927, **43** (3-4), 172-198.

[53] Sümer A., *Modern Teknik Fizik*, İstanbul Teknik Üniversitesi Matbaası, Gümüşsuyu İstanbul Türkiye, 1987.

[54] Wootters W., Zurek W., A Single Quantum Cannot Be Cloned, *Nature*, 1982, **299**, 802-803.

[55] Şahin A. B., Selçuk G., İletişim Ağ Güvenliğinde Son Aşama: Kuantum Kriptografi Ve Fiber Optik Ortamda Kuantum Temelli Rastsal Sayı Üretimi, *1. Ulusal Elektronik İmza Sempozyumu*, Ankara Türkiye, 7-8 Aralık 2006.

[56] Bennett C. H., Brassard G., The Dawn Of A New Era For Quantum Cryptography: The Experimental Prototype Is Working!, *Acm Sıgact News*, 1989, **20** (4), 78–80.

[57] Wang S., Chen W., Guo J. F., Yin Z. Q., Li H. W., Zhou Z., Guo G. C., Han Z. F., 2 Ghz Clock Quantum Key Distribution Over 260 km of Standard Telecom Fiber, *Optics Letters*, 2012, **37** (6), 1008-1010.

[58] Schmitt-Manderbach T., Weier H., Fürst M., Ursin R., Tiefenbacher F., et al., Experimental Demonstration Of Free-Space Decoy-State Quantum Key Distribution Over 144 Km, *Physical Review Letters*, 2007, **98** (1), 010504.

[59] Dixon A. R., Sato H., High Speed And Adaptable Error Correction For Megabit/S Rate Quantum Key Distribution, *Scientific Reports*, 2014, **4**, 72-75.

[60] Cobourne S., Quantum Key Distribution Protocols And Applications, Technical Report No: Rhul-Ma-2011-05, Department Of Mathematics, University Of London, England, 2011.

[61] Toyran M., Kuantum Kriptografi, Benzetimi Ve Analizleri, *15. İstatistik Araştırma Sempozyumu*, Ankara, Türkiye, 11-12 Mayıs 2006.

[62] Toyran M., Optik Ağlarda Kuantum Kriptografi Kullanarak Güvenli İletişim, *Elektrik, Elektronik, Bilgisayar Mühendisliği Sempozyumu (Eleco 2006)*, Bursa Türkiye, 6-10 Aralık 2006.

[63] Ekert A. K., Rarity J. G., Tapster P. R., Palma G. M., Practical Quantum Cryptography Based On Two-Photon Interferometry, *Physical Review Letters*, 1992, **69** (9), 1293-1295.

[64] Ekert A. K., Quantum Cryptography Based On Bell's Theorem, *Physical Review Letters*, 1991, **67** (6), 661-663.

[65] Bennett C. H., Quantum Cryptography Using Any Two Nonorthogonal States, *Physical Review Letters*, 1992, **68** (21), 3121.

[66] Bruss D., Optimal Eavesdropping In Quantum Cryptography With Six States, *Physical Review Letters*, 1998, **81** (14), 3018-3021.

[67] Bechmann-Pasquinucci H., Gisin N., Incoherent And Coherent Eavesdropping In The 6-State Protocol Of Quantum Cryptography, *Physical Review A*, 1999, **59** (6), 4238-4248.

[68] Scarani V., Acín A., Ribordy G., Gisin N., Quantum Cryptography Protocols Robust Against Photon Number Splitting Attacks For Weak Laser Pulse Implementations, *Physical Review Letters*, 2004, **92** (5), 057901.

[69] Ralph T. C., Continuous Variable Quantum Cryptography, *Physical Review A*, 1999, **61** (1), 010303.

[70] Hillery M., Quantum Cryptography With Squeezed States, *Physical Review A*, 2000, **61**, 022309.

[71] Cerf N. J., Lévy M., Assche G. V., Quantum Distribution Of Gaussian Keys Using Squeezed States, *Phys. Rev. A*, 2001, **63**, 052311.

[72] Gottesman D., Preskill J., Secure Quantum Key Distribution Using Squeezed States, *Physical Review A*, 2001, **63**, 022309.

