İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

KUANTUM BİLGİSAYARLARDA POLİNOM İNTERPOLASYONU İLE ANAHTAR DAĞITIMI

YÜKSEK LİSANS TEZİ

Berrak UZUN

Matematik Mühendisliği Anabilim Dalı

Matematik Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Ergün YARANERİ

ARALIK 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

KUANTUM BİLGİSAYARLARDA POLİNOM İNTERPOLASYONU İLE ANAHTAR DAĞITIMI

YÜKSEK LİSANS TEZİ

Berrak UZUN
509161202

Matematik Mühendisliği Anabilim Dalı

Matematik Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Ergün YARANERİ
Eş Danışman: Doç. Dr. Enver ÖZDEMİR

ARALIK 2019

İTÜ, Fen Bilimleri Enstitüsü'nün 509161202 numaralı Yüksek Lisans Öğrencisi Berrak UZUN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "KUANTUM BİLGİSYARLARDA POLİNOM İNTERPOLASYONU İLE ANAHTAR DAĞITIMI" başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Doç. Dr. Ergün YARANERİ** İstanbul Teknik Üniversitesi	.....
Eş Danışman :	**Doç. Dr. Enver ÖZDEMİR** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :	**Doç. Dr. Burcu TUNGA** İstanbul Teknik Üniversitesi	.....
	Dr. Öğr. Üyesi Elif Segah ÖZTAŞ Karamanoğlu Mehmetbey Üniversitesi	.....
	Dr. Öğr. Üyesi Tuğba YILDIRIM İstinye Üniversitesi	.....

Teslim Tarihi : 14 Kasım 2019
Savunma Tarihi : 13 Aralık 2019

Eşime ve aileme,

ÖNSÖZ

Yüksek lisans eğitimim boyunca bilgi ve deneyimlerini benimle paylaşan, güvenini ve desteğini her daim hissettiren değerli danışman hocalarım Doç. Dr. Enver Özdemir'e ve Doç. Dr. Ergün Yaraneri'ne sonsuz teşekkürlerimi sunarım. Bununla birlikte tüm eğitim-öğretim hayatım boyunca bana gösterdikleri desteği esirgemeyen anne ve babama, çalışmalarım esnasında bana güç verip destekleyen eşime teşekkürlerimi borç bilirim.

Aralık 2019

Berrak UZUN
(İş Analisti)

İÇİNDEKİLER

Sayfa

KISALTMALAR

AES **:** Gelişmiş Şifreleme Standardı (Advanced Encryption Standard)

RSA **:** Ron Rives, Adi Shamir, Leonard Aldeman

DES **:** Veri Şifreleme Standardı (Data Encryption Standard)

ASCII **:** Bilgi Değişimi İçin Amerikan Standart Kodlama Sistemi (American Standard Code for Information Interchange)

SEMBOLLER

$\boldsymbol{\varphi(n)}$	**:** n sayısının kendisinden küçük pozitif ve kendisi ile asal olan tam sayıların sayısını belirtir. Euler Totient fonksiyonu olarak adlandırılır.
$\boldsymbol{\gcd(a,b)}$	**:** a ve b sayısının en büyük ortak bölenini ifade etmektedir.
$\boldsymbol{Z_p}$	**:** p modülüne göre kalan sınıfların kümesidir.
$\boldsymbol{GF(p)}$	**:** p elemanlı sonlu bir cisimi ifade etmektedir.

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

KUANTUM BİLGİSAYARLARDA POLİNOM İNTERPOLASYONU İLE ANAHTAR DAĞITIMI

ÖZET

Bu çalışmada, kuantum bilgisayarlarda gönderici ve alıcı arasında anahtar paylaşımının güvenli ve hatasız bir şekilde paylaşan algoritmaların detaylı analizi amaçlanmıştır. Bu paylaşım, Newton bölünmüş farklar polinom interpolasyon yöntemi kullanılarak sağlanmaya çalışılmıştır. Yeni çıkartılan kuantum anahtar değişim algoritması birçok yönden incelenmiş, gerçeklenmiş ve geliştirilmesine yönelik tavsiyeler ortaya konmuştur.

Birinci bölüm giriş bölümü olup, teknolojinin gelişimi, kriptolojiye ait bilgiler ve kuantum kriptografinin amaçlarından bahsedilmiştir.

İkinci bölümde kriptolojiye ait terminolojik terimler, klasik şifreleme algoritması olan Sezar şifreleme ve modern şifreleme algoritmalarından asimetrik ve simetrik şifrelemelere ait açıklamalarda bulunulmuştur.

Üçüncü bölümde kriptoloji biliminin asal sayılar, sonlu cisimler ve modüler aritmetikle iç içe olmasından dolayı sonlu cisimlere ait bilgiler ve sonlu cisimler üzerinde polinom işlemleri açıklanmış ve bu çalışmada da sıklıkla kullanılan modüler aritmetikte ters alma işleminin kullanımlarına ilişkin bilgiler verilmiştir.

Dördüncü bölümde klasik bilgisayar ile kuantum bilgisayarlara ait bilgiler ele alınmıştır ve çalışma şekillerine ait bilgilere yer verilmiştir.

Beşinci bölümde kuantum bilgisayarlarda anahtar dağıtımı için tasarlanan ilk protokol olan BB84 protokolünden bahsedilmiş ve kuantum kriptografide hata düzeltme protokolü olan Cascade protokolüne dair bilgiler yer almaktadır.

Altıncı bölümde, beşinci bölümde ele alınan Cascade protokülünün eksiklerine dayanarak gizli anahtarın güvenli bir şekilde paylaşılması için polinom interpolasyonu kullanıldığı belirtilip, Newton bölünmüş farklar yöntemi ile polinom interpolasyonundan bahsedilmiştir. Kuantum bilgisayarlarda anahtar değişiminde polinom interpolasyonunun nasıl kullanıldığı hakkında bilgiler verilip protokolün güvenlik analizi yapılmıştır. İmplementasyonda, $(x_1, f_1), (x_2, f_2), \dots, (x_n, f_n)$ değerleri kullanılarak $r < (n-2)$ olmak üzere r. dereceden polinom elde edilmiştir. Fakat bu algortimanın çalışabilmesi için kullanılan noktalar dışında en az bir adet noktanın karşı tarafa doğru ulaşması gerekmektedir. Bu polinomun sonuçlarıyla gerçek polinomun sonucu karşılaştırılmış olup doğruluk değeri yüzde cinsinden hesaplanmıştır. Hesaplanan doğruluk değeri istenilen doğruluk değerinden büyükse algoritma sonlandırılır. Aksi durumda yeni x değerleri kullanılarak polinom elde edilip doğruluk değerine tekrar bakılır ve doğruluk değeri istenilen doğruluk değerinden büyük olana kadar işleme devam edilir. İşlem sonunda hangi x noktasının hatalı olduğu tespit edilir.

Bu çalışmanın yapılmasının amacı kuantum fiziği yasaları ve kuantum kanallarındaki fotonların fiziksel özelliklerinden dolayı anahtar iletimi esnasında anahtarın değişime uğramasından veya araya giren bir kişinin anahtar ile ilgili bilgileri ele geçirmesinden dolayı gerçekleşecek sorunların giderilmesidir. Bu implementasyonda programlama dili olarak C tercih edilmiştir.

KEY DISTRIBUTION WITH POLYNOMIAL INTERPOLATION IN QUANTUM COMPUTERS

SUMMARY

The person holding the information or the impact on the vital activities of institutions determine the value of information. For example, for the sake of security and continuity in terms of states, the national intelligence convert data into information and protect the information in the best way. To protect data is requied to aware of possible threat. Threatening security information elements is grouped in 4 topics such as information loss (security breach), information changes (violation of integrity) and handled by someone else passages (violation of privacy).

Cryptology deals with the confidentiality, integrity, authentication and nonrepudiation of information. Confidentiality means that information cannot be understood by unauthorised people. Integrity for an entire connection or for a single piece of data, no modification, insertion, rearrangement, as the message was not guaranteed. Authentication, guarantee that the sender of the transmitted message is actually the sender. The nonrepudiation is that the recipient or the sender cannot deny the transmitted message. So that when a message is sent, the recipient can prove that the sender has sent the message, and the sender can prove that the recipient has received the message.

As the Internet network expands, elements that threaten information security have become more intimidating. The development of new methods and technologies for information security or the improvement of existing ones has become a constant cycle of need.

When this continuous need cycle is examined, it is seen that cryptology is mostly used in military and diplomatic fields from past to present. Today, communication channels, debit transactions, credit card applications, internet browsing, e-commerce and e-government applications, mobile phones, billing systems, city networks, missile systems, transportation tools etc. have used cryptology.

Because of the extent of the areas of cryptology is the growing and differentiating threats against information security. This situation has triggered a growing network of internet. Elements of this threat people, institutions and nations.

In order to ensure that the information is shared, it is necessary to know who is communicating with, to ensure that the information is sent correctly or completely to the other party and not to be monitored by others. In order to accomplish all this, components which are the most fundamental elements of cryptography science are designed. A variety of encryption and decryption algorithms developed through cryptography science in order to send and receive data securely. No matter how complex the encryption algorithm is, when the key used in encryption is seized, the password does not matter. Therefore, the most important examination of cryptology is the transmission of the key and the protocols used in key transmission.

Numerous algorithms have been developed for key transmission used in encryption. However, these algorithms are not absolutely safe. Quantum cryptography provides this security for now, the key between the receiver and the transmitter makes the exchange rule safe. The technique of quantum cryptography is based on Heisenberg's uncertainty principle, a fundamental law of physics. According to this principle, two

properties of a photon, which is the basic element of quantum mechanics, cannot be known at the same time. This makes it impossible to clone a photon in the communication channel.

In this study, it is aimed to analyze the algorithms that share the key safely and correct sharing between sender and receiver in quantum computers. These sharing algorithms are tried to be achived by using Newton divided differences polynomial interpolation method. In this quantum key exchange algorithm has analyzed in every aspect and implemented.

Improvement of the technology, informations about cryptology and purposes of quantum cryptography have explained at the first chapter.

In the next chapter, it has been explained that terminological terms which belong to cryptology such as plaintext, ciphertext, encryption, decryption and key. On the other hand, classic and modern encryption algorithms have examined. The modern encryption algorithm is divided into two. For example, Caesar algorithm as classical encryption algorithm, AES algorithm as symmetric encryption algorithm and RSA algorithm as asymmetric encryption algorithm and their properties are mentioned. Then the diffee-helman algorithm, which is the key exchange algorithm, is explained.

At third chapter, because of cryptography being related with prime numbers, finite field and modular aritmetics some knowledge has given such that information about finite field, polynomial calculation on finite field and modular inverse.

After third chapter, information about the properties and working methods of classical and quantum computers have given. The main difference between classical and quantum computers is working principle, classical computers use bits (0 or 1), quantum computers use qubits. Quantum computers have parallel process properties; therefore, it is working fast with respect to classical computers.

At fifth chapter, information about quantum key distribution and quantum channel are given. There are two channels on communication, one of channels is classical chanel and the other one is quantum channel. The quantum channel is used for key sharing. All other steps are performed on the classical channel. The first protocol designed for key distribution in quantum computers, the BB84 protocol, has mentioned and include information about the Cascade protocol, the error correction protocol.

