TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BCH, QC-MDPC, GABIDULIN KODLARININ KRİPTOGRAFİK UYGULAMALARI VE BAZI KOD TABANLI KUANTUM SONRASI ALGORİTMALARIN PERFORMANS ANALİZLERİ

YÜKSEK LİSANS TEZİ

Burcu Ecem YILMAZ

Matematik Anabilim Dalı

Tez Danışmanı: Doç. Dr. Zülfükar SAYGI

AĞUSTOS 2019

Fen Bilimleri Enstitüsü Onayı

.....
Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

.....
Prof. Dr. Oktay DUMAN
Anabilimdalı Başkanı

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 162111008 numaralı Yüksek Lisans öğrencisi **Burcu Ecem YILMAZ**'ın ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"BCH, QC-MDPC, GABIDULIN KODLARININ KRİPTOGRAFİK UYGULAMALARI VE BAZI KOD TABANLI KUANTUM SONRASI ALGORİTMALARIN PERFORMANS ANALİZLERİ"** başlıklı tezi **01.08.2019** tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı:	**Doç. Dr. Zülfükar SAYGI** TOBB Ekonomi ve Teknoloji Üniversitesi	.....
Jüri Üyeleri:	**Prof. Dr. Hüseyin MERDAN (Başkan)** TOBB Ekonomi ve Teknoloji Üniversitesi	.....
	Doç. Dr. Oğuz YAYLA Hacettepe Üniversitesi	.....

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Burcu Ecem Yılmaz

ÖZET

Yüksek Lisans Tezi

BCH, QC-MDPC, GABIDULIN KODLARININ KRİPTOGRAFİK UYGULAMALARI VE BAZI KOD TABANLI KUANTUM SONRASI ALGORİTMALARIN PERFORMANS ANALİZLERİ

Burcu Ecem Yılmaz

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Matematik Anabilim Dalı

Tez Danışmanı: Doç. Dr. Zülfükar Saygı

Tarih: Ağustos 2019

Bu tezde, NIST'in düzenlediği Kuantum Sonrası Kriptografi Standartlaştırma çağrısı kapsamında aday gösterilen kod tabanlı kuantum sonrası algoritmalardan bazıları incelenmiş ve 128-bit, 192-bit, 256-bit güvenlik seviyeleri için belirli bilgisayarlarda çalıştırılarak performansları ölçülmüştür. Bu kapsamda öne çıkan HQC ve BIKE algoritmaları incelenmiştir. Bunun yanı sıra HQC ile temel özellikleri benzer olan RQC aday algoritması da ele alınmıştır. Bu amaçla algoritmalarda kullanılan BCH, QC-MDPC ve Gabidulin kod aileleri incelenmiş ve örneklendirilmiştir. Ayrıca, BCH kod ailesini kullanan HQC algoritmasının çalışma adımları örnek üzerinde gerçeklenmiştir.

Anahtar Kelimeler: Kuantum sonrası kriptografi, Kod tabanlı, Anahtar kapsülleme mekanizması, Şifreleme, Şifre çözme, Kodlama, Kod çözme, BCH, QC-MDPC, Gabidulin.

ABSTRACT

Master of Science

CRYPTOGRAPHIC APPLICATIONS OF BCH, QC-MDPC, GABIDULIN CODES AND PERFORMANCE ANALYSES OF SOME CODE-BASED POST-QUANTUM ALGORITHMS

Burcu Ecem Yılmaz

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Mathematics

Supervisor: Assoc. Prof. Dr. Zülfükar Saygı

Date: August 2019

In this thesis, some of the code-based algorithms nominated within the scope of NIST Post-Quantum Cryptography Standardization call are examined and their performance is measured by running on specific computers for 128-bits, 192-bits and 256-bits security levels. In this context, prominent HQC and BIKE algorithms were examined. Besides, the RQC candidate algorithm, which has similar characteristics with HQC, is also considered. For this purpose, BCH, QC-MDPC and Gabidulin code families used in algorithms were examined and exemplified. Further, the operation steps of the HQC algorithm using the BCH code family are implemented on the sample.

Keywords: Post-quantum cryptography, Code-based, Key encapsulation mechanism, Encryption, Decryption, Coding, Encoding, BCH, QC-MDPC, Gabidulin.

TEŞEKKÜR

Yüksek lisans eğitimim boyunca beni bilgileriyle yönlendiren ve destekleyen değerli danışman hocam Doç. Dr. Zülfükar SAYGI'ya; kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Matematik Bölümü öğretim üyelerine; değerli jüri üyeleri Prof. Dr. Hüseyin MERDAN'a, Doç. Dr. Oğuz YAYLA'ya teşekkürlerimi sunarım. Maddi manevi destekleri ile her zaman yanımda olup beni her kararımda destekleyen en başta annem olmak üzere aileme ve eğitim hayatım boyunca beni yalnız bırakmayan çalışma arkadaşlarım Sevde KARA ve Kübra Asena GELİŞLİ'ye çok teşekkür ederim. Son olarak yüksek lisans eğitimimde sağladığı burstan dolayı TOBB Ekonomi ve Teknoloji Üniversitesine teşekkürlerimi sunarım.

İÇİNDEKİLER

ÇİZELGE LİSTESİ

KISALTMALAR

BCH : Bose-Chaudri-Hocquenghem
BIKE : Bit Flipping Key Encapsulation
GJS : Guo, Johansson and Stankovski
HQC : Hamming Quasi-Cyclic
IND-CCA : INDistinguishability under Chosen Ciphertext Attack
IND-CPA : INDistinguishability under Chosen Plaintext Attack
MDPC : Modarate Density Parity-Check
NIST : National Institute of Standards and Technology
QC : Quasi-Cyclic
RQC : Rank Quasi-Cyclic

SEMBOL LİSTESİ

Bu tezde kullanılan simgeler açıklamaları ile birlikte aşağıda yer almaktadır.

Simgeler	**Açıklama**
$\mathbb{Z}$	Tam sayılar kümesi
$\mathbb{F}$	Sonlu cisim
$\mathbb{F}^*$	$\mathbb{F}\backslash\{0\}$
$\mathbb{F}_q$	q elemanlı sonlu cisim
$\mathbb{F}_q[x]$	Katsayıları $\mathbb{F}_q$ da olan polinomlar halkası
$\mathbb{F}_q^n$	$\mathbb{F}_q$ cismi üzerinde uzunluğu n olan vektörlerin kümesi
$\mathbb{F}_q^{n\times k}$	$\mathbb{F}_q$ cismi üzerinde $n \times k$ boyutundaki matrislerin kümesi
$\mathbb{F}_{q^m}$	q^m elemanlı sonlu cisim
$\mathcal{C}$	Kod kelimeleri kümesi
$\mathrm{wt}(\cdot)$	Bir vektörün Hamming ağırlığı
$\|\|\cdot\|\|$	Bir vektörün rank ağırlığı
$\approx$	Yaklaşık olarak eşit
$\star$	Vektörleri terim terime çarpma işlemi
$u \xleftarrow{\$} U$	U kümesinden rastgele u değişkeninin düzgün seçilmesi

1. GİRİŞ

Günümüzde kullanılan açık anahtarlı şifreleme, imzalama ve anahtar değişimi algoritmaları mevcut bilgisayarlarla polinom zamanda kırılamamaktadır. Bu açık anahtarlı kripto sistemlerden en çok bilinenler güvenilirlikleri çarpanlara ayırma problemine dayalı RSA, ayrık logaritma problemine dayalı Diffie-Hellman anahtar değişimi ve imzalama algoritması DSA, eliptik eğri ayrık logaritma problemine dayalı ECC algoritmalarıdır. Ancak yeterince güçlü kuantum bilgisayarlar ortaya çıktığında günümüzde kullanılan açık anahtarlı kripto sistemlerin dayandığı problemler Shor algoritması [23] ile polinom zamanda kırılabileceğinden kuantum saldırılara karşı zayıf hale geleceklerdir. Mevcut şifreleme ve anahtar değişimi algoritmalarının kırılması ise günümüz internet ve dijital haberleşme platformlarının güvenilirliğini ve bütünlüğünü tehlikeye atacaktır.

Kuantum sonrası kriptografi, kuantum ve klasik bilgisayarlarla yapılan ataklara karşı güvenli olan kriptografik sistemlerin geliştirilmesini amaçlamaktadır. Kuantum bilgisayarlar üzerine yapılan çalışmaların son yıllarda artmasıyla NIST'in düzenlediği Kuantum Sonrası Kriptografi Standartlaştırma çağrısı [26] kapsamında toplam 69 adet olmak üzere kafes tabanlı, kod tabanlı, özet fonksiyonları tabanlı, çok değişkenli polinom tabanlı ve diğer bazı özel algoritmalar standartlaşma süreci için aday gösterilmiş ve NIST'in sayfasında ilan edilmiştir [27]. NIST'in çağrısına aday olarak sunulan kod tabanlı algoritmaların isimleri ve türleri (imzalama, anahtar kapsülleme, şifreleme) Çizelge 1.1'de verilmiştir. 30 Ocak 2019'da henüz kırılamamış ve gerekli koşulları sağlayan ikinci tur aday algoritmalar açıklanmıştır. Bunların çoğu yeniden düzenlenerek, gerekli ekleme-çıkarmalar yapılarak ve hatta bazıları birleştirilerek ikinci turda [28] yer almışlardır. İlk turda 69 olan algoritma sayısı Çizelge 1.2'de görülebileceği gibi ikinci turda 26'ya düşmüştür.

Eksilen 43 algoritma arasında süreçten çekilen, kriptografik olarak zafiyetleri ortaya çıkarılan ve benzer özellikler taşımasından dolayı birleşerek sürece devam eden algoritmalar mevcuttur.

Çizelge 1.1: 1. tur kod tabanlı aday algoritmalar (* Süreçten çekilen algoritmalar)

İmzalama	**Anahtar Kapsülleme**	**Şifreleme**
pqsigRM RaCoSS RankSign*	BIG QUAKE BIKE Klasik McEliece DAGS Edon-K* HQC LAKE LEDA-KEM NTS-KEM LOCKER Ouroboros-R QC-MDPC KEM Ramstake RLCE-KEM RQC	LEDApkc McNie

Çizelge 1.2: 2. tur aday algoritmalar

	İmzalama	**Anahtar Kapsülleme**	**Şifreleme**
Kod Tabanlı		BIKE Klasik McEliece HQC ROLLO LEDAcrypt NTS-KEM RQC	LEDAcrypt
Kafes Tabanlı	CRYSTALS-DILITHIUM FALCON qTESLA	CRYSTALS-KYBER FrodoKEM Round5 LAC NewHope NTRU NTRU Prime SABER Three Bears	LAC NTRU Round5
Çok Değişkenli	GeMSS LUOV MQDSS Rainbow		
Diğer	SPHINCS+ Picnic	SIKE	

Bu algoritmaların günlük hayatımızdaki güvenlik uygulamalarının içerisinde kullanılmaları planlandığı için klasik bilgisayarlardaki çalışma zamanları da önem

arzetmektedir. Dolayısıyla bu çalışmada, süreçte aday gösterilen kod tabanlı algoritmalardan bazıları güvenlik seviyelerine bağlı olarak farklı platformlarda çalıştırılmış ve performansları karşılaştırılmıştır. Performans analizleri sonucunda öne çıkan algoritmaların kullandığı sistemler ve kod aileleri incelenmiştir. Bu incelemeler sonucunda öne çıkan HQC, BIKE ve RQC, HQC ile benzer sisteme sahip, kuantum sonrası algoritmaları ayrıntılarıyla beraber verilmiştir. Üç algoritmanın kullandığı kod aileleri sırasıyla BCH, QC-MDPC ve Gabidulin şeklindedir, bu kod aileleri ile ilgili bilgiler Bölüm 2'de detaylı bir şekilde yer almıştır. Algoritmaların detaylarının anlaşılması açısından tezin içeriğinde öncelikle Bölüm 2'de kodlama teorisi bilgileri verilmiş, Bölüm 3, 4 ve 5'te ise BCH, Gabidulin ve QC-MDPC kod ailelerini kullanan kuantum sonrası algoritmalar detaylandırılmıştır. Son olarak Bölüm 6'da performans analiz sonuçları verilmiştir. Tez boyunca vektörler kalın, küçük harflerle ve $\mathbb{F}_q$ üzerinde $(abcd)$ formunda, $\mathbb{F}_{q^m}$ üzerinde (a,b,c,d) formunda ifade edilecektir. Matrisler ise büyük harflerle gösterilecektir.