[73] Grosshans F., Grangier P., Continuous Variable Quantum Cryptography Using Coherent States, *Physical Review Letters*, 2002, **88** (5), 057902.

[74] Silberhorn C., Ralph T. C., Lütkenhaus N., Leuchs G., Continuous Variable Quantum Cryptography: Beating The 3 Db Loss Limit, *Physical Review Letters*, 2002, **89** (16), 167901.

[75] Inoue K., Waks E., Yamamoto Y., Differential Phase Shift Quantum Key Distribution, *Physical Review Letters*, 2002, **89** (3), 037902.

[76] Stucki D., Brunner N., Gisin N., Scarani V., Zbinden H., Fast And Simple One-Way Quantum Key Distribution, *Applied Physics Letters*, 2005, **87** (19), 194108.

[77] Nıst, Announcing The Advanced Encryptıon Standard (Aes), *Technical Report No: Fıps 197, National Institute Of Standards And Technology (Nıst)*, 2001, USA.

[78] Uyar A., Kılınç H. H., Erdem S. S., Toyran M., Use Of Rijndael Block Cipher On J2ME Devices For Encryption And Hashing, *10th Nordic Workshop On Secure It-Systems (Norsec 2005)*, Tartu, Estonia, 20-21 October 2005.

[79] Gisin N., Ribordy G., Tittel W., Zbinden H., Quantum Cryptography, *Reviews Of Modern Physics*, 2002, **74** (1), 145-195.

[80] Williams C. P., Clearwater S. H., *Explorations In Quantum Computing*, 2nd Ed., Springer, New York USA, 2011.

[81] Nielsen M. A., Chuang I. L., *Quantum Computation And Quantum Information*, Cambridge University Press, Cambridge England, 2000.

[82] Hamitoğulları C., Sınır E. Y., *Kod Kitabı, Eski Mısır'dan Kuantum Kriptolojisine Gizlilik Bilimi*, İstanbul Klan Yayınları, İstanbul Türkiye, 2004.

[83] Ii-Yung R., A Probabilistic Analysis Of Bınary And Cascade, http://math.uchicago.edu/~may/reu2013/reupapers/ng.pdf; (Ziyaret Tarihi: 14 Nisan 2017).

[84] Oesterling L., Hayford D., Friend G., Comparison Of Commercial And Next Generation Quantum Key Distribution: Technologies For Secure Communication Of Information, *IEEE Conference On Technologies For Homeland Security*, Boston, Massachusetts, USA, 13 - 15 November 2012.

[85] Calver T.I., An Empirical Analysis Of The Cascade Secret Key Reconciliation Protocol For Quantum Key Distribution, Msc Thesis, Air Force Institute Of Technology, Ohio USA, 2011.

[86] Jiang X.Q., Huang P., Huang D., Lin D., Zeng G., Secret Information Reconciliation Based On Punctured Low-Density Parity-Check Codes For Continuous-Variable Quantum Key Distribution, *Physical Review A*, 2017, **95**, 022318.

[87] Qian C.C., Zhao S.M., Mao Q.P., Reconciliation Of Continuous Variable QKD Using Gaussian Post-Selection And Systematic Polar Code, *8th International Conference On Wireless Communications And Signal Processing (WCSP)*, Jiangsu China, 13-15 October 2016.

[88] Kim Y., Suh C., Rhee J. K. K., Reconciliation With Polar Codes By Gaussian Approximation For Continuous-Variable Quantum Key Distribution, *7th International Conference On Quantum Cryptography*, Cambridge UK, 18-22 September 2017.

[89] Martinez-Mateo J., Efficient Information Reconciliation For Quantum Key Distribution, Phd Thesis, Universidad Politécnica De Madrid, Madrid Spain, 2011.

[90] Gisin N., Ribordy G., Tittel W., Zbinden H., Quantum Cryptography, *Rev Mod Phys*, 2002, **74**, 145–195.