At sixth chapter, Newton divided difference polynomial interpolation has used for sharing secret key safely contrast to lack of security in cascade algorithm. Information has given about how polynomial interpolation is used in key exchange in quantum computers and the security analysis of the protocol has performed. It has explained that how to implement Newton divided difference polynomial interpolation and security analyze. $(x_1, f_1), (x_2, f_2), \dots, (x_n, f_n)$ values have used for fitting r^{th} degree polynomial for all $r < (n-2)$. One value must transmit perfectly except used values for interpolation to work algorithm properly. And then, percent accuracy is calculated with respect to difference between polynomial interpolation and real values receiving from channel. The algorithm is running unless percent accuracy is big enough. End of the algorithm, place of wrong number is detecting and correcting for secure communication. Besides, quantum key sharing protocol has investigated from the point of security analysis. For Eve obtain the key, Eve must perform a brute-force attack. For security analysis, it is determined that the quantum and classical computers perform brute force attack according to the key length. The main reason for error in

transmitting number is physical properties of quantum which is interrupted by Eve. C programming language is used for implementation.

1. GİRİŞ

Teknolojinin hızla ilerlemesi ile birlikte internet hemen hemen her birey için vazgeçilmez olmuştur. Bu ilerleme ile birlikte insanlar ağ üzerinden bilgi alışverişini kullanmaya başlamışlardır. Günümüzde iletişim ağları, bankamatik işlemleri, internet ortamında gezinti, alışveriş, ulaşım sistemleri (GPS) vb. alanlarda kullanılmaktadır. İnternet ağı büyüdükçe bilgi güvenliği konusu da ön plana çıkmaktadır ve bununla birlikte çeşitli tehdit unsurları oluşmaya başlamıştır.

Bilgilerin paylaşımınının güvenli olması için kim ile haberleşme gerçekleştirildiğinin bilinmesi, karşı tarafa bilgilerin doğru ya da eksiksiz gönderildiğinden emin olunması ve başkaları tarafından izlenmiyor olunması gerekmektedir. Tüm bunları gerçekleştirmek için kriptografi bilimin en temel öğeleri olan bileşenler tasarlanmaktadır. Bu çalışmanın ilk kısmında en temel kriptografik bileşenler hakkında kısa bilgi verilecek özellikle bunlardan simetrik ve asimetrik şifreleme algoritmaları detaylı anlatılacaktır. Simetrik şifreleme algoritmalarında, şifreleme ve deşifreleme sırasında aynı gizli anahtar kullanılmaktadır. Asimetrik şifreleme algortimalarında ise gönderici ve alıcı kendilerine ait anahtar çifti kullanmaktadır. Kripto sistemleri güvenli haberleşmenin en temel yapıtaşlarını oluşturmaktadır. Güvenli haberleşme ile kastedilen hususlar aşağıda verilmiştir.

- Gizlilik: Bilgi iletimi sırasında taşınan bilginin içeriğinin gizli kalmasıdır.
- Bütünlük: Taşınan bilginin üçüncü kişiler tarafından bilgi iletimi sırasında değiştirilememesidir.
- Kimlik Doğrulama: Mesajı alan kişinin mesajın doğru kişiden geldiğinden ve mesajı gönderen de doğru kişinin okuduğundan emin olmasıdır.
- İnkar Edememe: Mesajı ileten kişinin daha sonra mesajı kendisinin oluşturduğunu veya gönderdiğini inkar edememesidir [14].

Bu özelliklerin her biri ayrı kriptografik bileşenler ile sağlanmaktadır. Mesela, gizlilik için en çok simetrik şifreleme kullanılmaktadır. Simetrik şifreleme kullanıldığından

haberleşme yapan iki tarafında aynı gizli anahtara sahip olması gerekmektedir. Bu durumda gizli anahtarın güvenli bir şekilde iletilmesi ve saklanabilmesi ciddi bir problem olarak karşımıza çıkmaktadır. Anahtar iletimi sırasında araya girilerek anahtar elde edilebilmekte ve bunun anlaşılması gerçekten çok zor olabilmektedir. Anahtar değişimi veya kimlik doğrulama ise genelde asimetrik şifreleme algoritmaları ile yapılmaktadır. Örneğin bunlardan biri RSA'de ise açık anahtar herkes tarafından bilinmektedir ve iki sayının çarpımından oluşan açık anahtarın çarpanları bilinirse gizli anahtar elde edilir. Fakat günümüz bilgisayarlarının büyük sayılar için çarpanlara ayırma işlemini gerçekleştirmesi neredeyse imkansız gözükmektedir. Her ne kadar imkansız görünsede bu konuda yapılan çalışmalar ve bilgisayar sistemlerinin gelişmesi elimizde bulunan asimetrik şifreleme algoritmalarının geliştirilmesini şart koşmaktadır. Mesela, günümüzde kullanılan tüm anahtar değişimi algoritmaları kuantum bilgisayarlar tarafından kırılabilmektedir. Kuantum sistemleri için daha güçlü kriptosisteme ihityaç duyulmasından dolayı, adını sıklıkla duyduğumuz kuantum mekaniği yasalarını kullanarak anahtar iletimini gerçekleştiren algoritmalar doğmuştur. Kuantum kriptografi, şifrelemede kullanılacak anahtarı üretmek ve anahtarı güvenli bir kanal üzerinden alıcıya iletmek amacıyla kullanılan bir yöntemdir. Bu yöntem fizik kurallarını ve fotonların özelliklerini kullanmaktadır. Böylece iletişimin güvenli olmasını sağlamak hedeflenmektedir. Fakat bazı işlemlerin herkese açık iletişim kanallarından gerçekleştirilmesi paylaşılan anahtar ile ilgili bilgilerin açığa çıkmasına neden olmaktadır. Bu çalışmada, anahtar paylaşımındaki bilgilerin açığa çıkma oranını en aza indirmek için polinom interpolasyonu yöntemi detaylı bir şekilde incelenmiştir. Polinom interpolasyonu kullanılarak elde edilen yeni yöntem detaylı incelenmiş, güvenlik analizi yapılmış ve gerçeklenmiştir.

2. KRİPTOLOJİ

Bu bölümde kriptolojiye ait terimler ve şifreleme algoritmaları ile ilgili bilgiler yer almaktadır.

2.1 Tanımlar

Kriptoloji, mesajların belli sisteme göre şifrelenmesi ve mesajın alıcıya güvenli bir şekilde gönderilmesi ve bu mesajın deşifre edilmesini sağlayan bileşenlerin oluşturduğu bilim dalıdır. Kriptoloji iki ayrı bilim dalından oluşmaktadır. Bunlar; kriptografi, yani mesajı şifreli hale dönüştürme, anlamsız hale getirme ve kriptoanaliz, yani şifrelenmiş mesajı çözme ya da analiz ederek şifreleme sisteminin zayıf ve kuvvetli yönlerini ortaya koymaktır. Kriptolojiye ait temel terimler şu şekildedir:

Düz metin (Plaintext): Bir şifreleme algoritması kullanılarak şifrelenecek temel veriye düz metin denir.

Şifreli metin (Ciphertext): Düz metinin şifrelenmiş halidir. Bu metin bir birey ya da bilgisayar tarafından okunsa bile anlamsızdır. Uygun bir şifreleme algoritmasının tersi veya deşifreleme algoritması şifreli metine uygulandığında düz metin elde edilmiş olur.

Şifreleme (Encryption): Düz metinin şifreli metne dönüştürülmesidir.

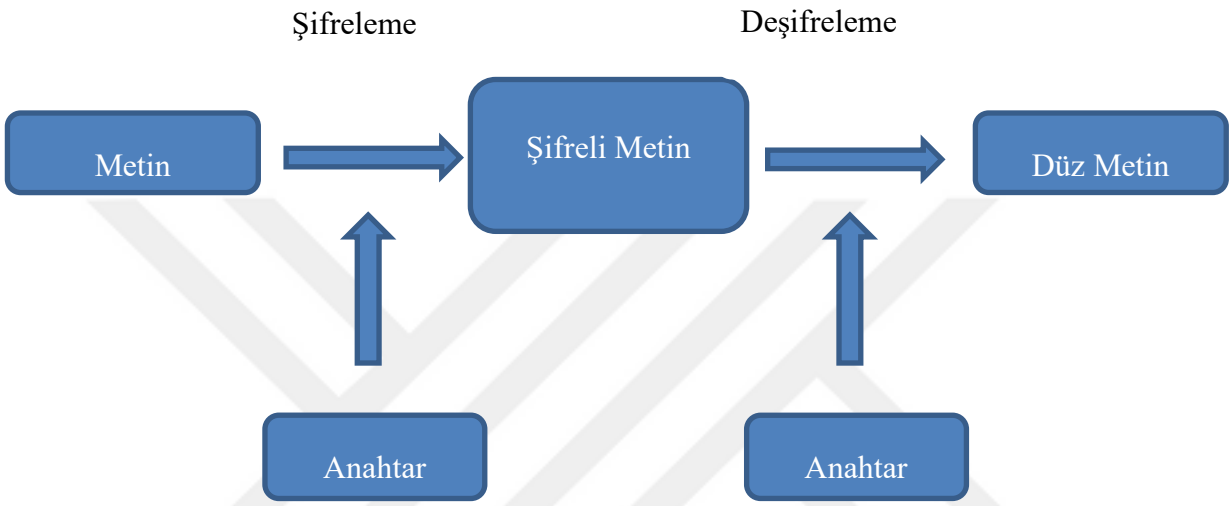
Deşifreleme (Decryption): Şifreleme işlemini tersine gerçekleştirir ve şifreli metni düz metine dönüştürür.

Anahtar (Key): Şifreleme ve deşifrelemede kullanılır. Yanlızca birbirleriyle şifrelenmiş bilgi alışverişinde bulunan bireyler veya sistemler tarafından bilinir ve kullanılan algoritmaya bağlı olarak uzunluğu değişen bir karakter dizisinden oluşmaktadır.

Kriptografinin temel çalışma mekanizması Şekil 2.1 ‘de verilmiştir.

2.2 Kriptoloji Algoritmaları

Algortimalar klasik ve modern olmak üzere 2'ye ayrılmakta olup, aşağıda örnekleme amacıyla en basit simetrik şifreleme metodlarından Sezar (Caeser) şifreleme algortimasından bahsedilecektir. Yukarıda belirtildiği gibi modern şifreleme algortimaları kullanılan şifreleme ve deşifreleme anahtarlarına göre 2'ye ayrılmaktadır. Bunlar simetrik ve asimetrik şifreleme algortimalarıdır.



Şekil 2.1 : Temel şifreleme algoritması.

2.2.1 Sezar şifreleme algortiması

Bu algoritma Sezar tarafından M.Ö 58 yılı civarında kullanılmıştır. Sezar, askeri mesajların düşmanın eline geçmesini engellemek amacıyla mesajın her harfini değiştirmiş ve böylece mesaj düşmanın eline geçse bile mesaj anlamlı olmamasını sağlamıştır.

Bu yöntem alfabetik sıralamanın önemli olduğu bir şifreleme algoritmasıdır. Elimizde öncelikle bir alfabe bulunmaktadır ve metni şifreleyecek kişi ile metni deşifreleyecek kişi belli bir kaydırma değerinde anlaşmaktadır. Bu kaydırma değeri bizim anahtar sayımız olacaktır. Düz metinde bulunan her bir harfi anahtar sayısı kadar ileriye götürerek yeni bir şifreli metin elde edilecektir [15]. Bunu bir örnekle açıklamak gerekirse;

Türkçe alfabeyi kullanarak "KİTAP" kelimesini şifreli metin haline dönüştürelim. Kaydırma değerimiz 2 olsun. Şekil 2.2 ve Şekil 2.3'te orjinal ve kaydırılmış alfabe yer

almakta olup orjinal alfabede “kitap” kelimesinin kaydırılmış alfabedeki değerlerini yazdığımızda şifreli metnimiz “MKÜCS” şeklinde elde edilmektedir [1].