1.1 Genel Tanımlar

Bu bölümde tez boyunca kullanılacak olan sonlu cisimlerle ilgili tanımlara ve örneklere yer verilmiştir. Daha ayrıntılı bilgiler için [13], [15] ve [24] kaynakları incelenebilir.

Tanım 1.1 (İndirgenemez Polinom)**.** *Eğer $p(x) \in \mathbb{F}[x]$ sabit olmayan bir polinom ve $b(x), c(x) \in \mathbb{F}[x]$ olmak üzere $p(x) = b(x)c(x)$ iken $b(x)$ ya da $c(x)$ polinomlarından biri sabit polinom olmak zorunda ise $p(x)$ polinomuna $\mathbb{F}[x]$ üzerinde indirgenemez denir.*

Tanım 1.2 (Primitif Eleman)**.** *α, $\mathbb{F}_q$ sonlu cisminin bir elemanı olsun. Eğer*

$$\{\alpha^i | i \geq 0\} = \mathbb{F}_q^*$$

ise α elemanına $\mathbb{F}_q$ sonlu cisminin primitif elemanı veya $\mathbb{F}_q^*$ ın üreteci denir.

Örnek 1. $\mathbb{F}_2$ üzerinde indirgenemez bir polinom olan $f(x) = 1 + x^2 + x^3$ kullanılarak $\mathbb{F}_{2^3}$ inşa edildiğinde $f(\alpha) = 0$ olmak üzere cismin tüm elemanları,

$$0, 1, \alpha, 1+\alpha, \alpha^2, 1+\alpha^2, \alpha+\alpha^2, 1+\alpha+\alpha^2$$

şeklinde yazılabilir. α nın bir primitif eleman olup olmadığını kontrol etmek için kuvvetleri alınırsa $1+\alpha^2+\alpha^3=0$ olduğundan,

$$\begin{aligned} \alpha^0 &= 1 & \alpha^4 &= \alpha+\alpha^3 = 1+\alpha+\alpha^2 \\ \alpha^1 &= \alpha & \alpha^5 &= \alpha+\alpha^2+\alpha^3 = 1+\alpha \\ \alpha^2 &= \alpha^2 & \alpha^6 &= \alpha+\alpha^2 \\ \alpha^3 &= 1+\alpha^2 \end{aligned}$$

elemanları elde edilir. α elemanı $\mathbb{F}_{2^3}^*$ çarpımsal grubunun tüm elemanları ürettiği için bir primitif elemandır.

Tanım 1.3 (Minimal Polinom)**.** *$\alpha \in \mathbb{F}_{q^m}$ bir eleman olsun. $\mathbb{F}_q[x]$ üzerinde $f(\alpha)=0$ olmak üzere derecesi en küçük sıfırdan farklı monik $f(x)$ polinomuna α nın $\mathbb{F}_q$ üzerindeki minimal polinomu denir.*

Tanım 1.4 (Eşlenik)**.** *$\alpha \in \mathbb{F}_{q^m}$ için $\alpha^{q^t}=\alpha$ olacak şekilde en küçük pozitif tam sayı t olsun. Bu durumda,*

$$C(\alpha) = \left\{\alpha, \alpha^q, \alpha^{q^2}, \alpha^{q^3}, \ldots, \alpha^{q^{t-1}}\right\}$$

kümesinin elemanlarına α nın $\mathbb{F}_q$ üzerindeki eşlenikleri denir.

Teorem 1.1. *$\mathbb{F}_{q^m}$ üzerinde bir β elemanının minimal polinomu $m_\beta(x)$ olsun. β elemanının eşlenik elemanlarından oluşan küme $C(\beta)$ olmak üzere,*

$$m_\beta(x) = \prod_{a \in C(\beta)} (x-a) \tag{1.1}$$

olur.

Eşlenik köklerin her birinin minimal polinomlarının aynı olduğu bu teoremden tespit edilebilir. Aynı zamanda teorem sayesinde kolaylıkla hesaplanabilen minimal polinomlar, BCH kodunun üreteç polinomunu elde etmekte kullanılabilir.

Açıklama 1.1. *Eğer $p(x) \in \mathbb{F}_q[x]$ olmak üzere $p(x) \mid x^n - 1$ olacak şekilde en küçük n tam sayısı $n = q^m - 1$ ise $\mathbb{F}_q$ üzerinde m dereceli indirgenemez $p(x)$ polinomuna primitif polinom denir ve her bir kökü $\mathbb{F}_{q^m}$ cisminin bir primitif elemanı olur.*

Kod tabanlı algoritmalarda çoğu zaman $p = 2$ seçilerek $\mathbb{F}_2$ de işlem yapılır. Çizelge 1.3'te $\mathbb{F}_2$ üzerinde bazı primitif polinomların listesi verilmiştir.

Çizelge 1.3: $\mathbb{F}_2$ üzerinde derecesi m olan bazı primitif polinomlar

m	Polinom	m	Polinom
3	$1+x+x^3$	10	$1+x^3+x^{10}$
4	$1+x+x^4$	11	$1+x^2+x^{11}$
5	$1+x^2+x^5$	12	$1+x+x^4+x^6+x^{12}$
6	$1+x+x^6$	13	$1+x+x^3+x^4+x^{13}$
7	$1+x^3+x^7$	14	$1+x+x^6+x^{10}+x^{14}$
8	$1+x^2+x^3+x^4+x^8$	15	$1+x+x^{15}$
9	$1+x^4+x^9$		

Örnek 2. $\mathbb{F}_2$ üzerinde $p(x) = 1+x+x^3$ primitif polinomu verilsin. α bu polinomun kökü olmak üzere $\mathbb{F}_{2^3}$ cisminin elemanları Çizelge 1.4'teki gibidir.

Çizelge 1.4: $p(x) = 1+x+x^3$ tarafından üretilen $\mathbb{F}_{2^3}$ cisminin elemanları

Kuvvet Gösterimi	Polinom Gösterimi	3-lü Gösterimi
0	0	(0 0 0)
1	1	(1 0 0)
α	α	(0 1 0)
α^2	α^2	(0 0 1)
α^3	$1+\alpha$	(1 1 0)
α^4	$\alpha+\alpha^2$	(0 1 1)
α^5	$1+\alpha+\alpha^2$	(1 1 1)
α^6	$1+\alpha^2$	(1 0 1)

Tanım 1.4 gereğince 0 ve 1 elemanlarının her kuvveti yine kendisini verdiğinden eşlenikleri 0 ve 1 bulunur. α elemanının eşlenikleri,

$$C(\alpha) = \left\{\alpha, \alpha^2, \alpha^{2^2} = \alpha^4\right\}$$

kümesinin elemanlarıdır ve α^2, α^4 elemanlarının eşleniklerinin kümesi de aynı

kümedir. α^3 elemanının eşlenikleri,

$$C(\alpha^3) = \left\{\alpha^3, (\alpha^3)^2 = \alpha^6, (\alpha^3)^{2^2} = \alpha^{12} = \alpha^5\right\}$$

kümesinin elemanlarından oluşur. Benzer şekilde α^5, α^6 elemanlarının eşlenikleri aynı kümenin elemanlarından oluşur.

Teorem 1.1 kullanılarak her bir elemanın minimal polinomu bulunabilir. 0 elemanının minimal polinomu,

$$m_0(x) = \prod_{a \in C(0)} (x-a) = (x-0) = x$$

olur. 1 elemanının minimal polinomu,

$$m_1(x) = \prod_{a \in C(1)} (x-a) = (x-1) = 1+x$$

bulunur. α elemanının minimal polinomu ise,

$$\begin{aligned} m_\alpha(x) = \prod_{a \in C(\alpha)} (x-a) &= (x-\alpha)(x-\alpha^2)(x-\alpha^4) \\ &= (x^2+\alpha^4 x+\alpha^3)(x+\alpha^4) \\ &= 1+x+x^3 \end{aligned}$$

olur. Eşlenik elemanların minimal polinomları eşit olduğundan α^2 ve α^4 elemanlarının minimal polinomları $m_\alpha(x)$ polinomuna eşittir. α^3 elemanının minimal polinomu,

$$\begin{aligned} m_{\alpha^3}(x) = \prod_{a \in C(\alpha^3)} (x-a) &= (x-\alpha^3)(x-\alpha^5)(x-\alpha^6) \\ &= (x^2+\alpha^2 x+\alpha)(x+\alpha^6) \\ &= 1+x^2+x^3 \end{aligned}$$

olarak bulunur ve $\alpha^3, \alpha^5, \alpha^6$ elemanları eşlenik olduğundan minimal polinomları da aynıdır.

2. LİNEER KODLAR

Bu bölümde lineer kodlar ve bazı özel kod aileleri hakkında bilgiler ve örnekleri bulunmaktadır. Daha ayrıntılı bilgi için [13], [15] ve [24] kaynakları incelenebilir. Lineer kodların cebirsel yapısı, etkili kodlama ve kod çözme algoritmalarının inşa edilebilmesi için alt yapı sağlar.

Tanım 2.1 (Lineer Kod)**.** *$\mathbb{F}_q$ üzerindeki n boyutlu vektör uzayı $\mathbb{F}_q^n$ nin k boyutlu bir alt uzayına, uzunluğu n ve boyutu k olan $\mathcal{C}$ lineer kodu denir ve $[n,k]$ şeklinde gösterilir. $\mathcal{C}$ nin eleman sayısı $|\mathcal{C}| = q^k$ dır. $\mathcal{C}$ nin elemanları kod kelimeleri şeklinde ifade edilir.*

Kod kelimeleri vektörlerle ifade edilecektir, örnegin n uzunlukta bir $\mathbf{c}$ kod kelimesi $c = (c_0 c_1 \dots c_{n-1})$ şeklindedir. Lineer kodlar k uzunluktaki bir mesajı alarak n uzunluktaki bir kod kelimesine çevirirler. Devirli kodlar ve BCH kodlar lineer kod ailelerine örnek olarak verilebilir.

Örnek 3. Aşağıdaki kodlar lineer kodlara örnektir;

(i) $\mathcal{C} = \{(00000), (00010), (01000), (01010)\} \subseteq \mathbb{F}_2^5$

(ii) $\mathcal{C} = \{(0000), (0001), (0002), (1100), (2200), (1102), (2201), (2202)\} \subseteq \mathbb{F}_3^4$

(iii) $\mathcal{C} = \{(0000000), (1111111)\} \subseteq \mathbb{F}_2^7$.