[91] Toyran M., Kuantum Anahtar Dağıtımında Bilgi Uzlaştırma, Doktora Tezi, Gebze Teknik Üniversitesi, Gebze Kocaeli, Türkiye, 2016.

[92] Chen K., Improvement Of Reconciliation For Quantum Key Distribution, Master's Thesis, Department Of Computer Science, Rochester Institute Of Technology, 2000.

[93] Chen K., Reconciliation By Public Discussion: Throughput And Residue Error Rate, Unpublished Draft, 2001.

[94] Dusek, M., Lütkenhaus,N. And Hendrych, M., Quantum Cryptography. http://arxiv.org/abs/quant-ph/0601207 (Ziyaret Tarihi: 25 Kasım 2018).

KİŞİSEL YAYINLAR VE ESERLER

[1] **Toyran M.**, Kayran A. H., Örtüşmüş Düşük Çözünürlüklü Görüntülerden Süper Çözünürlüklü Görüntü Oluşturma, *IEEE SIU Kurultayı*, Didim/Aydın, 20-22 Nisan 2008.

[2] **Toyran M.**, Toyran M., Öztürk S., Kuantum Anahtar Dağıtımında (KAD) Bilgi Uzlaştırma (BU) İçin Cascade Tekniği, *IEEE 25. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU 2017)*, Antalya, Türkiye, 15-18 Mayıs 2017

[3] **Toyran M.**, Toyran M., Öztürk S., New Approaches to Increase Efficiency of Cascade Information Reconciliation Protocol, *7th International Conference on Quantum Cryptography*, Cambridge, UK, 18-22 September 2017.

[4] **Toyran M.**, Toyran M., Öztürk S., Cascade Protokolünün Hızlı ve Verimli Gerçeklemeleri, *IEEE 26. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU 2018)*, İzmir, Türkiye, 2-5 Mayıs 2018.

[5] **Toyran M.**, Toyran M., Öztürk S., Cascade Protokolü, Verimli Gerçeklemeleri ve Analizleri, *SAVTEK 2018, 9. Savunma Teknolojileri Kongresi*, ODTÜ, Ankara, 27-29 Haziran 2018.

[6] **Toyran M.**, Toyran M., Öztürk S., Optimized Cascade Protocol For Efficient Information Reconciliation In Quantum Key Distribution Systems, *Quantum Information & Computation (QIC)*, 2018, **7&8**, 0541-0552.

ÖZGEÇMİŞ

Metin Toyran, 16.06.1983 tarihinde Tokat'ta doğdu. İlk ve orta eğitimini İstanbul'da tamamlamıştır. İstanbul Orhan Cemal Fersoy Lisesi'nden 2001 yılında mezun olmuş ve aynı yıl İstanbul Teknik Üniversitesi (İTÜ) Telekomünikasyon Mühendisliği bölümünde lisans eğitimine başlamıştır. 2005 yılında İTÜ'deki lisans öğrenimini bölüm birincisi olarak tamamlamıştır. Lisans eğitimi sonunda geliştirilen bitirme projesi ile Siemens tarafından Mükemmellik Ödülü'nü kazanmaya hak kazanmıştır. Aynı yıl içerisinde İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü'nde Telekomünikasyon Mühendisliği programında lisansüstü eğitimine başlamıştır. Yüksek Lisans eğitimi boyunca TÜBİTAK Yurtiçi Yüksek Lisans Bursu almıştır. 2008 yılında "Düşük Çözünürlüklü Görüntülerden Süper Çözünürlüklü Görüntü Oluşturma" isimli yüksek lisans tezini tamamlayarak mezun olmaya hak kazanmıştır. 2008 yılında yüksek lisanstan mezun olduktan sonra askerlik görevini yapmak üzere Malatya 2. Ordu Komutanlığı'ndaki birliğine teslim olmuştur. Askerlik görevini tamamladıktan sonra, 2010 yılında Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü'nde Doktora öğrenimine başlamıştır. 2005 yılından itibaren Türkiye Bilimsel ve Teknolojik Araştırmalar Kurumu (TÜBİTAK) - Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü (UEKAE)'de araştırmacı olarak görev yapmaktadır.