A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z

Şekil 2.2 : Orijinal alfabe.

C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	A	B

Şekil 2.3 : Kaydırılmış alfabe.

2.2.2 Simetrik şifreleme (Gizli anahtarlı kriptografi)

Simetrik şifrelemede, şifreleme ve deşifreleme işlemleri esnasında tek bir anahtar kullanılır. Bu anahtara gizli anahtar (secret key) denir. Gönderici düz metni gönderirken mesajı bir anahtarla şifreler ve alıcı mesajı aynı anahtarla deşifreler (Şekil 2.4). Bu anahtar sadece birbiri ile şifreli haberleşen taraflarca bilinmelidir. Bu yüzden anahtar alıcıya güvenli iletişim kanalları aracılığı ile iletilmelidir [2].

Şekil 2.4 : Simetrik şifreleme.

Simetrik algoritmalara örnek olarak AES, DES, 3DES, Blowfish, RC4 verilebilir. Günümüzde artık standart hale gelen bu algoritmalardan AES’i ele alacağız.

AES:

- Blok şifrelemedir.
- Girdi ve çıktı verileri sabit olup 128 bittir.
- Anahtar uzunluğu 128, 192 veya 256-bit uzunluğunda olabilir.
- AES, durum denilen 4x4 lük bayt matrisi üzerinde çalışılır. Matris üzerindeki işlemler sonlu cisim (finite field) üzerinde yapılmaktadır.
- AES-128, 10 turdan oluşmaktadır. İlk olarak 128 bitlik anahtar 10 turda farklı şekilleriyle kullanılması için genişletilmektedir.
- Tur anahtarını ekleme adımı gerçekleştirilir.

- Bu adımdan sonra 10 tur gerçekleşir.
- Bu turlar esnasında sırasıyla bayt değiştirme, satırları kaydırma, sütunları karıştırma ve tur anahtarını ekleme adımları uygulanmaktadır.
- 10. Adımda sütunları karıştırma işlemi uygulanmamaktadır.

Simetrik şifrelemenin avantaj ve dezavantajları şu şekildedir:

- Algoritmalar hızlıdır.
- Donanımla birlikte etkin bir şekilde kullanılabilir.
- Güvenlidir.
- Anahtar dağıtımı gerekli ve aynı zamanda zordur.
- Bütünlük ve kimlik doğrulama işlemlerini güvenli şekilde gerçekleştirmek zordur [19].

2.2.3 Asimetrik şifreleme (Açık anahtarlı kriptografi)

Asimetrik şifrelemede, şifreleme ve deşifreleme işlemleri iki farklı anahtar ile gerçekleştirilmektedir. Bu anahtarlar açık (public) ve özel (private) anahtarlar olarak adlandırılmaktadır (Şekil 2.5). Açık anahtarlar ağ üzerindeki herkes ile paylaşılabilir fakat özel anahtar sadece kişinin kendisi tarafından bilinmelidir. Bu iki anahtar arasında matematiksel bir bağlantı olmasına rağmen anahtarın birini kullanarak diğer anahtarı elde etmek neredeyse imkansızdır [2].

Şekil 2.5 : Asimetrik şifreleme.

Asimetrik şifreleme algortimalarına örnek olarak RSA, Diffie-Helman ve El-Gamal verilebilir. Bu algoritmalardan RSA'i ele alalım.

RSA:

- 1977 yılında Ron Rives, Adi Shamir ve Leonard Aldeman tarafından bulunmuş olup, adı bu üç kişinin soy isimlerinin baş harflerinden oluşur.
- Birbirinden farklı yeterince büyük p ve q gibi iki asal sayı seçilir.

$$n = pq \tag{2.1}$$

- n değeri denklem 2.1 yardımıyla hesaplanır

$$\varphi(n) = (p-1)(q-1) \tag{2.2}$$

- n'nin Euler sayısı $\varphi(n)$, denklem 2.2 ile hesaplanır.

$$1 < e < \varphi(n) \tag{2.3}$$

- 2.3'deki eşitsizliğini sağlayan ve $\varphi(n)$ ile aralarında asal olan bir e sayısı bulunur. e, n şifreleme anahtar çiftleridir ve herkes tarafından bilinmesi için yayınlanır.

$$Ebob\big(e, \varphi(n)\big) = 1 \tag{2.4}$$

- Denklem 2.4'ten dolayı

$$de \equiv 1\ mod(\varphi(n)) \tag{2.5}$$

$$1 < d < \varphi(n) \tag{2.6}$$

- 2.5 denkleminin 2.6'da yer alan eşitsizliğini sağlayan ve bir tek olan d doğal sayısı bulunur.
- Böylece d, n şifre çözme anahtar çifti elde edilmiş olur.

$$c \equiv m^e\ mod\ n \tag{2.7}$$

$$m \equiv c^d\ mod\ n \tag{2.8}$$

- Şifreleme işlemi 2.7'de yer alan denklemle hesaplanır.
- Deşifreleme işlemi 2.8'de yer alan denklemle hesaplanır.

Asimetrik şifrelemenin avantaj ve dezavantajları şu şekildedir:

- Şifrelerin deşifre edilmesi nispeten daha zordur.

- Kimlik doğrulama, bütünlük ve gizlilik adımlarını gerçekleştirmek için güvenli bir yoldur.
- Şifrelemede kullanılan özel anahtarların karşılıklı aktarılmasına gerek yoktur.
- Şifrelemede iki anahtar kullanıldığından dolayı sayısal imza ile inkar edilememezlik sağlanır.
- Algoritma yavaş çalışır.
- Anahtar uzun olduğundan bit sayıları da uzundur [19].

2.3 Diffie-Helman Anahtar Değişimi

Açık anahtarlı sistemlerle gerçekleştirilen şifreleme işlemleri yavaş işlemlerdir. Fakat anahtar dağıtım kolaylığı açısından daha fazla tercih edilmektedir [5]. 1976 yılında, Diffie-Hellman anahtar değişimi açık kanal üzerinde anahtar dağıtım problemi için geliştirilen ilk matemaiksel çözümdür.

Diffie-Helman algoritmasının geliştirilmesi ile birlikte simetrik şifrelemelerde gizli anahtar paylaşma sorunu büyük ölçüde ortadan kalkmıştır. Diffie-Helman algoritması bir şifreleme algortiması olmayıp ortak gizli anahtarı belirlemek için kullanılmaktadır [6].

Algoritma şu şekilde çalışmaktadır:

p, Z_p de ayrık logaritma problemi pratikte çözülemeyecek kadar yeterince büyük bir asal sayı olsun.

q, Z_p'de primitif bir kök olmak üzere, p ve q herkes tarafından biliniyor olsun. Primitif kök; tüm $s < p - 1$ olmak üzere 2.9 denkleminin sağlanmasıdır.

$$q^s \not\equiv 1 \ (mod\ n) \tag{2.9}$$

Alice $0 \leq a \leq p - 2$ eşitsizliğini sağlayan bir a sayısı seçer, 2.10 denlemini hesaplar ve bunu Bob'a gönderir.

$$A \equiv g^a \ (mod\ p) \tag{2.10}$$

Bob, $0 \leq b \leq p-2$ eşitsizliğini sağlayan bir b sayısı seçer, 2.11 denklemini hesaplar ve bunu Alice'e gönderir.

$$B \equiv g^b \ (mod\ p) \tag{2.11}$$

Alice A yı Bob'a, Bob B yi Alice'e iletir. Alice seçtiği a gizli sayısını kullanarak, 2.12 denklemini, Bob seçitği b gizli sayısnı kullanarak 2.13 denklemini hesaplar.

$$A' \equiv B^a \ (mod\ p) \tag{2.12}$$

$$B' \equiv b \ (mod\ p) \tag{2.13}$$

Hesaplamalar sonucunda denklem 2.14 elde edilir.

$$A' \equiv B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \equiv B' \ (mod\ p) \tag{2.14}$$

Böylece Alice ve Bob ortak gizli anahtarlarını A' ve B' yi kullanarak elde ederler [3, 4, 5].

3. ASAL SAYILAR VE MODÜLER ARİTMETİK

Bu bölümde asal sayılar, modüler aritmetik, öklid algoritması, modüler çarpma ve sonlu cisimler ele alınmıştır.

3.1 Asal Sayılar

1 den büyük olan, 1 ve kendisinden başka böleni olmayan tam sayılara asal sayı denir. 2,3,5,7,11,13 asal sayılara örnektir.

3.2 Aralarında Asal Sayılar

a ve b iki tam sayı olmak üzere $\gcd(a, b) = 1$ koşulu sağlanıyorsa bu sayılara aralarında asal sayılar denir.

3.3 Modüler Aritmetik

Bir x sayısının y sayısına bölümünden kalan r ve bölüm q ile gösterildiğinde x, 3.1 ile ifade edilebilir:

$$x = q.y + r \tag{3.1}$$

Yani y sayısına bölünen bir sayının bu bölümden kalanı $r < y$ dir. y'nin kalan denklik sınıfını kümesindeki sayılar $\{0, \dots, y-1\}$ şeklindedir. y'nin kalan denklik sınıfı kümesine y tabanına göre mod denmektedir ve

$$A \equiv B(mod\ y) \tag{3.2}$$

şeklinde ifade edilmektedir.

3.3.1 Öklid algoritması

Öklid algoritması a ve b gibi iki doğal sayının en büyük ortak bölenini $\big(\gcd(a, b)\big)$

ve $^1/_a\ (mod\ b)$ ifadesini bulmak için kullanılan bir yöntemdir.

Algoritma şu şekilde çalışmaktadır:

- $a > b$ olmak üzere, a sayısı b sayısına bölünür. Bölüm q_1, kalan r_1 olsun.

$$a = b.q_1 + r_1 \tag{3.3}$$

- Daha sonra ikinci bölme işlemi gerçekleştirilir. b, r_1'e bölünür ve bölüm q_2, kalan r_2 olur.

$$b = q_2.r_1 + r_2 \tag{3.4}$$

- Üçüncü adım olarak r_1, r_2'ye bölünür ve bölüm q_3, kalan ise r_3 olur.

$$q_2 = q_3.r_2 + r_3 \tag{3.5}$$

- Bu işlemler aynı kalanı 0 bulana kadar tekrarlanarak öyleki son adımda r_{n-1}, r_n'e bölünür ve bölüm q_{n+1}, kalan ise $r_{n+1} = 0$ olur.

$$r_n - 1 = q_n + 1.r_n + r_n + 1 \tag{3.6}$$

- $r_{n+1} = 0$ olduğundan dolayı r_n değeri a ve b tamsayılarının en büyük ortak böleni olarak elde edilir. Bu da $\gcd(a, b) = r_n$ şeklinde ifade edilir. Bu algoritma ile aynı zamanda $ax + by = r_n$ ifadesini sağlayan x ve y değerleri de bulunur [17]. Bu eşitlikte r_n, 1'e eşitse, x denklem 3.7 ile hesaplanır.

$$r_n = 1 \Rightarrow\ x = {}^1/_a\ (mod\ b) \tag{3.7}$$

3.3.2 Modüler çarpma

Bu başlık altında 3.2.1 de bahsedilmiş olan Öklid Algortimasının gemişletilmiş hali incelenecek olup bu metodun amacı belirli bir tabana göre verilen sayının kolayca tersini bulmaya yöneliktir. Bilinen bir e ve p sayısı için 3.8 denklemi çözülmekte olup bulunacak olan sayı d sayısıdır bu sayı p tabanında e sayısının tersidir.