Tanım 2.2 (Vektörün Hamming Ağırlığı)**.** $\mathbf{x} = (x_0 x_1 \dots x_{n-1}) \in \mathbb{F}_q^n$ *bir vektör olsun.* $\mathbf{x}$ *vektörünün sıfırdan farklı bileşenlerinin sayısına bu vektörün Hamming ağırlığı denir ve* $wt(\mathbf{x})$ *ile gösterilir.*

Tanım 2.3 (Kodun Hamming Ağırlığı)**.** *Bir $\mathcal{C}$-$[n,k]$ kodunun Hamming ağırlığı,*

$$wt(\mathcal{C}) = \min\{wt(\mathbf{x}) : \mathbf{x} \in \mathcal{C}, \mathbf{x} \neq \mathbf{0}\}$$

şeklinde tanımlanır.

Tanım 2.4 (Vektörün Hamming Uzaklığı)**.** $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ *iki kod kelimesi olsun. Bu iki kod kelimesinin Hamming uzaklığı* $d(\mathbf{x}, \mathbf{y})$ *birbirlerinden farklı koordinatlarının sayısıdır.*

Örnek 4. $\mathbf{x} = (02110), \mathbf{y} = (21010)$, $\mathbb{F}_3$ cismi üzerinde iki kod kelimesi olsun. İlk 3 koordinatları birbirinden farklı olduğundan Hamming uzaklıkları $d(\mathbf{x}, \mathbf{y}) = 3$ bulunur.

Sonuç 2.1. $d(\mathbf{x}, \mathbf{y}) = wt(\mathbf{x} - \mathbf{y})$ *olduğu görülebilir.*

Tanım 2.5 (Minimum Uzaklık)**.** *Bir* $\mathcal{C}$-$[n,k]$ *kodunun minimum uzaklığı,*

$$d_{min} = d(\mathcal{C}) = \min \{d(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in \mathcal{C}, \mathbf{x} \neq \mathbf{y}\}$$

şeklinde tanımlanır. Başka bir ifadeyle,

$$d_{min} = \min_{\mathbf{x},\mathbf{y} \in \mathcal{C}, \mathbf{x} \neq \mathbf{y}} wt(\mathbf{x} - \mathbf{y}).$$

Bu kısımdan itibaren tez boyunca Hamming ağırlık ifadesi yerine ağırlık, Hamming uzaklık yerine uzaklık kullanılacaktır.

Teorem 2.1. *Bir* $\mathcal{C}$-$[n,k]$ *kodunun uzaklığı* d *olsun. Bu durumda,*

$$d = wt(\mathcal{C})$$

olur.

Minimum uzaklığı d olan bir kod $d-1$ tane hata tespit etme ve $\delta = \left\lfloor \frac{d-1}{2} \right\rfloor$ ye kadar hata düzeltme kapasitesine sahiptir.

Tanım 2.6 (Üreteç Matris)**.** $\mathbb{F}_q$ *üzerinde bir* $\mathcal{C}$-$[n,k]$ *kodu verilsin. Satırları* $\mathcal{C}$ *kodunun bazını oluşturan* $k \times n$ *boyutundaki* G *matrisine bu kodun üreteç matrisi denir.*

Üreteç matrisi $G \in \mathbb{F}_q^{k \times n}$ olan $\mathcal{C}$-$[n,k]$ kodunun elemanları,

$$\mathcal{C} = \left\{ \mathbf{m}G \mid \mathbf{m} \in \mathbb{F}_q^k \right\}$$

şeklinde ifade edilir.

Tanım 2.7 (Dual Kod)**.** *$\mathbb{F}_q$ üzerinde bir $\mathcal{C}$-$[n,k]$ kodu verilsin. $\mathcal{C}$ kodunun ortogonal tümleyeni $\mathcal{C}^{\perp}$,*

$$\mathcal{C}^{\perp} = \left\{\mathbf{x} \in \mathbb{F}_q^n : \forall \mathbf{y} \in \mathcal{C}, \mathbf{x}\cdot\mathbf{y} = \mathbf{0}\right\}$$

şeklinde tanımlanır. $\mathcal{C}^{\perp}$-$[n,n-k]$ koduna $\mathcal{C}$-$[n,k]$ kodunun dual kodu denir.

Tanım 2.8 (Eşlik-Denetim Matrisi)**.** *$\mathbb{F}_q$ üzerinde bir $\mathcal{C}$-$[n,k]$ kodu verilsin. $\mathcal{C}^{\perp}$ dual kodunun üreteç matrisi H ye $\mathcal{C}$ kodunun eşlik-denetim matrisi adı verilir.*

Eşlik-denetim matrisi $H \in \mathbb{F}_q^{(n-k)\times n}$ olan $\mathcal{C}$-$[n,k]$ kodu,

$$\mathcal{C} = \left\{\mathbf{c} \in \mathbb{F}_q^n \mid H\mathbf{c}^T = 0\right\}$$

şeklinde ifade edilir. Bu kodun dual kodu ise,

$$\mathcal{C}^{\perp} = \left\{\mathbf{u}H \mid \mathbf{u} \in \mathbb{F}_q^{n-k}\right\}$$

şeklindedir.

k boyutlu bir vektör uzayının bazı tek olmadığından bir $\mathcal{C}$ kodunun üreteç matrisi de tek değildir. G matrisine elementer satır işlemleri uygulandığında elde edilen matrisler de $\mathcal{C}$ kodunu üretir.

Tanım 2.9. *$G = [I_k \mid A]$ formunda yazılan üreteç matrisine standart formda üreteç matris denir. Benzer şekilde $H = [B \mid I_{n-k}]$ formunda yazılan eşlik-denetim matrisine standart formda eşlik-denetim matrisi denir.*

Standart formdaki G üreteç matrisinin ilk k-bitinin yerleri bilgi bitlerini gösterir. Yani bu bitlere bakılarak k uzunlukta bir $\mathbf{m}$ mesajı hakkında bilgi sahibi olunabilir.

Örnek 5. $\mathbb{F}_2$ üzerinde baz elemanları $\mathbf{v}_1 = (11010)$, $\mathbf{v}_2 = (10000)$, $\mathbf{v}_3 = (11101)$ vektörlerinden oluşan bir $\mathcal{C}$-$[5,3]$ kodu tanımlansın. Bazın ürettiği kod,

$$\mathcal{C} = \{(00000),(11010),(10000),(11101),(01010),(01101),(00111),(10111)\}$$

için Tanım 2.6'dan üreteç matris,

$$G = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}_{3\times 5}$$

olarak elde edilir. Eğer mesaj $\mathbf{m} = (m_1 m_2 m_3) \in \mathbb{F}_2^3$ ise $\mathcal{C}$ de buna karşılık gelen kod kelimesi, $\mathbf{m}G = m_1\mathbf{v}_1 + m_2\mathbf{v}_2 + m_3\mathbf{v}_3$ olur. Mesela $\mathbf{m} = (001)$ ise

$$\mathbf{m}G = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \end{bmatrix}_{1\times 5}$$

olur. İlk 3-bite bakıldığında mesajla ilgili bilgi elde edilemiyor. Eğer $\mathcal{C}$ kodu için $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ yerine üreteç matris standart forma getirilirse, $\mathbf{u}_1 = (10000)$, $\mathbf{u}_2 = (01010)$, $\mathbf{u}_3 = (00111)$ bazı seçilirse, ki burada aynı 3 boyutlu alt uzay elde edilir. Üreteç matris,

$$G' = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}_{3\times 5}$$

olur. Aynı mesaja denk gelen kod kelimesi,

$$\mathbf{m}G' = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \end{bmatrix}_{1\times 5}$$

bulunur. Bu kod kelimesinin ilk 3-bitinin mesajla aynı olduğu görülür.

Lemma 2.1. *$\mathbb{F}_q$ üzerinde bir $\mathcal{C}$-$[n,k]$ kodu, $k \times n$ lik G ve $(n-k) \times n$ lik H matrisleri verilsin. $\mathcal{C}$ kodunun üreteç matrisi G dir ancak ve ancak G nin satırları lineer bağımsız ve $GH^T = 0$ dır. Benzer şekilde, $\mathcal{C}$ kodunun eşlik-denetim matrisi H dir ancak ve ancak H nin satırları lineer bağımsız ve $HG^T = 0$ dır.*

Teorem 2.2. *Eğer $\mathcal{C}$-$[n,k]$ kodunun standart formdaki üreteç matrisi $G = [I_k \mid A]$ ise $\mathcal{C}$ nin eşlik-denetim matrisi $H = \left[-A^T \mid I_{n-k}\right]$ dir.*

Örnek 6. $q = 3$ olsun ve

$$S = \{(10101010), (11001101), (11110000), (01100110), (00111100)\}$$

bazının oluşturduğu $\mathcal{C}$-$[8,5]$ kodu için üreteç matrisi ve eşlik-denetim matrisini hesaplayalım. Bu bazdaki vektörler lineer bağımsız olduğundan üreteç matrisin tanımı gereğince,

$$G' = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}_{5\times 8}$$

yazılabilir. Daha sonra satırca indirgeme işlemleriyle eşelon forma getirilerek standart formdaki üreteç matris,

$$G = [I_5 \mid A] = \left[\begin{array}{ccccc|ccc} 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 \end{array}\right]_{5\times 8}$$

olarak elde edilir. Teorem 2.2 kullanılarak,

$$H = \left[-A^T \mid I_3\right] = \left[\begin{array}{ccccc|ccc} 1 & 2 & 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 & 1 & 0 & 0 & 1 \end{array}\right]_{3\times 8}$$

eşlik-denetim matrisi kolaylıkla elde edilir. Kontrol edildiğinde $\mathbb{F}_3$ üzerinde $GH^T = 0$ olduğu görülür.

Tanım 2.10 (Sendrom)**.** *$\mathcal{C}$-$[n,k]$ kodunun eşlik-denetim matrisi $H \in \mathbb{F}_q^{(n-k)\times n}$ ve $\mathbf{c} \in \mathbb{F}_q^n$ bir kod kelimesi olsun. O halde $\mathbf{c}$ kod kelimesinin sendromu $H\mathbf{c}^T$ olarak tanımlanır. $\mathbf{c} \in \mathcal{C}$ ancak ve ancak $H\mathbf{c}^T = 0$ dır.*

Kodlama teorisinde sendrom hesaplanarak kod çözümü yapılabilir.

2.1 Devirli Kodlar

Tanım 2.11 (Devirli)**.** *Eğer* $(a_0a_1\dots a_{n-2}a_{n-1}) \in S$ *iken* $(a_{n-1}a_0a_1\dots a_{n-2}) \in S$ *ise* $\mathbb{F}_q^n$ *cisminin alt uzayı S devirli alt uzaydır.*

Tanım 2.12 (Devirli Kod)**.** *Eğer* $\mathcal{C}$ *devirli bir alt uzaysa* $\mathcal{C}$ *lineer kodu devirli koddur.*

$\mathcal{C} = \{(000),(011),(101),(110)\} \subseteq \mathbb{F}_2^3$ kodu lineer devirli kodlara örnektir.

Devirli kodlar cebirsel yapıya aşağıdaki gibi dönüştürülür:

$$\begin{array}{rccl} \pi: & \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q[x]/\langle x^n-1\rangle \\ & (a_0a_1\dots a_{n-1}) & \longmapsto & a_0+a_1x+\cdots+a_{n-1}x^{n-1}. \end{array}$$

π dönüşümü $\mathbb{F}_q^n$ üzerinde bir lineer dönüşümdür. Bu kısımdan itibaren bazı durumlarda $\mathbb{F}_q^n$ uzayı $\mathbb{F}_q[x]/\langle x^n-1\rangle$ ile ve $\mathbf{a} = (a_0a_1\dots a_{n-1})$ vektörü $a(x) = a_0+a_1x+\cdots+a_{n-1}x^{n-1}$ polinomuyla ifade edilecektir. $\mathbb{F}_q[x]/\langle x^n-1\rangle$ bir halkadır ancak $n=1$ haricinde cisim değildir.