Bunu bir örnekle açıklamak istersek;

$$d.e \equiv 1\ (mod\ p) \tag{3.8}$$

p=3000 ve e=197 sayıları verilmiş olsun. 197 sayısının 3000 tabanında bulunması istendiğinde,

$$3000 = 15.197 + 45$$

$$197 = 4.45 + 17$$

$$45 = 2.17 + 11$$

$$17 = 1.11 + 6$$

$$11 = 1.6 + 5$$

$$6 = 1.5 + 1$$

Bu adımdan sonra 3000 tabanında

$$1 = 197.x + 3000.y$$

şeklinde bir ifade elde edilmelidir.

$$1 = 6 - 1.5$$

$$1 = 2.6 - 1.11$$

$$1 = 2.17 - 3.11$$

$$1 = 8.17 - 3.45$$

$$1 = 8.197 - 35.45$$

$$1 = 533.197 - 35.3000$$

Bu adımda elde etmek istediğimiz denkleme ulaştık ve 3000 tabanında işlemler yapıldığından dolayı elde edilen denklemin son hali

$$1 = 533.197\ (mod\ 3000)$$

şeklindedir. Bu da 3000 tabanında 197 sayısının tersinin 533 sayısı olduğunu göstermektedir.

3.4 Sonlu Cisimler

Kriptolojide genellikle sonlu cisimler kullanıldığı için, polinom sonlu cisimler üzerinde inşa edilmektedir.

Bunun için öncelikle sonlu cisimler (Finite Field)'in tanımını ele alalım. Sonlu cisimler aynı zamanda Galois cisimleri olarakta adlandırılmaktadır. Sonlu cisimlerin eleman sayıları (order) asal sayı veya asal sayının kuvveti şeklindedir. Her asal sayı kuvveti için kesinlikle bir $GF(p^n)$ sonlu cismi vardır ve F_{p^n} şeklinde ifade edilmektedir. $GF(p)$, eleman sayısı p olan asal cisimdir ve $mod\ p$ ye göre kalan sınıfların cismidir. Elemanları $0,1,\dots,p-1$ şeklindedir. $GF(p)$ cisiminde $a = b$ ile $a = b\ (mod\ p)$ aynı ifadelerdir [7].

$n > 1$ olmak üzere $GF(p^n)$, katsayıları $GF(p)$'nin elemanları olan polinomların denklik sınıflarının cismi olarak ifade edilebilir. Sonlu cisimler üzerinde işlemlere dair örnek yapacak olursak; $GF(7)$ sonlu cismi (Z_7 ile de gösterilebilir) üzerinde polinomlar ile toplama ve çarpma işlemleri yapalım.

$$f(x) = 5x^2 + 2x + 6$$

$$g(x) = 2x + 1$$

olsun.

$f(x) + g(x) = 5x^2 + 4x + 7$ dir. Fakat $mod\ 7$ ye göre işlem yapıldığından dolayı elde edilen sonuç $f(x) + g(x) = 5x^2 + 4x$ şeklindedir.

$f(x)g(x) = 10x^3 + 9x^2 + 14x + 6$ dır. $mod\ 7$ ye göre işlem yapılırsa elde edilen sonuç $f(x)g(x) = 3x^3 + 2x^2 + 6$ şeklindedir.

4. KLASİK VE KUANTUM BİLGİSAYARLAR

Bu bölümde klasik ve kuantum bilgisayarların gelişim süreçlerinden bahsedilip özellikleri ele alınacaktır.

4.1 Klasik Bilgisayarlar

Bilgisayar, çok sayıda aritmetiksel veya mantıksal işlemlerden oluşan bir işi kısa sürede yapıp, yaptığı işlemden elde ettiği sonucu saklayabilen ve istendiğinde geri getiren elektronik bir araçtır. Bilgisayarların dilleri 0 ve 1 rakamlarından oluşmaktadır. Bunlara "bit" adı verilmektedir. Bit 0 veya 1 değerini alabilir, ikisine aynı anda sahip olamaz. Kullanıcısını iletmek istediği şeyleri kendi dillerinde algılarlar. Örneğin klavye tuşlarına basarak "Matematik" kelimesini yazıldığında bilgisayar bu kelimeyi 0 ve 1 olarak algılar. Her harfin karşılığına denk gelen 1 ve 0 lardan oluşan sayılar bütünü olarak algılanır. Bu harflere karşılık gelen sayılar ASCII karakterler olarak tanımlanmaktadır (Şekil 4.1). 0 ve 1 ikili sayı sisteminin elemanlarıdır. Bilgisayar bilimlerinin temeli olan bu sistemle tüm sayılar 0 ve 1 sayıları kullanılarak ifade edilirler. Bu sistemde birşeyin varlığı 1, yokluğu 0 şeklinde ifade edilir. Bu iki sayı sayesinde ekranda yazıları okuyabiliyor, resimleri görüntüleyebiliyor ve videolar izlenebiliyor.

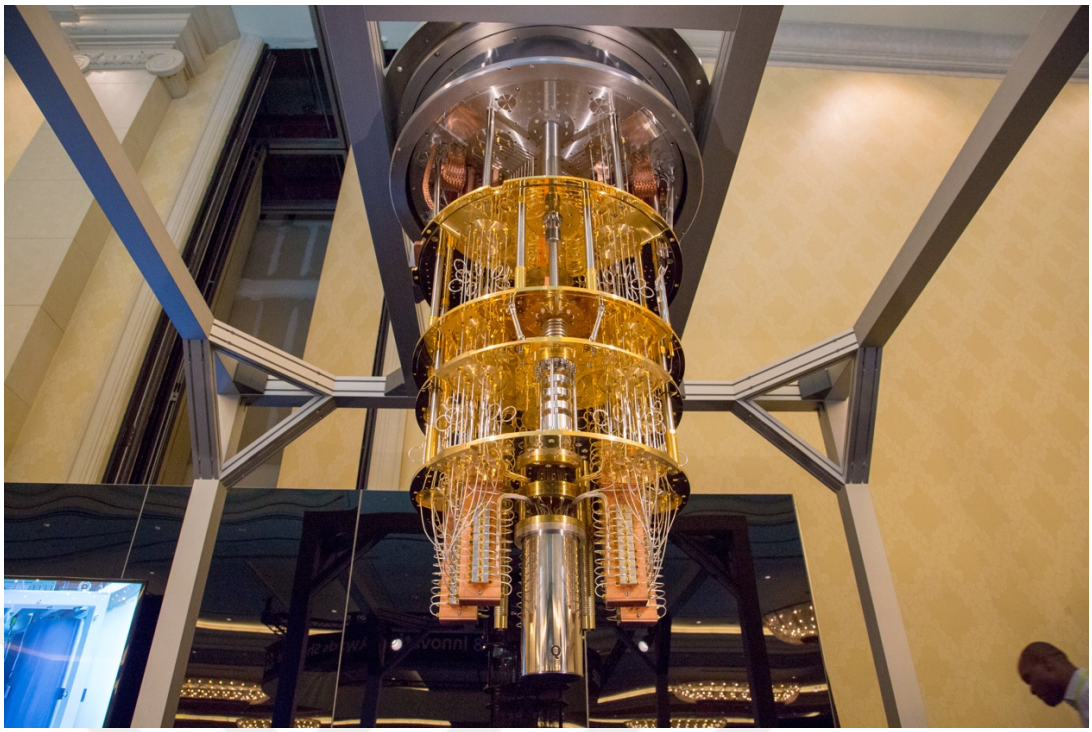
4.2 Kuantum Bilgiayarlar

Son yıllarda adını sıklıkla duyduğumuz kuantum bilgisayarlar, kuantum fiziğinin mekaniklerine göre çalışan bilgisayarlardır. Klasik bilgisayarlar geleneksel fizik kurallarına göre çalışırken kuantum bilgisayarlar kuantum fiziğini kullanmaktadırlar. Fikir ilk olarak 1959 yılında bilim insanı Amerikalı Richard Feynman, kuantum mekaniklerinin bilgisayarlar üzerinde kullanılabileceğini ve böylece çok yüksek hesaplama kapasitesine sahip bilgisayarlar yapılabileceğini öngördü. Klasik bilgisayarlarda bit fonksiyonları kullanılırken kuantum bilgisayarlarda bitin yerini kubitler almaktadır. Kubit fonksiyonu, 0 veya 1 olmak zorunda değildir. Aynı anda hem 0 hem de 1 olabilir. Buna süper pozisyon denmektedir. Normal bilgisayarlara

göre çok daha yüksek miktarda veri taşıma kapasitesine sahiplerdir. Kuantum bilgisayarların çalışma prensibini basitçe bir örnekle açıklamak istersek; klasik bilgisayar işlemcisini 1000 odalı bir plaza olarak düşünelim ve çalışan bir kişi çantasını bu odalardan birinde kaybettiğini varsayalım. Çalışanın çantasını bulması için 1000 odayı tek tek dolaşması gerekir. Fakat kuantum bilgisayarlarda, bilgisayar, çalışanın 1000 kopyasını oluşturuyor ve böylece çanta daha kısa sürede bulunuyor. Klasik bilgisayarlarda, veri depolamak için transistörlere ihtiyaç duyulurken, kuantum bilgisayarlarda ise parçacığın hareket edebilmesi gerektiğinden dolayı transistörlere ihtiyaç duyulmamakdatır. Fakat en ufak bozucu hareket sonucun hatalı olmasına neden olabilmektedir. Araştırmacılar parçacığın sabit durduğu süreyi uzatmaya çalışmaktadır. IBM'in 50 kubit gücündeki kuantum bilgisayarı Şekil 4.2'de yer almaktadır.

0	<NUL>	32	<SPC>	64	@	96	`	128	Ä	160	†	192	¿	224	‡
1	<SOH>	33	!	65	A	97	a	129	Å	161	°	193	¡	225	·
2	<STX>	34	"	66	B	98	b	130	Ç	162	¢	194	¬	226	‚
3	<ETX>	35	#	67	C	99	c	131	É	163	£	195	√	227	„
4	<EOT>	36	$	68	D	100	d	132	Ñ	164	§	196	ƒ	228	‰
5	<ENQ>	37	%	69	E	101	e	133	Ö	165	•	197	≈	229	Â
6	<ACK>	38	&	70	F	102	f	134	Ü	166	¶	198	Δ	230	Ê
7	<BEL>	39	'	71	G	103	g	135	á	167	ß	199	«	231	Á
8	<BS>	40	(	72	H	104	h	136	à	168	®	200	»	232	Ë
9	<TAB>	41	)	73	I	105	i	137	â	169	©	201	…	233	È
10	<LF>	42	*	74	J	106	j	138	ä	170	™	202		234	Í
11	<VT>	43	+	75	K	107	k	139	ã	171	´	203	À	235	Î
12	<FF>	44	,	76	L	108	l	140	å	172	¨	204	Ã	236	Ï
13	<CR>	45	-	77	M	109	m	141	ç	173	≠	205	Õ	237	Ì
14	<SO>	46	.	78	N	110	n	142	é	174	Æ	206	Œ	238	Ó
15	<SI>	47	/	79	O	111	o	143	è	175	Ø	207	œ	239	Ô
16	<DLE>	48	0	80	P	112	p	144	ê	176	∞	208	–	240	
17	<DC1>	49	1	81	Q	113	q	145	ë	177	±	209	—	241	Ò
18	<DC2>	50	2	82	R	114	r	146	í	178	≤	210	"	242	Ú
19	<DC3>	51	3	83	S	115	s	147	ì	179	≥	211	″	243	Û
20	<DC4>	52	4	84	T	116	t	148	î	180	¥	212	'	244	Ù
21	<NAK>	53	5	85	U	117	u	149	ï	181	µ	213	′	245	ı
22	<SYN	54	6	86	V	118	v	150	ñ	182	∂	214	÷	246	ˆ
23	<ETB>	55	7	87	W	119	w	151	ó	183	∑	215	◊	247	˜
24	<CAN>	56	8	88	X	120	x	152	ò	184	∏	216	ÿ	248	¯
25	<EM>	57	9	89	Y	121	y	153	ô	185	π	217	Ÿ	249	˘
26	<SUB>	58	:	90	Z	122	z	154	ö	186	∫	218	⁄	250	˙
27	<ESC>	59	;	91	[	123	{	155	õ	187	ª	219	€	251	°
28	<FS>	60	<	92	\	124	|	156	ú	188	º	220	‹	252	¸
29	<GS>	61	=	93	]	125	}	157	ù	189	Ω	221	›	253	˝
30	<RS>	62	>	94	^	126	~	158	û	190	æ	222	fi	254	˛
31	<US>	63	?	95	_	127	<DEL>	159	ü	191	ø	223	fl	255	ˇ

Şekil 4.1 : ASCII tablosu.



Şekil 4.2 : Kuantum bilgisayar.

5. KUANTUM ANAHTAR DAĞITIMI VE PROTOKOLLERİ

Bu bölümde kuantum anahtar dağıtımının işleyişi, ilk kuantum anahtar dağıtım protokolü BB84 ve Cascade hata düzeltme protokolünden bahsedilmiştir.

5.1 Kuantum Anahtar Dağıtımı

Kuantum anahtar dağıtımı, klasik şifrelemenin kullandığı matematiksel olarak karmaşık hesaplamlara dayanmak yerine anahtar dağılımında gizliliği korumak için kuantum fiziği yasalarını kullanmaktadır. Bu yasalar evrensel doğa kanunları olduğu için doğruluğundan şüphe edilmemektedir. Klasik sistemlerde 0 veya 1 durumu kuantum bilgisayarlarda 0 ve 1 şeklindedir, yani hem 0 hem 1 olabilir. Bu durum super pozisyon şeklinde adlandırılmaktadır.

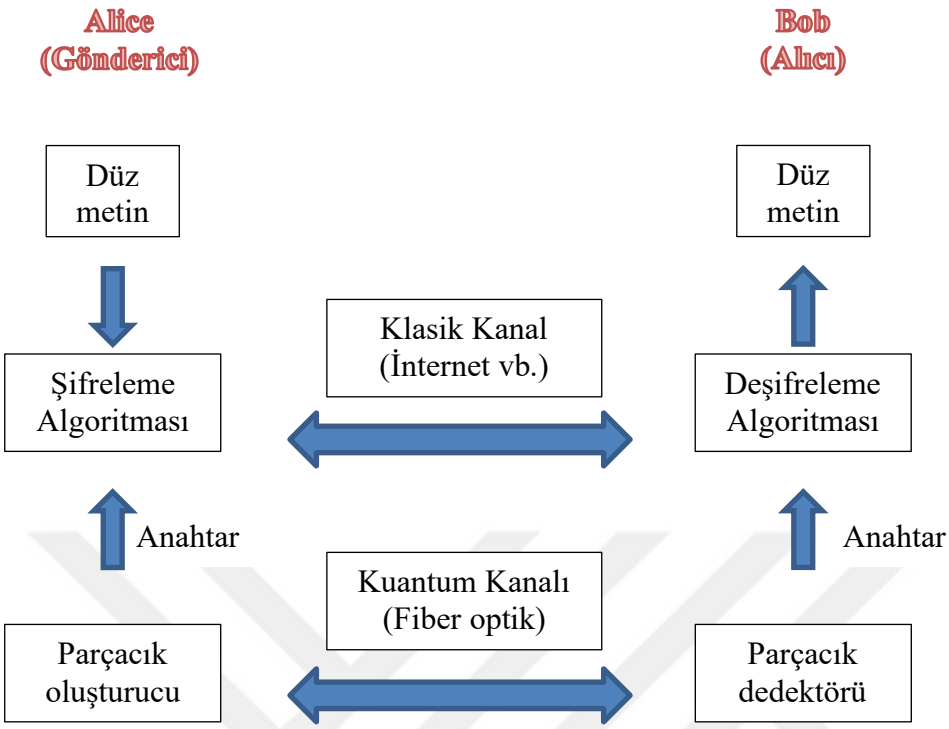
Şekil 5.1'de de görüldüğü gibi haberleşmede iki ayrı kanal kullanılmaktadır. Kuantum anahtar dağıtımında iletişim kuantum kanalında başlayıp klasik kanalda sona ermektedir. Kuantum kanalı mesaj iletimi için kullanılmamaktadır. Bu kanalda fotonların iletilmesiyle anahtar paylaşımı gerçekleşmektedir. Diğer tüm adımlar klasik kanallarda yapılmaktadır [8].

Kuantum kriptografi Heisenberg'in belirsizlik ilkesine dayandığından dolayı kuantum mekaniğinin temel ögesi foton'un aynı anda iki özelliği (konum ve momentum) ölçülemez ve bu özelliklerden biri için sırayla yapılan ilk ölçüm ikinci ölçümün sonucunu belirsiz hale getirmektedir. Bu ilke ile fotonlar kullanılarak optik kanal üzerinden iletişim gerçekleştirildiğinden dolayı fotonun polarizasyonuna bağlı olarak kübitlerin arka arkaya gerçekleşirilen ölçümler ile bozulacağı açıklanmaktadır. Böylelikle iletişim sırasında gerçekleştirilen bir müdahale kolayca açığa çıkmaktadır [16]. Kuantum kriptografi bu özelliği kullanarak anahtar iletimi için güvenli bir kanal oluşturur [9]

5.2 Kuantum Anahtar Dağıtım Protokolleri

Bu kısımda kuantum anahtar dağıtımında kullanılan ilk protokol BB84 ve hata

düzeltme protokolü olan Cascade protokollerinden bahsedilecektir.



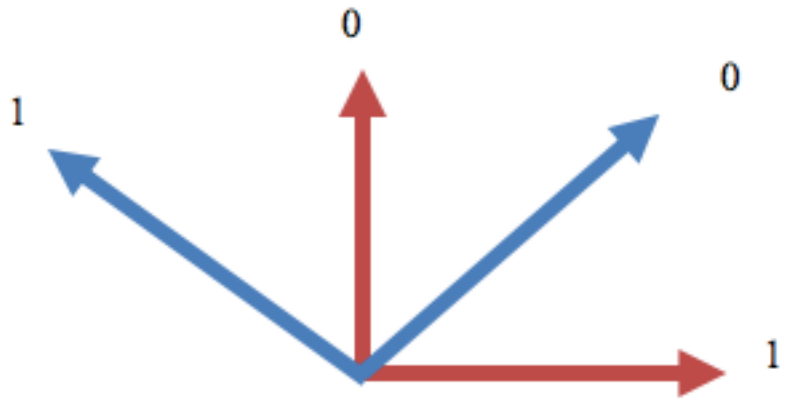
Şekil 5.1 : Kuantum kriptografi.

5.2.1 BB84 protokolü

İlk anahtar dağıtım protokolü olan BB84, 1984 yılında Charles Bennet ve Gilles Brassard tarafından geliştirilen yöntemdir.

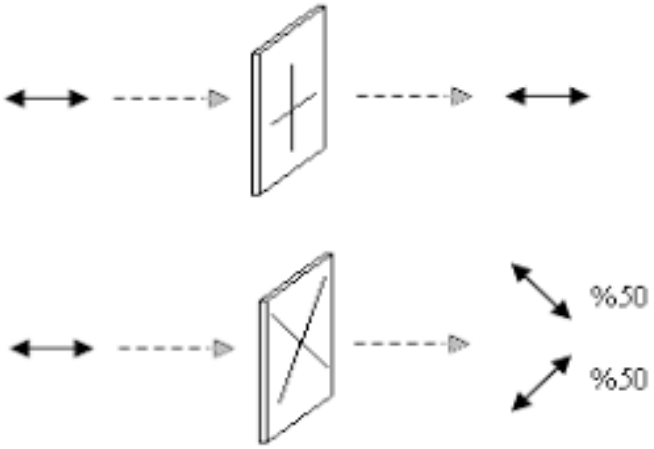
Anahtar iletiminda kullanılan fotonların dört değişik açısı bulunmaktadır. Bunlar yatay (-) (horizontal), dikey (|) (vertical), sağ diyagonal (/) (right diagonal) ve sol diyagonal (\) (left diagonal) şeklindedir. Şekil 5.2'de görüldüğü gibi polarizsyonlar 0 ve 1 olmak üzere farklı iki kubit değerini ifade etmektedirler. 45° ve 90° değerlerine ait fotonlar 0 ile, 0° ve 135° değerlerine ait fotonlar 1 ile işlem yapmaktadırlar.

Fotonların durumunu belirlemek için iki farklı filtre kullanımaktadır. Bunlar, yatay ve dikey polarizasyonlu fotonları belirlemek için kullanılan + şeklindeki filtre ve 45° ve -45°'lik fotonları belirlemek için kullanılan X şeklindeki filtrelerdir. Şekil 5.3'te, her bir fotonun ölçülecek yöntemi (+, X) rastgele olarak seçilmektedir. Eğer uygun bi ölçüm gerçekleştirilirse foton bu ölçümden değişmeden çıkar ve sonuç doğru olur, fakat ölçüm yöntemi gelen fotona karşı uygun olmazsa bu durumda foton değişir ve sonuç yanlış olarak alıcıya iletilir [8,9,10]. Şekil 5.3'te görüldüğü üzere ilk adımda yatay (-) foton + şeklindeki filtreden geçtiğinde herhangi bir bozulma olmayacaktır.



Şekil 5.2 : Polarizasyon durumu.

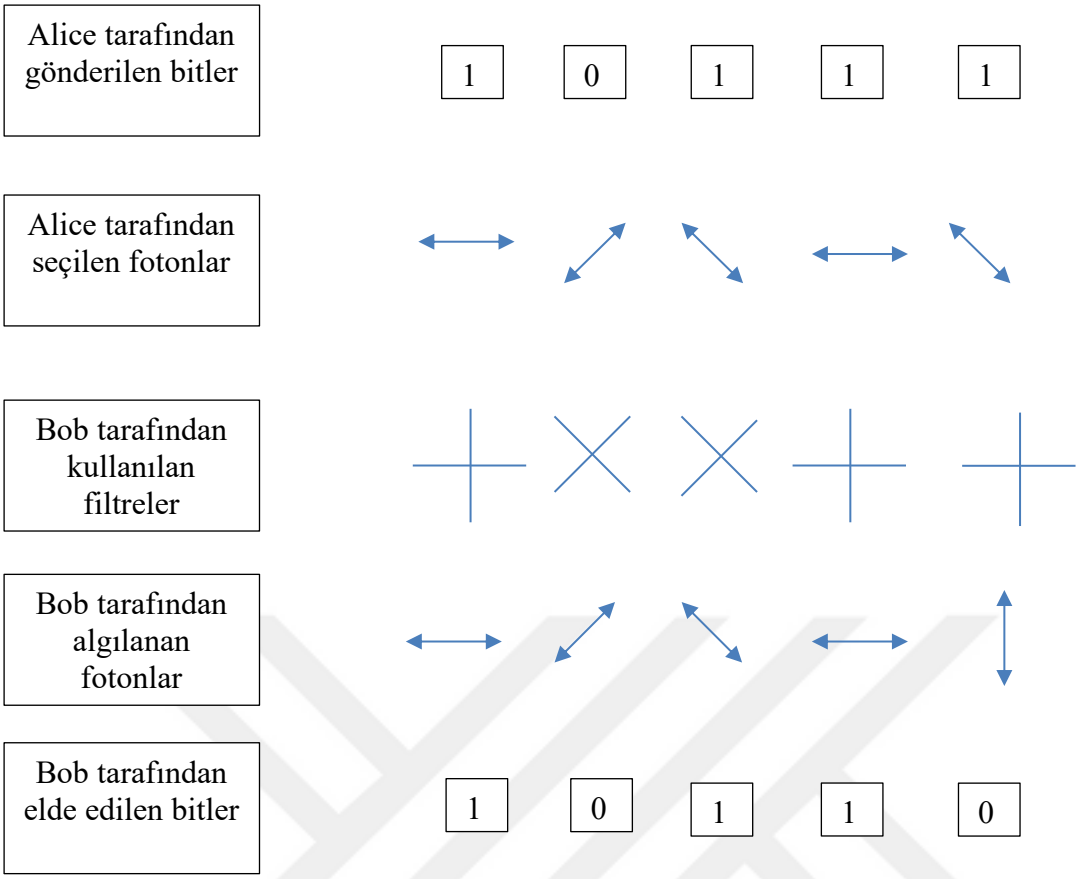
Fakat ikinci durumda yine aynı foton X şeklindeki filtreden geçtiğinde aynı şekilde kalamayıp %50 olasılıkla / şekline ya da yine %50 olasılıkla \ şekline dönüşecektir.