Örnek 7. $\mathcal{C} = \{(0000),(1100),(1010),(0110),(1001),(0101),(0011),(1111)\} \subseteq \mathbb{F}_2^4$ devirli kodu için,

$$\pi(\mathcal{C}) = \left\{0, 1+x, 1+x^2, x+x^2, 1+x^3, x+x^3, x^2+x^3, 1+x+x^2+x^3\right\} \subset \mathbb{F}_2[x]/\left\langle x^4-1\right\rangle$$

şeklindedir.

Teorem 2.3. $\mathbb{F}_q^n$ *uzayının boştan farklı bir alt kümesi* $\mathcal{C}$ *ancak ve ancak* $\mathbb{F}_q[x]/\langle x^n-1\rangle$ *halkasının bir ideali* $\pi(\mathcal{C})$ *ise bir devirli koddur.*

Örnek 7'deki devirli koda karşılık gelen $\mathbb{F}_2[x]/\left\langle x^4-1\right\rangle$ deki ideal $\pi(\mathcal{C})$ kümesidir.

Teorem 2.4. $\mathbb{F}_q[x]/\langle x^n-1\rangle$ *üzerinde I sıfırdan farklı bir ideal ve bu ideal içinde derecesi en küçük olan sıfırdan farklı monik polinom* $g(x)$ *olsun. Bu durumda* $g(x)$ *polinomu I idealinin üretecidir ve* $g(x) \mid x^n-1$ *dir.*

Örnek 7 deki $\pi(\mathcal{C})$ idealinin içindeki en küçük dereceli monik polinom $1+x$ dir ve $x^4-1 = (1+x)^4 \in \mathbb{F}_2[x]$ olduğundan $1+x \mid x^4-1$.

Tanım 2.13 (Üreteç Polinom)**.** *$\mathbb{F}_q[x]/\langle x^n-1\rangle$ üzerinde sıfırdan farklı bir I idealinin derecesi en küçük olan sıfırdan farklı tek monik polinomuna I nın üreteç polinomu denir. $\mathcal{C}$ devirli kodu için $\pi(\mathcal{C})$ nin üreteç polinomu aynı zamanda $\mathcal{C}$ kodunun üreteç polinomudur.*

Bu tanım gereğince,

$$\mathcal{C} = \{(0000),(1100),(1010),(0110),(1001),(0101),(0011),(1111)\} \subseteq \mathbb{F}_2^4$$

devirli kodunun üreteç polinomu $1+x$ dir.

Teorem 2.5. *$\mathbb{F}_q[x]/\langle x^n-1\rangle$ halkasının bir idealinin üreteç polinomu $g(x)$ olsun. Eğer $g(x)$ polinomunun derecesi $n-k$ ise ideale karşılık gelen devirli kodun boyutu k olur.*

Örnek 8. $x^4-1=(2+x)(1+x)(1+x^2)\in\mathbb{F}_3[x]$ çarpanlar kullanılarak iki adet üçlü $[4,2]$-devirli kodu bulunabilir. İlki $p(x)=(2+x)(1+x)=2+x^2$ polinomunun ürettiği,

$$\mathcal{C}_1 = \{(0000),(2010),(0201),(1020),(0102),(2211),(2112),(1122),(1221)\} \subseteq \mathbb{F}_3^4$$

kodu, ikincisi ise $q(x)=1+x^2$ polinomunun ürettiği,

$$\mathcal{C}_2 = \{(0000),(1010),(0101),(1111),(2020),(0202),(1212),(2121),(2222)\} \subseteq \mathbb{F}_3^4$$

kodudur.

Teorem 2.6. *$\mathbb{F}_q[x]/\langle x^n-1\rangle$ üzerinde $\mathcal{C}$ devirli kodunun* $\deg(g(x))=n-k$ *olacak şekilde üreteç polinomu $g(x)=g_0+g_1x+\cdots+g_{n-k}x^{n-k}$ olsun. Bu durumda,*

$$G=\begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix} = \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ \vdots & & & & & \ddots & & \vdots \\ 0 & 0 & \cdots & g_0 & g_1 & \cdots & & g_{n-k} \end{bmatrix}_{k\times n} \tag{2.1}$$

matrisi $\mathcal{C}$ kodunun üreteç matrisidir.

Örnek 8'deki $\mathcal{C}_1$ kodunun üreteç polinomu $g(x)=2+x^2$ için üreteç matris aşağıda

verilmiştir;

$$G = \begin{bmatrix} g(x) \\ xg(x) \end{bmatrix} = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \end{bmatrix}_{2\times 4}.$$

Devirli kodun üreteç matrisine elementer satır işlemleri uygulanarak eşlik-denetim matrisi bulunabilir. Ayrıca dual kodun da üreteç polinomu kullanılarak eşlik-denetim matrisine ulaşılabilir.

Tanım 2.14 (Ters Polinom (Reciprocal)). *$\mathbb{F}_q$ üzerinde $h(x) = \sum_{i=0}^{k} a_i x^i$ $(a_k \neq 0)$ derecesi k olan bir polinom olsun. $h(x)$ in ters polinomu,*

$$h_R(x) := x^k h(1/x) = \sum_{i=0}^{k} a_{k-i} x^i$$

şeklinde tanımlanır.

Teorem 2.7. *$\mathbb{F}_q$ üzerinde $\mathcal{C} - [n,k]$ devirli kodunun üreteç polinomu $g(x)$ ve $h(x) = \dfrac{x^n - 1}{g(x)}$ olsun. Bu durumda $h(x)$ polinomunun sabit terimi h_0 olmak üzere $\mathcal{C}^{\perp}$ kodunun üreteç polinomu $h_0^{-1} h_R(x)$ dir.*

Tanım 2.15 (Eşlik-Denetim Polinomu). *$\mathbb{F}_q$ üzerinde n uzunlukta bir devirli kod $\mathcal{C}$ ve $h(x) = \dfrac{x^n - 1}{g(x)}$ olsun. $h(x)$ polinomunun sabit terimi h_0 olmak üzere $h_0^{-1} h_R(x)$ e eşlik-denetim polinomu denir.*

Sonuç 2.2. *$\mathbb{F}_q$ üzerinde $\mathcal{C}$-$[n,k]$ devirli kodunun üreteç polinomu $g(x)$ ve $h(x) = \dfrac{x^n - 1}{g(x)}$ olsun. $h(x) = h_0 + h_1 x + \cdots + h_k x^k$ olmak üzere,*

$$H = \begin{bmatrix} h_R(x) \\ x h_R(x) \\ \vdots \\ x^{n-k-1} h_R(x) \end{bmatrix} = \begin{bmatrix} h_k & h_{k-1} & \cdots & h_0 & 0 & 0 & \cdots & 0 \\ 0 & h_k & h_{k-1} & \cdots & h_0 & 0 & \cdots & 0 \\ \vdots & & & & & \ddots & & \vdots \\ 0 & 0 & \cdots & h_k & h_{k-1} & \cdots & & h_0 \end{bmatrix}_{(n-k)\times n}$$

matrisi $\mathcal{C}$ kodunun eşlik-denetim matrisidir.

Örnek 9. $\mathbb{F}_2$ üzerinde $\mathcal{C}$ kodu $g(x) = 1 + x^2 + x^3 + x^4$ polinomu tarafından üretilen bir $[7,3]$-devirli kodu olsun. $h(x) = \dfrac{x^n - 1}{g(x)} = \dfrac{x^7 - 1}{1 + x^2 + x^3 + x^4} = 1 + x^2 + x^3$ bulunur.

Eşlik-denetim polinomu,

$$\begin{aligned} h_R(x) &= x^k h(1/x) \\ &= x^3(1+(1/x)^2+(1/x)^3) \\ &= 1+x+x^3 \end{aligned}$$

olur. $\mathcal{C}$ kodunun eşlik-denetim matrisi aşağıdaki gibidir.

$$H = \begin{bmatrix} h_R(x) \\ xh_R(x) \\ x^2h_R(x) \\ x^3h_R(x) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}_{4\times 7}.$$

2.2 Yarı-Devirli Kodlar

Bu bölümdeki ifadeler için [1], [2], [14] ve [19] kaynaklarından yararlanılmıştır.

Tanım 2.16 (Çevrimsel Matris)**.** $\mathbf{v} = (v_0 \dots v_{n-1}) \in \mathbb{F}_q^n$ *olsun.* $\mathbf{v}$ *vektörü tarafından üretilen çevrimsel matris,*

$$\mathbf{rot}(\mathbf{v}) = \begin{bmatrix} v_0 & v_{n-1} & \cdots & v_1 \\ v_1 & v_0 & \cdots & v_2 \\ \vdots & \vdots & \ddots & \vdots \\ v_{n-1} & v_{n-2} & \cdots & v_0 \end{bmatrix} \in \mathbb{F}_q^{n\times n}$$

şeklinde tanımlanır.

Tanım 2.17 (Yarı-Devirli Kodlar (QC))**.** n_0*,* k_0*, r pozitif tam sayılar,* $n = n_0 r$*,* $k = k_0 r$ *ve* $r \neq 1$ *olmak üzere* $[n,k]$*-lineer kodu* $\mathcal{C}$ *verilsin.* $\mathbf{x} = (x_0 x_1 \dots x_{r-1}) \in \mathbb{F}_q^n$ *vektörü r tane ardışık bloğa ayrılmış n uzunlukta bir vektör olsun. Eğer* $\forall \mathbf{c} = (c_0 c_1 \dots c_{r-1}) \in \mathcal{C}$ *iken* $c_0 c_1 \dots c_{r-1}$ *bloklarının çevrimsel kaydırılmasıyla oluşan yeni vektör kod kelimesi oluyorsa* $\mathcal{C}$ *koduna yarı-devirli kod denir.*

Başka bir ifadeyle, her bir $\mathbf{c}_i$ *bloğu* $\mathcal{R} = \mathbb{F}_q[x]/\langle x^{n_0}-1\rangle$ *halkasında bir polinom olsun. Eğer herhangi* $\mathbf{c} = (c_0 \dots c_{r-1}) \in \mathcal{C}$ *için* $(xc_0 \dots xc_{r-1}) \in \mathcal{C}$ *ise* $\mathcal{C}$ *kodu yarı-devirlidir.*

r sayısına yarı-devirli kodun mertebesi, n_0 sayısına ise indeksi denir.

G_i matrisleri $k_0 \times n_0$ boyutunda alt matrisler olmak üzere yarı-devirli kodun üreteç matrisi,

$$G = \begin{bmatrix} G_0 & G_{n-1} & \cdots & G_1 \\ G_1 & G_0 & \cdots & G_2 \\ \vdots & \vdots & \ddots & \vdots \\ G_{n-1} & G_{n-2} & \cdots & G_0 \end{bmatrix} \in \mathbb{F}_q^{k \times n}$$

şeklinde yazılabilir.

Örnek 10. [14] $\mathbb{F}_2$ üzerinde

$$G = \left[\begin{array}{ccc|ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{array}\right]_{3\times 9}$$

üreteç matrisi $[9,3]$ yarı-devirli kodunu üretir. Bu kodun elemanları ise Çizelge 2.1'de verilmiştir. İkinci kod kelimesi (111100110) üçlü bloklara ayrılıp bir kere çevrimsel

Çizelge 2.1: $[9,3]$ yarı-devirli kodunun kod kelimeleri

(000 000 000)
(111 100 110)
(110 111 100)
(100 110 111)
(001 011 010)
(011 010 001)
(010 001 011)
(101 101 101)

kayrıdırılırsa (110111100) yani üçüncü kod kelimesi elde edilir. Aynı kelime ikili bloklara ayrılıp kaydırılırsa (101111001) kelimesi elde edilir ancak bu kod ailesinin elemanı değildir. Buradan $r = 3$ olduğu görülebilir.