Şekil 5.3 : Filtre kullanımı.

Bahsedilenleri bir örnekle açıklamak gerekirse Şekil 5.4'te yer alan bilgiler incelendiğinde Alice tarafından gönderiler bitler 10111 iken Bob'un elde ettiği bit dizisi 10110 şeklindedir. Bob, Alice ile açık kanallardan iletişime geçerek 1., 2., 3., 4. adımlarda kullandığı filtrelerin doğru olduğunu tespit etmiştir. Bu durumda yeni anahtar 10111 şeklinde olacaktır. Araya herhangi bir dinleyici girmezse Alice ve Bob aynı anahtara sahip olabilirler.

Şimdi Alice ve Bob arasındaki iletişimi dinleyenin Eve olduğunu varsayalım. Eve'in yapabileceği şey Alice ve Bob'un yapıtığı gibi filtre kullanmasıdır ve Alice'in Bob'a göndermiş olduğu filtrelerin bilgilerini dinlemektir.



Şekil 5.4 : Örnek anahtar dağıtımı.

Alice tarafından gönderilen fotonları Eve elde eder ve bir bit dizisi belirler [10]. Kopyalanamazlık teoremi ve Heisenberg belirsizlik ilkesi gereğince Eve fotonları kopyalayamayacağı için tekrardan oluşturur ve Bob'a bu fotonları iletir. Alice ve Bob tekrardan iletişime geçtiğinde anahtarlarının farklı oldukları belirlenerek iletişim sırasında araya bir dinleyicinin girdiği ortaya çıkartılmış olur.

5.2.2 Cascade hata düzeltme protokolü

Cascade protokolü ilk olarak Kuantum Anahtar Dağıtımı sistemleri üzerinde kullanılmak üzere oluşturulmuş hata sezme ve düzeltme yöntemidir. Protokol, gönderici ve alıcı arasında oluşturmak istenen ortak anahtarda aynı olmayan bitler için düzeltme ve hatayı tespit etmeyi sağlamaktadır. Ana fikir, mesajı bloklar haline dönüştürülerek bu bloklar için pariti (eşlik) değerleri eşleştirilir ve böylece hata tespit edilerek düzeltme işlemi yapılır [12,13]. Bu algoritmada karıştırma, hata sezme iteraktif hata düzeltme şeklinde adımlardan oluşmaktadır.

Protokol şu şekilde çalışmaktadır:

- Alice ve Bob, N uzunluklu bit dizilerine sahiptir.

- Orijinal dizi Alice'tekidir. Bunu A ile gösterelim.
- Bob'un dizisi (B), Alice'in dizisinin en çok $p = \%15$ kadar bozulmuş halidir.
- Alice ve Bob açık kanallardan iletişime geçerek karıştırıcı bir fonksiyon üzerinde anlaşırlar ve kendi bitlerini bu aynı karıştıma fonksiyonundan geçirirler.
- Karıştırma fonksiyonundan geçirdikten sonra elde edilen bit dizileri bloklara bölüp her bloğa ait parity değerlerini hesaplarlar. Alice, bloklara ait tüm parity değerlerini Bob'a iletir.
- Blokların paritilerinde uzlaşamadıklarındaki blok ikiye bölünür ve yarılarının paritileri hesaplanıp karşılaştırılır.
- Paritilerin uyuşmadığı yarı, hata bulunana ve düzeltilene kadar tekrar ikiye bölünür ve karşılaştırılır.

Bu iletişimler, herkese açık olan klasik kanal üzerinden gerçekleştirilmektedir. Bu algoritmayı bir örnekle açıklamak gerekirse:

Alice ilk olarak Bob'a Şekil 5.5'teki gibi anahtar yollamış olsun ve kuantum kanalından geçerek Bob'a ulaşan anahtar Şekil 5.6'teki gibi olsun. Burada Bob'un anahtarında 8. bitte hata olduğu görülmektedir. Daha sonra Alice ve Bob herkese açık iletişim kaynaklarından haberleşerek bir karıştırıcı fonskiyon üzerinde anlaşırlar ve Bob anahtar bitlerinin yerlerini rastgele değiştirir. Anahtar bitlerinin değişmiş hali Şekil 5.7'da yer almaktadır.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	1	1	0	0	1	0	1	1	0	0	1

Şekil 5.5 : Alice'in anahtarı.

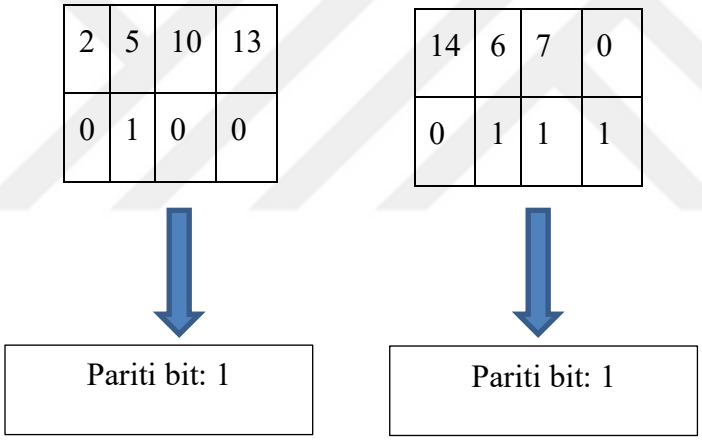
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	1	1	1	0	1	0	1	1	0	0	1

Şekil 5.6 : Bob'un anahtarı.

2	5	10	13	14	6	7	0	1	4	8	3	9	11	12	15
0	1	0	0	0	1	1	1	1	0	0	0	1	1	1	1

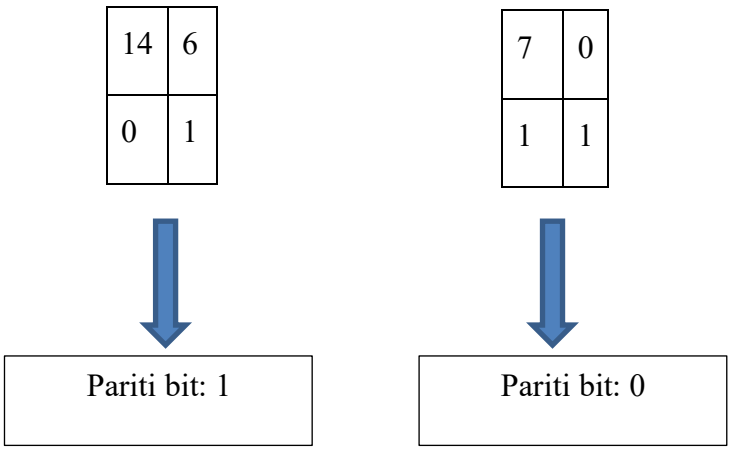
Şekil 5.7 : Karıştırma fonksiyonundan sonra elde edilen anahtar.

Hataları düzeltmek için öncelikle 16 bitlik anahtar 8'er bit olmak üzere iki kısıma ayrılır. İlk 8 bitlik kısım için pariti hesaplanır. Pariti bit sayısı, bitlerin ikilik tabandaki değerine göre ifade edilir, bit değerlerinin toplamı tek sayıda ise "1", çift sayıda ise "0" şeklinde ifade edilir. Şekil 5.7'de yer alan bit dizisinin (01000111) paritisi "0" dır. Diğer kısım için, Alice tüm bit dizisinin pariti sayısını Bob ile paylaştığından Bob diğer 8 bitlik kısmın paritisini doğrulamış olur. Bob bunu Alice ile paylaşır ve Alice kendi anahtarındaki 8 bitin paritisi ile karşılaştırır. Karşılaştırma sonucunda Alice'in paritisinin "0" olduğu anlaşılır. Bu adımdan sonra Şekil 5.8'de de görüldüğü gibi ilk 8 bitlik kısımı eşit parçaya bölerek karşılaştırma işlemine devam edilir.



Şekil 5.8 : 8 bitlik kısmı ayırma işlemi.

Şekil 5.8'de Bob'un anahtarının 4 bitlik kısımlar için paritinin "1" geldiği görülmüş olup Alice aynı sıradaki bit değerleri için pariti kontrolü yaptığında ikinci 4 bitlik kısım için "0" elde eder. Artık bir sonraki adım Şekil 5.9'da yer alan 4 bitlik diziyi ikiye ayırmak olacaktır. Bir önceki adımda yapılan pariti kontrol işlemleri tekrarlandığında bu kez ikinci 2 bitlik kısım için Bob'un paritisi "0" iken Alice'in "1" gelmektedir. Son adım olarak son 2 bitlik kısım ikiye ayrılır ve artık hangi bitte hata olduğu bulunur. Elde edilen sonuçlar neticesinde artık Bob yanlış olan 7. bitini düzelterek yeni anahtarı elde etmiş olur ve böylece Alice ile Bob ortak anahtarı elde etmiş olurlar. Fakat Cascade protokolü fazlasıyla interaktif bir protokol olduğu için protokol bu anlamda olumsuz etkilenebilir.



Şekil 5.9 : 4 bitlik kısmı ayırma işlemi.

Yani yoğun olarak bir haberleşme gerçekleştirildiği için protokolün hızı ve verimliliği bu adımda azalmaktadır. Ayrıca iletişim herkese açık kanaldan gerçekleştirildiği için araya giren kişinin mesajları değiştiremeyeceği fakat paylaşılan bilgilere ulaşabilir olduğu bilinmektedir.

Tüm bu bilgiler göze alındığında, bu çalışmada açıklığın giderilmesi gerektiği ve daha kısıtlı bilgiler paylaşılarak anahtar paylaşımı ve kontrolü yapılabilmesi amaçlanmıştır. Bir sonraki bölümde polinom interpolasyonu kullanılarak anahtarda kaç tane hatalı bit olduğu ve hatalı bitlerin yerleri tespit edilmiştir.

6. KUANTUM BİLGİSAYARLARDA ANAHTAR DAĞITIM PROTOKOLÜ VE İMPLENTASYON

Polinom interpolasyonu hesaplamalı bilimler ve kriptografide sıklıkla kullanılan bir yaklaşım metodudur. Bu bölümde polinom interpolasyonu kullanılarak anahtar dağıtım işleminin güvenli hale gelmesi sağlanmıştır. İlk olarak Newton polinom interpolasyonu ele alınmış daha sonra interpolasyon kullanılarak anahtar dağıtım protokolünün nasıl gerçekleştirileceğine değinilmiştir.