Yarı-devirli kodların üreteç matrisi ve eşlik-denetim matrisi $r \times r$ lik çevrimsel matrislerle de oluşturulabilir. Örneğin; $r = 5$ için $[15,5]$ yarı-devirli kodunun üreteç matrisi,

$$G = \left[\begin{array}{ccccc|ccccc|ccccc} 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{array}\right]_{5\times 15}$$

şeklindedir. Dikkat edilirse 5×5 boyutunda 3 matrise ayrılırsa her birinin çevrimsel matris olduğu görülebilir.

Tanım 2.18 (QC-MDPC Kodlar)**.** *Uzunluğu $n = n_0 r$, boyutu $k = k_0 r$, mertebesi r olan $[n,k]$ yarı-devirli kodunun eşlik-denetim matrisinin satır ağırlıkları $w = O(\sqrt{n})$ sabit sayısına eşit ise bu koda $[n,k,w]$-QC-MDPC kodu denir.*

$r = n-k$, $n = n_0 r$ ve r asal olsun. Bu durumda eşlik-denetim matrisi,

$$H = [H_0|H_1|\dots|H_{n_0-1}]_{(n-k)\times n}$$

şeklinde yazılır. Üreteç matrisi ise,

$$G = \left[\begin{array}{c|c} & (H_{n_0-1}^{-1}H_0)^T \\ & (H_{n_0-1}^{-1}H_1)^T \\ I_k & \vdots \\ & (H_{n_0-1}^{-1}H_{n_0-2})^T \end{array}\right]_{k\times n}$$

ile bulunabilir.

Örnek 11. $r = 5$ asal sayısı, $n = 10$ ve $w = 6$ olsun bu durumda $n_0 = 2$, $k = 5$ olur. $\mathbb{F}_2$ üzerinde, $[10,5,6]$-QC-MDPC kodunun eşlik-denetim matrisi,

$$H = [H_0|H_1] = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{array}\right]_{5\times 10}$$

olsun. Dikkat edilirse her bir satır ağırlığının 6 olduğu ve H_0, H_1 matrislerinin

çevrimsel olduğu görülebilir. Buradan yola çıkarak,

$$H_1^{-1} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}_{5\times 5}$$

olmak üzere,

$$G = \left[I_5 | (H_1^{-1}H_0)^T\right] = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{array}\right]_{5\times 10}$$

üreteç matrisi bulunur.

2.3 BCH Kodları

Bu bölümdeki ifadeler için [14], [15], [19] ve [24] kaynaklarından yararlanılmıştır.

Tanım 2.19 (BCH Kodu)**.** *α, $\mathbb{F}_{q^m}$ üzerinde bir primitif eleman ve $i \in \mathbb{Z}$ olmak üzere α^i elemanının $\mathbb{F}_q$ üzerinde minimal polinomu $m_{\alpha^i}(x)$ olsun. $\mathbb{F}_q$ üzerinde uzunluğu $n = q^m - 1$, planlanmış uzaklığı δ olan primitif BCH kodu, a bir tam sayı olmak üzere $g(x) = ekok\left(m_{\alpha^a}(x), m_{\alpha^{a+1}}(x), \ldots, m_{\alpha^{a+\delta-2}}(x)\right)$ polinomu tarafından üretilen bir q-lu devirli koddur.*

Bu tezde $a = 1$ seçilerek devam edilecektir. Eğer $q \neq 2$ ve her $1 \leq e \leq \delta - 1$ için $ebob(q^m - 1, e) = 1$ ise uzunluğu $n = q^m - 1$, planlanmış uzaklığı δ olan q-lu BCH kodunun boyutu $k = q^m - 1 - m(\delta - 1)$ dir. Bu kodun minimum uzaklığı da en az δ dır.

HQC algoritmasında ikili BCH kodları kullanılacaktır. Kodun hata düzeltme kapasitesi t ile minimum uzaklığı δ arasında $\delta = 2t + 1$ ilişkisi vardır. Herhangi pozitif $m \geq 3$

tam sayısı için,

- Blok uzunluğu $n = 2^m - 1$,
- Eşlik-denetim basamaklarının sayısı $n - k \leq mt$ öyle ki t kodun hata düzeltme kapasitesi ve k bilgi bitlerinin sayısı,
- Minimum uzaklık $d_{min} \geq 2t + 1 = \delta$.

Bu kod ailesi t veya daha az hata düzeltebildiği için t-hata düzelten BCH kod olarak adlandırılır ve BCH $[n,k,t]$ ile gösterilir. α, $\mathbb{F}_{2^m}$ üzerinde bir primitif eleman olmak üzere bu kodun $g(x)$ üreteç polinomu Tanım 2.19'dan,

$$g(x) = ekok\left(m_{\alpha^1}(x), m_{\alpha^2}(x), \ldots, m_{\alpha^{2t}}(x)\right)$$

şeklindedir. Yani $\mathbb{F}_2$ üzerinde kökleri $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$ olan en düşük dereceli polinomdur ($g(\alpha^i) = 0, 1 \leq i \leq 2t$). Aynı zamanda $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$ elemanlarının eşlenikleri de aynı minimal polinomlara sahip olduğundan bu eşlenik elemanlar da $g(x)$ polinomunun kökleridir.

α^i elemanının minimal polinomu $m_{\alpha^i}(x)$ olsun. Eğer i çift tam sayı ise i' tek sayı ve $l \geq 1$ olmak üzere,

$$i = i'2^l$$

şeklinde ifade edilebilir. $\alpha^i = (\alpha^{i'})^{2^l}$, $\alpha^{i'}$ nin eşleniğidir. Bu durumda α^i ve $\alpha^{i'}$ aşağıda verilen aynı minimal polinoma sahip olur,

$$m_{\alpha^i}(x) = m_{\alpha^{i'}}(x).$$

Dolayısıyla α elemanının her çift kuvveti kendinden önceki tek kuvvetiyle aynı minimal polinoma sahiptir. Sonuç olarak $n = 2^m - 1$ uzunluklu t-hata düzelten ikili BCH kodunun $g(x)$ üreteç polinomu,

$$g(x) = ekok\left(m_{\alpha^1}(x), m_{\alpha^3}(x), \ldots, m_{\alpha^{2t-1}}(x)\right) \tag{2.2}$$

ifadesine indirgenir. Bu BCH kodunun eşlik-denetim matrisi ise;

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{n-1} \\ 1 & (\alpha^3) & (\alpha^3)^2 & (\alpha^3)^3 & \cdots & (\alpha^3)^{n-1} \\ \vdots & & & & & \\ 1 & (\alpha^{2t-1}) & (\alpha^{2t-1})^2 & (\alpha^{2t-1})^3 & \cdots & (\alpha^{2t-1})^{n-1} \end{bmatrix}_{(n-k)\times n} \tag{2.3}$$

ile hesaplanır.

Örnek 12. $m = 4$ olsun. $\mathbb{F}_2$ üzerinde $p(x) = 1 + x + x^4$ primitif polinomdur. $p(x)$ polinomunun bir kökü α olsun yani $p(\alpha) = 1 + \alpha + \alpha^4 = 0$. O halde $\alpha^4 = 1 + \alpha$ eşitliği kullanılarak $\mathbb{F}_{2^4}$ cismi Çizelge 2.2'deki gibi oluşturulur. Tanım 1.4 gereğince 0

Çizelge 2.2: $p(x) = 1+x+x^4$ tarafından üretilen $\mathbb{F}_{2^4}$ cisminin elemanları

Kuvvet Gösterimi	Polinom Gösterimi	4-lü Gösterimi
0	0	(0 0 0 0)
1	1	(1 0 0 0)
α	α	(0 1 0 0)
α^2	α^2	(0 0 1 0)
α^3	α^3	(0 0 0 1)
α^4	$1+\alpha$	(1 1 0 0)
α^5	$\alpha+\alpha^2$	(0 1 1 0)
α^6	$\alpha^2+\alpha^3$	(0 0 1 1)
α^7	$1+\alpha+\alpha^3$	(1 1 0 1)
α^8	$1+\alpha^2$	(1 0 1 0)
α^9	$\alpha+\alpha^3$	(0 1 0 1)
α^{10}	$1+\alpha+\alpha^2$	(1 1 1 0)
α^{11}	$\alpha+\alpha^2+\alpha^3$	(0 1 1 1)
α^{12}	$1+\alpha+\alpha^2+\alpha^3$	(1 1 1 1)
α^{13}	$1+\alpha^2+\alpha^3$	(1 0 1 1)
α^{14}	$1+\alpha^3$	(1 0 0 1)

ve 1 elemanlarının her kuvveti yine kendisini verdiğinden eşlenikleri 0 ve 1 bulunur. α elemanının eşlenikleri,

$$C(\alpha) = \left\{\alpha, \alpha^2, \alpha^{2^2} = \alpha^4, \alpha^{2^3} = \alpha^8\right\}$$

kümesinin elemanlarıdır. α^3 elemanının eşlenikleri ise,

$$C(\alpha^3) = \left\{\alpha^3, (\alpha^3)^2 = \alpha^6, (\alpha^3)^{2^2} = \alpha^{12}, (\alpha^3)^{2^3} = \alpha^{24} = \alpha^9\right\}$$

kümesinin elemanlarından oluşur. α^5 elemanının eşlenikleri,

$$C(\alpha^5) = \left\{\alpha^5, (\alpha^5)^2 = \alpha^{10}\right\}$$

kümesinde verilmiştir. Son olarak α^7 elemanının eşlenikleri,

$$C(\alpha^7) = \left\{\alpha^7, (\alpha^7)^2 = \alpha^{14}, (\alpha^7)^{2^2} = \alpha^{28} = \alpha^{13}, (\alpha^7)^{2^3} = \alpha^{56} = \alpha^{11}\right\}$$

kümesinin elemanlarıdır. Teorem 1.1 kullanılarak her bir elemanın minimal polinomu bulunabilir. 0 elemanının minimal polinomu,

$$m_0(x) = \prod_{a \in C(0)} (x-a) = (x-0) = x$$

olur. 1 elemanının minimal polinomu,

$$m_1(x) = \prod_{a \in C(1)} (x-a) = (x-1) = 1+x$$

bulunur. α elemanının minimal polinomu,

$$\begin{aligned} m_\alpha(x) = \prod_{a \in C(\alpha)} (x-a) &= (x-\alpha)(x-\alpha^2)(x-\alpha^4)(x-\alpha^8) \\ &= 1+x+x^4 \end{aligned}$$

olur. Eşlenik elemanların minimal polinomları eşit olduğundan $\alpha^2, \alpha^4, \alpha^8$ elemanlarının da minimal polinomları eşittir. α^3 elemanının minimal polinomu,

$$\begin{aligned} m_{\alpha^3}(x) = \prod_{a \in C(\alpha^3)} (x-a) &= (x-\alpha^3)(x-\alpha^6)(x-\alpha^9)(x-\alpha^{12}) \\ &= 1+x+x^2+x^3+x^4 \end{aligned}$$

bulunur. $\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$ elemanları eşlenik olduğundan minimal polinomları da

aynıdır. Benzer şekilde α^5 elemanının minimal polinomu,

$$\begin{aligned} m_{\alpha^5}(x) = \prod_{a \in C(\alpha^5)} (x-a) &= (x-\alpha^5)(x-\alpha^{10}) \\ &= 1+x+x^2 \end{aligned}$$

α^5, α^{10} elemanları eşlenik olduğundan minimal polinomları da aynıdır. Son olarak α^7 elemanının minimal polinomu,

$$\begin{aligned} m_{\alpha^7}(x) = \prod_{a \in C(\alpha^7)} (x-a) &= (x-\alpha^7)(x-\alpha^{11})(x-\alpha^{13})(x-\alpha^{14}) \\ &= 1+x^3+x^4 \end{aligned}$$

olur. $\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$ elemanları eşlenik olduğundan minimal polinomları da aynıdır. Çizelge 2.3'te eşlenik elemanlar ve minimal polinomları verilmiştir.

Çizelge 2.3: $p(x) = 1+x+x^4$ tarafından üretilen $\mathbb{F}_{2^4}$ cisminin elemanlarının minimal polinomları

Eşlenik Kökler	Minimal Polinomlar
0	x
1	$1+x$
$\alpha, \alpha^2, \alpha^4, \alpha^8$	$1+x+x^4$
$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	$1+x+x^2+x^3+x^4$
$\alpha^5, \alpha^{10},$	$1+x+x^2$
$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	$1+x^3+x^4$

Bu bilgiler kullanılarak $n = 2^m - 1 = 2^4 - 1 = 15$ uzunlukta ve 2-hata düzelten BCH kodunun üreteç polinomu,

$$\begin{aligned} g(x) &= ekok\left(m_{\alpha^1}(x), m_{\alpha^3}(x)\right) \\ &= m_{\alpha}(x) m_{\alpha^3}(x) \\ &= (1+x+x^4)(1+x+x^2+x^3+x^4) \\ &= 1+x^4+x^6+x^7+x^8 \end{aligned} \tag{2.4}$$

şeklinde bulunur. Bu durumda kod $d_{min} \geq 5$ olacak şekilde $[15, 7]$ devirli koddur. Üreteç polinomun ağırlığı 5 olduğundan bu kodun minimum uzaklığı kesinlikle 5 tir. BCH

devirli bir kod olduğundan üreteç matrisi (2.1) eşitliği kullanılarak,

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}_{7\times 15} \tag{2.5}$$

bulunur. G matrisinin standart formdaki hali,

$$G' = \left[\begin{array}{ccccccc|cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right]_{7\times 15}.$$

matrisidir. Eşlik-denetim matrisi ise (2.3) eşitliğinden yararlanılarak,

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} & \alpha^{21} & \alpha^{24} & \alpha^{27} & \alpha^{30} & \alpha^{33} & \alpha^{36} & \alpha^{39} & \alpha^{42} \end{bmatrix}$$

$$= \left[\begin{array}{ccccccccccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{array}\right]_{8\times 15}$$

bulunur. H matrisinin standart formdaki hali,

$$H' = \left[\begin{array}{ccccccc|cccccccc} 1&1&0&1&0&0&0&1&0&0&0&0&0&0&0\\ 0&1&1&0&1&0&0&0&1&0&0&0&0&0&0\\ 0&0&1&1&0&1&0&0&0&1&0&0&0&0&0\\ 0&0&0&1&1&0&1&0&0&0&1&0&0&0&0\\ 1&1&0&1&1&1&0&0&0&0&0&1&0&0&0\\ 0&1&1&0&1&1&1&0&0&0&0&0&1&0&0\\ 1&1&1&0&0&1&1&0&0&0&0&0&0&1&0\\ 1&0&1&0&0&0&1&0&0&0&0&0&0&0&1 \end{array}\right]_{8\times 15}$$

matrisidir. Burada $GH^T = 0$ ve $G'H'^T = 0$ olduğu görülebilir.

2.3.1 BCH kodları için kod çözme

$n = 2^m - 1$ ($m \geq 0$ pozitif tam sayı) olmak üzere BCH $[n,k,t]$ kodu tanımlansın ve gönderilen kod kelimesi $v(x) = v_0 + v_1x + \cdots + v_{n-1}x^{n-1}$, ulaşan kelime ise $r(x) = r_0 + r_1x + \cdots + r_{n-1}x^{n-1}$ olsun. Hata polinomu $e(x) = e_0 + e_1x + \cdots + e_{n-1}x^{n-1}$ şeklinde ifade edilsin. Bunun anlamı $e_i = 1$ ise i konumunda bir hata vardır. Dolayısıyla $r(x) = v(x) + e(x)$ olur.

$\mathbb{F}_{2^m}$ üzerinde α bir primitif eleman ve $S_i = r(\alpha^i)$ olmak üzere sendromların kümesi $S_1, S_2, \ldots, S_{2t}$ şeklinde tanımlansın. O halde $v(\alpha^i) = 0$ (v kod kelimesi) olduğundan $r(\alpha^i) = e(\alpha^i)$ dir. $e(x)$ polinomunda $j_1, \ldots, j_t$ konumlarında t adet hata olsun. Bu durumda $e(x) = x^{j_1} + x^{j_2} + \cdots + x^{j_t}$ olur. Buradan $\alpha^{j_1}, \alpha^{j_2}, \ldots, \alpha^{j_t}$ lerin nerede olduğu bilinmemekle beraber aşağıdaki denklem sistemi elde edilir:

$$\begin{cases} S_1 &= \alpha^{j_1} + \alpha^{j_2} + \cdots + \alpha^{j_t} \\ S_2 &= (\alpha^{j_1})^2 + (\alpha^{j_2})^2 + \cdots + (\alpha^{j_t})^2 \\ S_3 &= (\alpha^{j_1})^3 + (\alpha^{j_2})^3 + \cdots + (\alpha^{j_t})^3 \\ &\vdots \\ S_{2t} &= (\alpha^{j_1})^{2t} + (\alpha^{j_2})^{2t} + \cdots + (\alpha^{j_t})^{2t}. \end{cases}$$

BCH kod çözme algoritmasının amacı bu denklem sistemini çözmektir. Hata konum sayıları $\beta_i = \alpha^{j_i}$ ile belirtilsin. Yukarıdaki denklem aşağıdaki gibi olur:

$$\begin{cases} S_1 &= \beta_1+\beta_2+\cdots+\beta_t \\ S_2 &= \beta_1^2+\beta_2^2+\cdots+\beta_t^2 \\ S_3 &= \beta_1^3+\beta_2^3+\cdots+\beta_t^3 \\ &\vdots \\ S_{2t} &= \beta_1^{2t}+\beta_2^{2t}+\cdots+\beta_t^{2t}. \end{cases}$$

Hata konum polinomu,

$$\begin{aligned} \sigma(x) &= (1+\beta_1 x)+(1+\beta_2 x)+\cdots+(1+\beta_t x) \\ &= 1+\sigma_1 x+\sigma_2 x^2+\cdots+\sigma_t x^t \end{aligned}$$

ile tanımlansın. $\sigma(x)$in kökleri hata konum sayılarının tersine, $\beta_1^{-1},\beta_2^{-1},\ldots,\beta_t^{-1}$, eşittir. Bu köklerin tersi hesaplanarak $e(x)$ hata polinomu bulunabilir.

BCH $[n,k,t]$ kodu için kod çözme aşamaları özetle aşağıdaki gibidir.

1. İlk adımda alınan polinom kullanılarak $2t$ tane sendrom hesaplanır.
2. İkinci adımda birinci adımda hesaplanmış olan $2t$ tane sendromdan hata konum polinomu $\sigma(x)$ bulunur. Bunu bulmak için Simplified Berklekamp algoritması [12] veya Genişletilmiş Öklid algoritması [24, Bölüm 6.4] kullanılabilir.
3. Üçüncü adımda $\sigma(x)$ hata konum polinomunun kökleri hesaplanıp tersleri alınarak hata konum sayıları bulunur.
4. Son olarak dördüncü adımda ise elde edilen polinomdaki hatalar düzeltilir.

2.4 Tekrarlı Kod

Tanım 2.20 (Tekrarlı Kod)**.** *$\mathbb{F}_q$ üzerinde yalnızca $(00\cdots0),(11\cdots1),\ldots,((q-1)(q-1)\cdots(q-1))$ kod kelimelerini içeren $(n,1)$ lineer blok koduna tekrarlı kod denir. Bu kodun üreteç matrisi,*

$$G=[11\cdots1]_{1\times n}$$

şeklindedir ve $\mathbb{1}_n$ ile gösterilir.

Örneğin, $\mathcal{C} = \{(00000), (11111), (22222)\} \subseteq \mathbb{F}_3^5$ devirli kodu $\mathbb{F}_3$ üzerinde bir tekrarlı koddur.

2.5 Tensör Çarpım Kodları

Tanım 2.21 (Tensör Çarpım Kodu)**.** *$\mathcal{C}_1$ ve $\mathcal{C}_2$, $\mathbb{F}_2$ üzerinde $[n_1, k_1, d_1]$ ve $[n_2, k_2, d_2]$ lineer kodları olsun. $\mathcal{C}_1 \otimes \mathcal{C}_2$ şeklinde ifade edilen $\mathcal{C}_1$ ve $\mathcal{C}_2$ nin Tensör Çarpım Kodu, satırları $\mathcal{C}_1$ in ve sütunları ise $\mathcal{C}_2$ in kod kelimelerinden oluşan tüm $n_1 \times n_2$ boyutundaki matrislerin kümesidir.*
Başka bir ifadeyle, eğer $\mathcal{C}_1$, G_1 tarafından ve $\mathcal{C}_2$, G_2 tarafından üretiliyorsa,

$$\mathcal{C}_1 \otimes \mathcal{C}_2 = \left\{ G_2^T X G_1 \mid X \in \mathbb{F}_2^{k_2 \times k_1} \right\}$$

olur. İki lineer kodun tensör çarpımı $[n_1 n_2, k_1 k_2, d_1 d_2]$ lineer kodudur.

Örnek 13. $\mathbb{F}_2$ üzerinde üreteç matrisi,

$$G_1 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}_{2\times 5}$$

olan $[5,2,3]$-$\mathcal{C}_1$ lineer kodu ve üreteç matrisi,

$$G_2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}_{1\times 3}$$

olan $[3,1,3]$-$\mathcal{C}_2$ tekrarlı kodu verilsin. Bu durumda bu iki kodun tensör çarpım kodu,

$$\begin{aligned} \mathcal{C}_1 \otimes \mathcal{C}_2 &= \left\{ G_2^T X G_1 \mid X \in \mathbb{F}_2^{1\times 2} \right\} \\ &= \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} X \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \mid X \in \mathbb{F}_2^{1\times 2} \right\} \\ &= \left\{ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix} \right\} \end{aligned}$$

şeklinde bulunur. Bu kod $[15,2,9]$-lineer koddur.

2.6 Rank Metrik Kodları

Bu bölümde, daha önce anlatılan Hamming metrik kullanan kodlardan farklı olarak rank metrik kodları hakkında bilgi verilecektir. Daha detaylı bilgi için [11], [20] kaynakları incelenebilir.

$p(x) \in \mathbb{F}_q[x]$ derecesi n olan bir polinom olsun. $\mathbb{F}_{q^m}[x]$ kümesinin $p(x)$ tarafından üretilen ideali $\langle p(x) \rangle$ olmak üzere, $\mathbb{F}_{q^m}^n$ vektör uzayı $\mathbb{F}_{q^m}[x]/\langle p(x) \rangle$ halkası ile tanımlanabilir. Aralarındaki ilişki ise

$$\begin{array}{rccl} \Psi: & \mathbb{F}_{q^m}^n & \longrightarrow & \mathbb{F}_{q^m}[x]/\langle p(x) \rangle \\ & (v_0, v_1, \ldots, v_{n-1}) & \longmapsto & \sum_{i=0}^{n-1} v_i x^i \end{array} \tag{2.6}$$

dönüşümüyle tanımlanabilir.