6.1 Newton Polinom İnterpolasyonu

Polinom interpolasyonu $r+1$ tane nokta yardımı ile r. dereceden bir polinomun belirlenmesidir. Yani, $(x_0, y_0), \dots, (x_r, y_r)$ nokta çiftleri için $p_r(x_i) = y_i \ (0 \le i \le r)$ olacak şekilde derecesi en fazla r olan tek bir p_r polinomunun var olmasıdır. Bu polinomu belirlendikten sonra diğer ara değerler elde edilen polinom yardımı ile hesaplanabilir. Polinom interpolasyon yöntemleri içinde en kullanışlı olanlardan biri Newton yöntemidir.

Bu yöntem ile, $(x_0, y_0), \dots, (x_r, y_r)$ noktaları olsun, r dereceli bir $p_r(x)$ polinomu denklem 6.1 deki gibi oluşturulabilir ve $p_i(x_i) = y_i \ (i = 1,2, \dots, r.)$ kullanılarak sırasıyla c_i katsayıları elde edilir. Böylelikle tüm x ve y çiftlerinden geçen bir polinom elde edilir.

$$p_r(x) = c_0 + \sum_{i=1}^{r} c_i \prod_{j=0}^{i-1} (x - x_j) \tag{6.1}$$

Polinom interpoalsyonunda dikkat edilmesi gereken nokta, $p_r(x)$ polinomunu oluşturmak için, en az $r+1$ tane noktanın bilinmesi gereklidir. Eğer $r+1$ den az nokta biliniyorsa $p_r(x)$ polinomu elde edilemez [11,20].

6.2 Anahtar Dağıtım Protokolü

Bu bölümde amacımız güvenli bir kanal oluşturarak alıcı ve gönderici arasında ortak

anahtar oluşturan algoritmayı anlatmaktır. Newton polinom interpolasyon yöntemi güvenli haberleşme amacıyla çeşitli yerlerde kullanılmıştır. Bu tezde, Newton interpolasyon polinomu kullanılarak anahtar paylamışını gerçekleştiren algoritma detaylı olarak incelenmiş ve güvenlik analizi yapılmıştır. Bunların yanında [11] geliştirilen algoritma gerçeklenmiştir. İlk aşamada algoritmanın hataları nasıl tespit ettiği anlatılacaktır.

Bu protokolde;

- İlk olarak rastgele n tane x ve y çiftleri seçilir.
- Seçilen x ve y değerleri alıcıya gönderilir.
- y değerleri ve polinomun derecesi Alice tarafından Bob'a açık kanallar üzerinden iletilir.
- x değerleri anahtarın değerleridir ve Bob'a quantum kanalından iletilir.
- Alıcı tarafından, x ve y değelerini kullanılarak Newton interpolasyon yöntemi ile en fazla $n-3$. derece bir polinom elde edilir. Eğer polinomun derecesi $n-1$ olsaydı, gönderilen tüm noktalar doğru olacaktı ve böylece hatalı bitin olup olmayacağı anlaşılamayacaktı. Eğer polinomun derecesi $n-2$ olsaydı kullanılan noktalar dışında bir adet doğru nokta gönderilmediği için hesaplanamayacaktı.
- n tane x değeri bu polinomda yazılarak y' değerleri hesaplanır.
- y' değerleri ile 2. adımda alınan y değerleri karşılaştırılır, yanlış bitler bulunur ve doğruluk değeri hesaplanır.
- Karşılaştırma sonucunda kullanıcı tarafıdan girilen doğruluk değeri, hesaplanan doğruluk değerinden büyükse algortima durdurulur.
- Aksi durumda yeni x değerleri kullanılarak polinom elde edilip doğruluk değerine tekrar bakılır ve doğruluk değeri istenilen doğruluk değerinden büyük olana kadar işleme devam edilir.

6.3 Güvenlik Analizi

Kriptografide güvenlik analizi güvenlik seviyesi ile ölçülmektedir. Güvenlik seviyesi genellikle bitler üzerinden açıklanmaktadır. n-bitlik güvenlik seviyesinin anlamı, kaba

kuvvet ve herhangi bir saldırı metodu kullanarak atak yapan kişinin anahtarı elde etmek için 2^n sayıda işlem yapmasıdır. Örneğin, AES-128, 128-bit güvenlik seviyesine sahiptir. Klasik bilgisayarlarda, kaba kuvvet saldırısı ile anahtar boyutuna göre anahtarı elde etme süreleri Çizelge 6.1'de verilmiştir.

Çizelge 6.1 : Klasik bilgisayarda anahtarı elde etme süreleri.

Anahtar Boyutu	Anahtarı Elde Etme Süresi
56-bit	399 saniye
128-bit	$1.02\ x\ 10^{18}$ yıl
192-bit	$1.872\ x\ 10^{37}$ yıl
256-bit	$3.31\ x\ 10^{56}$ yıl

Klasik bilgisayarda 10000 yıl süren bazı işlemler, D-Wave 2X adlı kuantum bilgisayarında 1 saniye sürdüğü Google yöneticisi Harmut Neven tarafından açıklanmıştır [18]. Buna örnek olarak, Çizelge 6.1' de yer alan 128-bitlik anahtar boyutu için anahtarı elde etme süresini kuantum bilgiayarlar için hesaplanırsa;

$1.02\ x\ 10^{18}$ yıl $x\ \frac{1\ \text{saniye}}{10000\ \text{yıl}} = 1.02\ x\ 10^{14}$saniye.

$1.02\ x\ 10^{14}$saniye $x\ \frac{1\ \text{dakika}}{60\ \text{saniye}} = 17\ x\ 10^{11}$ dakika.

$17\ x\ 10^{11}$ dakika $x\ \frac{1\ \text{saat}}{60\ \text{dakika}} = 28\ x\ 10^{9}$ saat.

$28\ x\ 10^{9}$ saat $x\ \frac{1\ gün}{24\ saat} = 1.17\ x\ 10^{9}$ gün.

$1.17\ x\ 10^{9}\ gün\ x\ \frac{1\ yıl}{365\ gün} = 3\ x\ 10^{6}$ yıl.

elde edilir.

Yukarıdaki işlem doğrultusunda, Çizelge 6.1 deki değerlerin kuantum bilgisayarlar için güncellenmiş helleri Çizelge 6.2 de sunulmuştur.

Elde edilen değerler sonucunda 128-bitlik anahtar değerinin kaba kuvvet saldırısı ile ele geçirilmesi olanaksız olarak gözükmektedir. Bundan dolayı Çizelge 6.1'de

bahsedilen protokolde güvenlik seviyesi 128-bit olarak kabul edilir. Buna bağlı olarak seçilmesi gereken nokta sayısı şu şekilde belirlenmelidir;

Çizelge 6.2 : Kuantum bilgisayarda anahtarı elde etme süreleri.

Anahtar Boyutu	Anahtarı Elde Etme Süresi
56-bit	$4\ x\ 10^{-17}$ saniye
128-bit	$3\ x\ 10^{6}$ yıl
192-bit	$5.94\ x\ 10^{25}$yıl
256-bit	$1.1\ x\ 10^{45}$ yıl

Blok sayısı t bit olmak üzere, Alice tarafından oluşturulan polinomun derecesi d olsun. Polinomun derecesi ve kullanılan y değerleri açık kanaldan gönderilir ve araya giren kişi Eve'in polinomu elde edebilmesi için $d + 1$ nokta kullanması yeterlidir. Fakat doğruluğundan emin olunması için en az $d + 2$ noktaya ihtiyacı vardır.

Örneğin x_1, x_2, x_3, x_4, x_5 değerleri Alice tarafından Bob'a gönderilecek olsun. Alice tarafından x_1, x_2 ve x_3 kullanılarak ikinci derece polinom türetilirse, x_4 ve x_5 değerlerinden birinin doğru gelmesi gerekmektedir. Doğru gelmezse x_4 ve x_5 değerleri, ilk üç değerden üretilen polinom üzerinde olmaz. Yani bu durumda x_1, x_2 ve x_4 kullanılarak üretilen polinomda da aynı sonuç elde edilir.

Yukarıdaki örnekte ele alındığı gibi ikinci ($d.$) derece bir polinom üretilirken üç nokta kullanılmıştır. Fakat doğrulama için bir nokta daha kullanılması gerektiğinden dolayı en az dört $(d + 2)$ noktaya ihtiyaç vardır.

Eve'in y değerlerini kullanarak anahtar değeri olan x değerlerini elde edebilmesi için kaba-kuvvet saldırısını kullanması gerekmektedir.

Çizelge 6.3'te görüldüğü gibi herbir x değerinin blok uzunluğu t olmak üzere, $d + 2$ nokta kullanılacağından dolayı tüm olasılık $2^{(d+2)t}$ şeklindedir.

Yukarıda bahsedilen güvenlik seviyesinin 128-bit olmasından dolayı $2^{(d+2)t} \geq 2^{128}$ şeklinde olmalıdır. Buda $(d + 2)t \geq 128$ olmasını gerektirmektedir.

Bunu bir örnekle ele alacak olursak;

Çizelge 6.3 : x ve y değerlerinin bloklara ayrılmış hali.

t-bit	t-bit	t-bit	t-bit	t-bit	t-bit	t-bit
x_1	x_2	x_3	.	.	.	x_{d+2}
y_1	y_2	y_3	.	.	.	y_{d+2}

x değerlerinin blok uzunlukları 8-bit olsun. $(d+2)8 \geq 128$ ve buradan $d = 14$ elde edilir. Yani blok uzunluğu 8 olan x değerleri için derecesi 14 olan bir polinom kullanılabilir.

6.4 Protokolün İmplementasyonu

Algoritmanın implementasyonu Şekil 6.1, Şekil 6.2, Şekil 6.3, Şekil 6.4, Şekil 6.5 ve Şekil 6.6'da detaylı olarak anlatılmıştır.

```
#include <stdio.h>
#include <gmp.h>
#include<stdlib.h>
#include<math.h>
#include<time.h>
#include<stdint.h>
#define MAX 4000

int64_t k,n,i,j,m;
int64_t n;
int64_t kullanilan_nokta_sayisi;
int64_t data2[MAX];
int64_t step_number=0;
int64_t selected_x[MAX],selected_y[MAX];
int dogru_sayisi=0;
int desired_percent=0;
int calculated_percent=0;

mpz_t mod_n;
mpz_t dummy1;
mpz_t dummy2;
mpz_t dummy3;
mpz_t dummy4,sum,u;
mpz_t resultTemp1;
mpz_t resultTemp2;
mpz_t *gmp_data1;
mpz_t *gmp_data2;
mpz_t *estimated_y;
mpz_t *input_x,*input_y;
mpz_t *c_list;
mpz_t** fx;
```

Şekil 6.1 : Değişken ve kütüphanelerin tanımlanması.

```
void compare_results(){
    dogru_sayisi=0;
    for(i=0;i<n;i++){
        mpz_mod(resultTemp1,input_y[i],mod_n);
        if(!mpz_cmp(estimated_y[i], resultTemp1)){
            dogru_sayisi++;
        }
        else{
            printf("\n %lld. deger hatalıdır",i+1);
        }
    }
}
```