Tanım 2.22 (İdeal Matris)**.** *Derecesi n olan bir $p(x) \in \mathbb{F}_q[x]$ polinomu ve $\mathbf{v} \in \mathbb{F}_{q^m}^n$ verilsin.* $\mathbf{v}$ *tarafından üretilen ideal matris,*

$$\mathcal{IM}(\mathbf{v}) = \begin{bmatrix} \mathbf{v} \\ x\mathbf{v} \mod p(x) \\ \vdots \\ x^{n-1}\mathbf{v} \mod p(x) \end{bmatrix}_{n \times n}$$

şeklindeki $n \times n$ boyutundaki $\mathcal{IM}(\mathbf{v})$ matrisidir.

$\mathbb{F}_{q^m}[x]/\langle p(x) \rangle$ halkası üzerinde iki elemanın çarpımı:

$$\mathbf{uv} = \mathbf{u}\mathcal{IM}(\mathbf{v}) = \mathcal{IM}(\mathbf{u})^T\mathbf{v} = \mathbf{vu}$$

şeklindedir. $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) \in \mathbb{F}_{q^m}^n$ ve $\mathbb{F}_q$ üzerinde m boyutlu vektör uzayı olan $\mathbb{F}_{q^m}$ cisminin bir bazı $(\beta_0, \beta_1, \ldots, \beta_{m-1}) \in \mathbb{F}_{q^m}^m$ olsun. Her $\mathbf{x}_j$ koordinatı bu bazda $\mathbb{F}_q^m$ de bir vektöre denk gelir:

$$\mathbf{x}_j = \sum_{i=0}^{m-1} x_{ij}\beta_i \text{ yani } \mathbf{x}_j = (x_{0j} \ldots x_{(m-1)j}).$$

$\mathbf{x}$ vektörüne karşılık gelen $m \times n$ boyutundaki matris $0 \leq i \leq m-1, 0 \leq j \leq n-1$ olmak üzere $M(\mathbf{x}) = (x_{i,j}) \in \mathbb{F}_q^{m \times n}$ matrisidir, yani

$$M(\mathbf{x}) = \begin{bmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,(n-1)} \\ x_{1,0} & x_{1,1} & \cdots & x_{1,(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(m-1),0} & x_{(m-1),1} & \cdots & x_{(m-1),(n-1)} \end{bmatrix}_{m \times n}$$

matrisidir.

Tanım 2.23 (Rank Ağırlığı)**.** $\mathbf{x}$ *elemanının rank ağırlığı* $M(\mathbf{x})$ *matrisinin rankına eşittir,*

$$||\mathbf{x}|| = rank(M(\mathbf{x})).$$

şeklinde gösterilir.

İki $\mathbf{x}$ ve $\mathbf{y}$ elemanı arasındaki rank uzaklığı ise $d_R(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||$ şeklinde tanımlanır. $\mathcal{C}$ kodunun minimum rank uzaklığı $d_R(\mathcal{C})$ ise kodun birbirinden farklı olan bütün elemanlarının rank uzaklıklarının en küçüğüne eşittir. Minimum rank uzaklığı d_R olan $\mathcal{C}$ kodu $t = \left\lfloor \frac{d_R - 1}{2} \right\rfloor$ adet hata düzeltebilir.

Örnek 14. $\mathbb{F}_2$ üzerinde $p(x) = 1 + x + x^4$ polinomu verilsin. α bu polinomun kökü yani $1 + \alpha + \alpha^4 = 0$ olmak üzere bu polinom tarafından üretilen $\mathbb{F}_{2^4}$ cisminin elemanları Çizelge 2.2'de verilmiştir. $\mathbf{x} = (\alpha, \alpha^{12}, \alpha^{13}, \alpha^9) \in \mathbb{F}_{2^4}^4$ ve $\mathbf{y} = (1, \alpha^3, \alpha^5, \alpha^7) \in \mathbb{F}_{2^4}^4$ verilsin. Bu durumda

$$M(\mathbf{x}) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}_{4 \times 4} \qquad M(\mathbf{y}) = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}_{4 \times 4}$$

olmak üzere $||\mathbf{x}|| = \text{rank}(M(\mathbf{x})) = 3$ ve $||\mathbf{y}|| = \text{rank}(M(\mathbf{y})) = 4$ bulunur. Bu iki vektör arasındaki rank uzaklığı ise $d_R(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}|| = \text{rank}(M(\mathbf{x} - \mathbf{y})) = 4$ olarak bulunur.

Tanım 2.24 ($\mathbb{F}_{q^m}$-Lineer Kod)**.** *Rank metrik ile tanımlı* $\mathbb{F}_{q^m}^n$ *uzayının k boyutlu alt uzayı* $\mathcal{C}$ *koduna, uzunluğu n ve boyutu k olan* $\mathbb{F}_{q^m}$*-lineer kod denir.* $[n,k]_{q^m}$ *ile gösterilir.*

Her satırı $\mathcal{C}$ kodunun bazı olan $G \in \mathbb{F}_{q^m}^{k\times n}$ matrisi üreteç matris olmak üzere kod ailesi,

$$\mathcal{C} = \left\{ \mathbf{x}G \mid \mathbf{x} \in \mathbb{F}_{q^m}^{k} \right\}$$

ile veya $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ eşlik-denetim matrisi olmak üzere,

$$\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{F}_{q^m}^{n} \mid H\mathbf{x}^T = 0 \right\}$$

ile ifade edilir.

Tanım 2.25 (Vektörün Desteği)**.** $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) \in \mathbb{F}_{q^m}^{n}$ *olsun.* $\mathbf{x}$ *vektörünün koordinatlarıyla üretilen* $\mathbb{F}_{q^m}$ *cisminin* $\mathbb{F}_q$*-alt uzayına* $\mathbf{x}$ *vektörünün desteği denir ve* $Supp(\mathbf{x})$ *ile gösterilir. Bu durumda,*

$$Supp(\mathbf{x}) = span\left\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}\right\}_{\mathbb{F}_q}$$

ve $boy(Supp(\mathbf{x})) = ||\mathbf{x}|| = rank(M(\mathbf{x}))$ *olur.*

2.6.1 İdeal kodlar

Kod tabanlı kriptografinin zorluklarından biri anahtar boyutudur. Kodun gösteriminideki boyutu azaltmak için, ideal kod ailesi tanıtılmıştır. Bu kodlar ideal matris bloklarından oluşan standart bir üreteç matrise sahip kodlardır.

Tanım 2.26 (İdeal Kod)**.** $p(x) \in \mathbb{F}_q[x]$ *derecesi n olan bir polinom olsun.* $1 \leq i \leq s-t$ *ve* $1 \leq j \leq t$ *için* $(g_{i,j}) \in \mathbb{F}_{q^m}^{n}$ *vektörleri olmak üzere eğer bir* $[ns, nt]_{q^m}$ *kodu* $\mathcal{C}$*,*

$$G = \left[\begin{array}{c|ccc} & \mathcal{IM}(g_{1,1}) & \cdots & \mathcal{IM}(g_{1,s-t}) \\ I_{tn} & \vdots & \ddots & \vdots \\ & \mathcal{IM}(g_{t,1}) & \cdots & \mathcal{IM}(g_{t,s-t}) \end{array}\right]_{nt\times ns}$$

formundaki standart üreteç matrisine sahipse $\mathcal{C}$ *koduna* $[s,t]$*-ideal kod denir. Bu durumda* $\mathcal{C}$ *kodu* $(g_{i,j})$ *tarafından üretilir.*

İdeal kodlar, yarı-devirli kodların genelleştirilmiş hali olarak görülebilir. Yarı-devirli

kodun standart formdaki üreteç matrisi de aynı formdadır. Aradaki tek fark ise çevrimsel matrislerin yerini burada ideal matrislerin almasıdır. $n \times n$ boyutundaki çevrimsel matrisler $\mathbb{F}_{q^m}[x]/\langle x^n - 1\rangle$ kümesinin elemanı olarak görülebilir. Bu nedenle ideal kodlar yarı-devirli kodlardan $p(x)$ polinomunun seçimine bağlı olarak farklılaşır.

Eğer $(g_1, \ldots, g_{s-1})$ tarafından üretilen $\mathcal{C}$ kodu $[sn, n]$-ideal kodu ise $\mathcal{C} = \left\{(\mathbf{u}, ug_1, \ldots, \mathbf{u}g_{s-1}), \mathbf{u} \in \mathbb{F}_{q^m}^n\right\}$ şeklinde elde edilir.

$$H = \left[\begin{array}{c|c} & \mathcal{IM}(h_1) \\ I_{n(s-1)} & \vdots \\ & \mathcal{IM}(h_{s-1}) \end{array}\right]_{n(s-1)\times n}$$

matrisi bu kodun standart formdaki eşlik-denetim matrisidir.

Örnek 15. $s = 3, t = 1, n = 4$ ve $m = 3$ olsun. $p(x) = 1 + x + x^3$ tarafından üretilen $\mathbb{F}_{2^3}$ cisminin elemanları $1 + \alpha + \alpha^3 = 0$ olmak üzere Çizelge 2.2'de verilmiştir. $1 \leq i \leq 2$ ve $1 \leq j \leq 1$ için $(g_{i,j}) \in \mathbb{F}_{2^3}^4$ vektörleri $g_{1,1} = (\alpha, 1, \alpha)$, $g_{2,1} = (1, \alpha, \alpha^2)$ olsun. Tanım 2.22 gereğince,

$$\mathcal{IM}(g_{1,1}) = \begin{bmatrix} g_{1,1} \\ xg_{1,1} \mod p(x) \\ x^2 g_{1,1} \mod p(x) \end{bmatrix} = \begin{bmatrix} \alpha & 1 & \alpha \\ \alpha & 0 & 1 \\ 1 & \alpha^3 & 0 \end{bmatrix}_{3\times 3}$$

$$\mathcal{IM}(g_{2,1}) = \begin{bmatrix} g_{2,1} \\ xg_{2,1} \mod p(x) \\ x^2 g_{2,1} \mod p(x) \end{bmatrix} = \begin{bmatrix} 1 & \alpha & \alpha^2 \\ \alpha^2 & 1 & \alpha^4 \\ \alpha & \alpha^4 & \alpha^6 \end{bmatrix}_{3\times 3}$$

bulunur. Bu durumda Tanım 2.26'dan $[3,1]$-ideal kodun üreteç matrisi,

$$\begin{aligned} G &= \left[\begin{array}{c|cc} I_3 & \mathcal{IM}(g_{1,1}) & \mathcal{IM}(g_{1,2}) \end{array}\right] \\ &= \left[\begin{array}{ccc|cccccc} 1 & 0 & 0 & \alpha & 1 & \alpha & 1 & \alpha & \alpha^2 \\ 0 & 1 & 0 & \alpha & 0 & 1 & \alpha^2 & 1 & \alpha^4 \\ 0 & 0 & 1 & 1 & \alpha^3 & 0 & \alpha & \alpha^4 & \alpha^6 \end{array}\right]_{3\times 9} \end{aligned}$$

şeklindedir. Bu üreteç matrisin ürettiği kod ailesi aynı zamanda $[9,3]_{2^3}$-lineer koddur.