Şekil 6.2 : Algoritmadan elde edilen ve açık kanaldan alınan y değerlerinin karşılaştırılması.

```
void find_result(){

    for(int k=0;k<n;k++){
        mpz_add(sum, dummy1 , fx[1][1]);
        for(i=2;i<=n;i++)
        {
            mpz_set_str(u,"1",10);

            for(j=1;j<i;j++){
                mpz_sub(resultTemp1,input_x[k],gmp_data1[j]);
                mpz_mul(u,u,resultTemp1);
            }
            mpz_mul(u,u,fx[1][i]);
            mpz_add(sum,sum,u);
            mpz_mod(sum,sum,mod_n);
        }
        mpz_add(estimated_y[k],dummy1,sum);
    }
}
```

Şekil 6.3 : Tahmin edilen n. derece denklemde x değerleri koyularak y değerlerinin hesaplanması.

```
void find_coef()
{
    mpz_add(c_list[0], dummy1 , fx[1][1]);

    for(i=2;i<=kullanilan_nokta_sayisi;i++)
    {
        for(j=1;j<=kullanilan_nokta_sayisi-i+1;j++)
        {
            mpz_sub(resultTemp1,gmp_data1[j+i-1],gmp_data1[j]);
            mpz_invert(resultTemp1,resultTemp1,mod_n);
            mpz_sub(resultTemp2,fx[j+1][i-1],fx[j][i-1]);
            mpz_mul(resultTemp1,resultTemp1,resultTemp2);
            mpz_mod(resultTemp1,resultTemp1,mod_n);
            mpz_add(fx[j][i],dummy1,resultTemp1);
        }
        mpz_add(c_list[i-1],dummy1,fx[1][i]);
    }

    printf("\nNewton Divided Tablosu");
    printf("\n--------------------------------\n");
    printf("X F(x) ");
    for(i=1;i<kullanilan_nokta_sayisi;i++)
        printf("y%lld ",i);
    printf("\n--------------------------------\n");
    for(i=1;i<=kullanilan_nokta_sayisi;i++)
    {
        gmp_printf("%Zd\t ",gmp_data1[i]);
        for(j=1;j<=kullanilan_nokta_sayisi-i+1;j++)
            gmp_printf("%Zd ",fx[i][j]);
        printf("\n");
    }
    printf("\n--------------------------------\n");
    printf("\nKatsayilar\n");
    for(i=0;i<kullanilan_nokta_sayisi;i++){
        gmp_printf(" c[%d] = %Zd ... ",i,fx[1][i+1]);
    }
    printf("\nx degerleri\n");
    for(i=0;i<kullanilan_nokta_sayisi;i++){
        gmp_printf(" x[%d] = %Zd ... ",i,gmp_data1[i+1]);
    }

}
```

Şekil 6.4 : Newton bölünmüş farklar tablosunun oluşturulması.

```
void subset(int64_t arr[], int64_t data[], int64_t start, int64_t
end, int64_t index, int64_t r)
{
    int64_t i,j;

    if (index == r) {
        printf("\n----------------------------------------------
-----------------\n");

        for (j = 0; j < r; j++)
        {
            gmp_printf("\n x = %Zd , y = %Zd ", gmp_data1[j+1],
gmp_data2[j+1]);
            mpz_add(fx[j+1][1], dummy1 , gmp_data2[j+1]);
        }

        find_coef();
        find_result();
        compare_results();

        calculated_percent = dogru_sayisi*100/n;

        if(calculated_percent>=desired_percent){
            printf("\n ***** Islem Sonlandirildi ***** \n");
            return;
        }
        else
            printf("\n ***** SOnuc Bulunamadı ***** \n");
        printf("\n");
        return;
    }

    for (i = start; i <= end && end - i + 1 >= r - index; i++)
    {
        if(calculated_percent>=desired_percent){
            return;
        }

        mpz_add(gmp_data1[index+1], dummy1 , input_x[i]);
        mpz_add(gmp_data2[index+1], dummy1 , input_y[i]);

        subset(arr, data, i+1, end, index+1, r);
    }
}

void printsubset(int64_t x[], int64_t n, int64_t
kullanilan_nokta_sayisi)
{
    int64_t data[kullanilan_nokta_sayisi];
    subset(x, data, 0, n-1, 0, kullanilan_nokta_sayisi);
}
```

Şekil 6.5 : x ve y değerlerinin (n+1)'li kombinasyonlarının hesaplanması ve algoritmanın test edilmesi.

```
int main()
{
    fx = (mpz_t**)malloc(sizeof(mpz_t*) * 20);
    for(int i = 0; i < 20; i++)
    {
        fx[i] = (mpz_t*)malloc(sizeof(mpz_t) * 20);
        for(int j = 0; j < 20; j++)
            mpz_init(fx[i][j]);
    }

    input_x= malloc(100000000000 * sizeof(mpz_t));
    input_y = malloc(100000000000 * sizeof(mpz_t));

    gmp_data1= malloc(100000000000 * sizeof(mpz_t));
    gmp_data2 = malloc(100000000000 * sizeof(mpz_t));
    estimated_y = malloc(100000000000 * sizeof(mpz_t));
    c_list = malloc(100000000000 * sizeof(mpz_t));;

    mpz_t a,b,c;
    mpz_t d,e,f;

mpz_inits(a,b,c,d,e,f,mod_n,dummy1,dummy2,dummy3,dummy4,resultTem
p1,resultTemp2,sum,u);

    printf("\n ******  Basladi  ******");

    printf("\nHangi modda çalışılacak : ");
    gmp_scanf("%Zd",mod_n);

    printf("\nNokta sayisini girin : ");
    scanf("%lld",&n);

    printf("\nKullanılacak nokta sayisini girin : ");
    scanf("%lld",&kullanilan_nokta_sayisi);

    printf("\nYuzde kac dogru olacak : ");
    scanf("%d",&desired_percent);

    printf("\nX and f(x) degelerini sırasıyla giriniz . . . \n");

    for(i=0;i<n;i++)
    {
        printf("\nx ve y = ");
        gmp_scanf("%Zd %Zd",input_x[i],input_y[i]);
    }

    mpz_init_set_str(fx[1][2],"2017",10);
    printsubset(x,n,kullanilan_nokta_sayisi);
    return 0;
}
```

Şekil 6.6 : Algoritma girdilerinin alınması ve algoritmanın başlatılması.

7. SONUÇ VE ÖNERİLER

Bu çalışmada, Newton polinom interpolasyonu kullanılarak oluşturulan algoritma ile kuantum kanallarından karşı tarafa iletilen anahtar bilgisinin eksiksiz ve doğru bir şekilde olması amaçlanmıştır. Kuantum kanalarının fiziksel özelliklerinden dolayı anahtarın iletimi sırasında, anahtar değişime uğrayabilir ya da kanalı dinleyen bir kişi anahtar ile ilgili bilgileri elde edebilir.

Algortima kısaca şu şekilde çalışmaktadır; Alice oluşturduğu random x ve y çiftlerinden x değerleri anahtar bilgisi olup kuantum kanalından iletilirken y değerleri ve oluşturulan polinomun derecesini açık kanallardan Bob'a iletir. Bob 6.2 de anlatılan protokolün esaslarına göre bir polinom üretir ve üretilen polinom Alice tarafından üretilen polinom ile karşılaştırılır. Karşılaştırma sonunda eğer hatalı bit oluşmuşsa bu bit tespit edilir.

Algoritmada, araya giren kişi Eve'in anahtara dair elde ettiği bilgiler sayesinde anahtarı tam olarak elde etmesi imkansızdır. Bunun gerçeklenmesi için Eve'in kaba-kuvvet saldırısı gerçekleştirmesi gerekmektedir. Gerekli önlemler alındığında, bu kuantum bilgisayarlar için bile neredeyse imkansız gözükmektedir.

Gelecekteki çalışmalarda, anahtar paylaşımı gerçekleştirilirken açık kanaldan iletilen bilgiler daha kısıtlı hale getirilmeye çalışılacaktır ve polinom interpolasyonu dışında farklı matematiksel yöntemler geliştirilemeye çalışılacaktır.

KAYNAKLAR

[1] **Yeşilbaş, E.** (2016). Cebirsel Kriptoloji Yöntemleri ve Bazı Uygulamaları, *Yüksek Lisans Tezi*, Recep Tayyip Erdoğan Üniversitesi.

[2] **Gupta, A., Mittal R.** (2014). Symetric and Assymetric Cryptosystem, *International Journal of Engineering & Technology*.

[3] **Hoffstein, J., Silverman, J. H., & Hoffstein, J.** (2008). In *An Introduction to Mathematical Cryptography*, 65-67.

[4] **Geary A.,** (2009). Analysis of A Man-in-the-Middle Attack on the Diffie-Helman Key Exchange Protocol, Naval Postgraduate School, California.

[5] **Yerlikaya T., Buluş E., Buluş N. Araz, T.** (2006). Asimetrik Şifreleme Algoritmalarında Anahtar Değişim Sistemleri.

[6] **Diffie, W., & Hellman, M.** (1976). New directions in cryptography. IEEE transactions on Information Theory, 644-654.

[7] **Forney, G.,** (2007). Channel Coding: Techniques, Analysis and Design Principles Lecture Notes (Chapter-7). Massachusetts Instute of Technology.

[8] **Calver, T. I.** (2011). An empirical analysis of the cascade secret key reconciliation protocol for quantum key distribution.

[9] **Kara, H.** (2013). Kuantum Şifreleme, *Yüksek Lisans Tezi*, Yıldız Teknik Üniversitesi.

[10] **Türker, İ.** (2018). BB84 Yöntemi Kullanılarak Kuantum Şifreleme Benzetim Ortamı Geliştirilmesi, Yüksek Lisans Tezi, Harran Üniversitesi.

[11] **Kurt, G. K., Özdemir, E., Özkirisci, N. A., & Topal, O. A.** (2019). A Key Verification Protocol for Quantum Key Distribution. *IEEE Access*.

[12] **Toyran, M., Pedersen, T. B., Hasekioğlu, A. A., Can, M. A., & Berber, S.** (2013, April). A study on CASCADE error correction protocol. In 2013 21st Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

[13] **Toyran, M., Pedersen, T. B., Hasekioğlu, A. A., Can, M. A., & Berber, S.** (2012, April). Comparison of CASCADE error correction protocol and LDPC error correction codes. In 2012 20th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

[14] **Fischer, M. J.** (2017). CPSC 467: Cryptography and Computer Security Lecture 2.

[15] **Christensen, C.** (2010). Northern Kentucky University HNR304 Lecture Notes: Caesar Ciphers.

[16] **Gümüş, E.** (2011). Kuantum Kriptografi ve Anahtar Dağıtım Protokolleri.

[17] **Akyıldız, E., Doğanaksoy, A., Keyman, E., & Uğuz, M.** (2004). Odtü Uygulamalı Matematik Enstitüsü Uygulamalı Matematik Enstitiüsü Kriptolojiye Giriş Ders Notları. (2004)

[18] **Url-1**, *https://www.techtimes.com/articles/114614/20151209/googles-d-wave-2x-quantum-computer-100-million-times-faster-than-regular-computer-chip.htm.,* erişim tarihi 05.10.2019.

[19] **Paar, C., Pelzl, J.** (2009). Understanding cryptography: a textbook for student and practitioners, Springer Science & Business Media.

[20] **Cheney, W., Kincaid, D.** (2008). Numerical mathematics and computing. Thompson Learning, Inc., Belmont.

ÖZGEÇMİŞ

Ad-Soyad **:** Berrak UZUN

Doğum Tarihi ve Yeri **:** 15/09/1993 Zonguldak

E-posta **:** berrakaytas@gmail.com

ÖĞRENİM DURUMU:

- **Lisans:** 2016, Bülent Ecevit Üniversitesi, Fen Edebiyat Fakültesi, Matematik.

MESLEKİ DENEYİM VE ÖDÜLLER:

- Matematik Bölüm Birinciliği, 2016, (Lisans).