Singleton Sınırı. $\mathbb{F}_{q^m}$ üzerinde minimum rankı d_R olan $[n,k]$-lineer kodları için klasik *Singleton sınırı*, rank metrik için de uygulanabilir. Lineer kodlarda Hamming metrik için $d \leq n-k+1$ olan bu sınır rank metrik kodları için de $d_R \leq n-k+1$ şeklindedir. $n > m$ olduğunda bu sınır,

$$d_R \leq 1 + \left\lfloor \frac{(n-k)m}{n} \right\rfloor$$

şeklinde yeniden yazılabilir. Bu sınıra sahip olan kodlar *Maximum Rank Distance* (MRD) kodları olarak adlandırılır.

2.7 Gabidulin Kodları

Gabidulin kodları 1985'te ortaya çıkmıştır [10]. Bu kodlar Hamming metrikteki Reed-Solomon kodlarına [22] eşdeğerdir. Ancak burada normal polinomlar yerine q-polinomları kullanılarak cebirsel alt yapısı güçlendirilmiştir.

Tanım 2.27 (q-polinomları)**.** *$\mathbb{F}_{q^m}$ üzerindeki q-polinomlarının kümesi,*

$$\left\{ p(x) = \sum_{i=0}^{r} p_i x^{q^i} \mid p_i \in \mathbb{F}_{q^m}, p_r \neq 0 \right\}$$

dir. Bir $p(x)$ q-polinomunun q-derecesi $\deg_q(p) = r$ şeklinde tanımlıdır.

Tanım 2.28 (Gabidulin Kodu)**.** *$k \leq n \leq m$ olacak şekilde $k,n,m \in \mathbb{N}$ sayıları verilsin. $g = (g_1,\ldots,g_n)$, $\mathbb{F}_{q^m}$ cisminin elemanlarının bir $\mathbb{F}_q$ lineer bağımsız ailesi olsun. $\mathcal{G}_g[n,k,m]$ Gabidulin kodu,*

$$\left\{ p(g) \mid \deg_q(p) < k \right\} \quad \text{öyle ki} \quad p(g) := (p(g_1),\ldots,p(g_v))$$

olacak şekilde $[n,k]_{q^m}$ kodudur. $\mathcal{G}_g$ kodunun $n \times k$ boyutundaki üreteç matrisi,

$$G = \begin{bmatrix} g_1 & \cdots & g_1 \\ g_1^q & \cdots & g_n^q \\ \vdots & \ddots & \vdots \\ g_1^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{bmatrix}_{k \times n}$$

şeklindedir.

Bu kodlar $\left\lfloor \frac{n-k}{2} \right\rfloor$ hata düzeltebilir.

Örnek 16. $k=2$, $n=3$, $m=4$ ve $\mathbb{F}_2$ üzerinde $p(x)=1+x+x^4$ polinomu verilsin. α bu polinomun kökü yani $1+\alpha+\alpha^4=0$ olmak üzere bu polinom tarafından üretilen $\mathcal{C}$ cisminin elemanları Çizelge 2.2'de verilmiştir. $\mathbb{F}_2$ lineer bağımsız elemanlardan oluşan $g=(1,\alpha,\alpha^2)$ seçilsin. Bu durumda Tanım 2.28 gereğince $\mathcal{G}_g[3,2,4]$ Gabidulin kodunun üreteç matrisi,

$$G=\begin{bmatrix} 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha^4 \end{bmatrix}_{2\times 3}$$

şeklinde bulunur. Kod ailesi ise,

$$\left\{p(g) \mid \deg_q(p)<2\right\} \text{ öyle ki } p(g):=(p(1),p(\alpha),p(\alpha^2)) \tag{2.7}$$

kümesiyle ifade edilir. Tanım 2.27'den $\deg_q(p)<2$ olan iki adet polinom vardır: $p_0\neq 0$ olmak üzere $p(x)=p_0x^{2^0}=p_0x\in\mathbb{F}_{2^4}[x]$ ve $p_1\neq 0$ olmak üzere $p(x)=p_0x^{2^0}+p_1x^{2^2}=p_0x+p_1x^2\in\mathbb{F}_{2^4}[x]$.

Bu durumda (2.7) eşitliğindeki $p(g):=(p(1),p(\alpha),p(\alpha^2))$ elemanları bulunabilir. Öncelikle $p_0\neq 0$ olmak üzere $p(x)=p_0x$ olsun.

$$\begin{array}{llll}
p(x)=x & \Rightarrow p(g)=(1,\alpha,\alpha^2) & p(x)=\alpha^8x & \Rightarrow p(g)=(\alpha^8,\alpha^9,\alpha^{10}) \\
p(x)=\alpha x & \Rightarrow p(g)=(\alpha,\alpha^2,\alpha^3) & p(x)=\alpha^9x & \Rightarrow p(g)=(\alpha^9,\alpha^{10},\alpha^{11}) \\
p(x)=\alpha^2x & \Rightarrow p(g)=(\alpha^2,\alpha^3,\alpha^4) & p(x)=\alpha^{10}x & \Rightarrow p(g)=(\alpha^{10},\alpha^{11},\alpha^{12}) \\
p(x)=\alpha^3x & \Rightarrow p(g)=(\alpha^3,\alpha^4,\alpha^5) & p(x)=\alpha^{11}x & \Rightarrow p(g)=(\alpha^{11},\alpha^{12},\alpha^{13}) \\
p(x)=\alpha^4x & \Rightarrow p(g)=(\alpha^4,\alpha^5,\alpha^6) & p(x)=\alpha^{12}x & \Rightarrow p(g)=(\alpha^{12},\alpha^{13},\alpha^{14}) \\
p(x)=\alpha^5x & \Rightarrow p(g)=(\alpha^5,\alpha^6,\alpha^7) & p(x)=\alpha^{13}x & \Rightarrow p(g)=(\alpha^{13},\alpha^{14},1) \\
p(x)=\alpha^6x & \Rightarrow p(g)=(\alpha^6,\alpha^7,\alpha^8) & p(x)=\alpha^{14}x & \Rightarrow p(g)=(\alpha^{14},1,\alpha) \\
p(x)=\alpha^7x & \Rightarrow p(g)=(\alpha^7,\alpha^8,\alpha^9) & &
\end{array}$$

Benzer şekilde $p_1 \neq 0$ olmak üzere $p(x) = p_0 x + p_1 x^2$ polinomu içinde değerler verilerek kod kelimeleri bulunabilir.

Gabidulin kodları için kod çözme algoritmaları için [5, Bölüm 4, Algoritma 5], [16] ve [20] kaynakları incelenebilir.

3. HQC KUANTUM SONRASI ALGORİTMASI

Bu bölümde kod tabanlı açık anahtarlı kripto sistem olan HQC hakkında bilgiler verilecektir. Detaylı bilgi için [19] incelenebilir.

HQC kripto sisteminin içeriğinde hem şifreleme-şifre çözme algoritması hem de anahtar kapsülleme mekanizması vardır. Kodların yapısındaki sendromla kod çözmenin dayanıklılığı sayesinde IND-CPA[1] güvenli olduğunu kanıtlamıştır. Aynı zamanda IND-CCA2[2] güvenli ve iyi bir verimliliğe sahiptir. Kullanılan kodun gizli yapısını kurtarmayı hedefleyen ataklara karşı güvenlidir. Başarılı olarak ikinci tura geçmiştir. Ancak bu kripto sistemin açık anahtarlı şifreleme versiyonu için bir kısıt düşük şifreleme oranıdır. NIST tarafından uygun görülen 256-bitlik metni şifreleyebilir ama bu oranın yükselmesi parametrelerin yükselmesine neden olmaktadır.

Algoritma boyunca $n \in \mathbb{Z}$ pozitif tam sayısı için, $\mathcal{R} = \mathbb{F}_2[x]/\langle x^n - 1\rangle$ şeklinde tanımlıdır. Eğer $\dfrac{x^n-1}{x-1}$ polinomu $\mathcal{R}$ de indirgenemez ise n asal tam sayısı primitiftir. n nin primitif seçilmesinin sebebi ise $x^n - 1$ ifadesinin düşük dereceli çarpanları olduğunda yapılan ataklara karşı dayanıksızlaşmasıdır. Burada matrislerin çevrimsel olması HQC algoritmasına büyük avantaj sağlamaktadır. HQC iki çeşit kod kullanır: $G \in \mathbb{F}_2^{k\times n}$ matrisi tarafından üretilen $[n,k]$ kodu $\mathcal{C}$ ve eşlik-denetim matrisi $(\mathbf{1},\mathbf{h})$ olan rastgele çift çevrimsel $[2n,n]$ kodu.

Kapsülleme ve kapsülden çıkarma algoritmalarında kullanılan $\mathcal{G},\mathcal{H}$ ve $\mathcal{K}$ fonksiyonları özet fonksiyonlardır. NIST genellikle SHA512 kullanılmasını önerir.

[1]IND-CPA: Rakip, eşit uzunlukta iki mesaj üretir. Meydan okuyan, bunlardan birini şifrelemeye rastgele karar verir. Rakip, hangi mesajların şifrelenmiş olduğunu tahmin etmeye çalışır.

[2]IND-CCA1: Hedefi IND-CPA ile aynıdır. Rakip ek bir yeteneğe sahiptir: bir şifreleme veya şifre çözme oracle'ı çağırmak. Bunun anlamı, rakip, şifreli mesajı elde etmeden önce rastgele mesajları şifreleyebilir veya şifresini çözebilir.

IND-CCA2: IND-CCA1 kapsamındaki yeteneklerine ek olarak, karşı tarafa şifreli mesajı aldıktan sonra oracle'a erişme izni verilir, ancak şifreli metni şifre çözme sistemine gönderemez.

Ancak ataklarla $\mathbf{m}$ hakkında bilgi sahibi olunmasına sebebiyet verebileceğinden dolayı $\mathcal{G}$ ve $\mathcal{H}$ özet fonksiyonların aynı seçilmesi önerilmemektedir. Bu nedenle HQC algoritması için $\mathcal{G}$ için sözde rastgele fonksiyon (örneğin AES-based seed expander) $\mathcal{H}$ için SHA512 kullanılması önerilmektedir.

Bu kripto sistemde $\mathcal{C}_1$ kodu, uzunluğu n_1, boyutu k_1 ve hata düzeltme kapasitesi δ_1 olan BCH $[n_1, k_1, t_1]$ ve $\mathcal{C}_2$ kodu, uzunluğu n_2, boyutu 1 ve hata düzeltme kapasitesi $t_2 = \left\lfloor \frac{n_2-1}{2} \right\rfloor$ olan $\mathbb{1}_{n_2}$ tekrarlı kodu olmak üzere $\mathcal{C}_1 \otimes \mathcal{C}_2$ tensör çarpım kodu kullanılacaktır.

Kod çözme algoritması olarak herhangi bir algoritma kullanılabilir. Anahtar değişimi ya da kimlik doğrulama algoritmalarında verinin küçük bir kısmının (80, 128, 256-bit) şifrelenmesi gerekmektedir. Bu durum göz önüne alındığında HQC için birçok hata düzeltebilen ama düşük oranlı kodlara ihtiyaç duyulmaktadır. Yüksek hata oranına sahip verimli kod çözülebilir kodlar ile daha küçük hata oranına sahip daha az verimli kod çözebilir kodlar arasında bir seçim gereklidir. Bu şekilde iyi kod çözme özelliklerine sahip kodların ailesine örnek olarak Tensör Çarpım kodları verilmiştir. Bu kodlar benzer sorunlar ortaya çıktığında biometri [8] için kullanılmıştır.

Açıklama 3.1. *Kripto sistemin Hamming metrik versiyonunda,* $\mathbf{m} \in \mathbb{F}_2^{k_1}$ *mesajı ilk olarak BCH* $[n_1, k_1 = k, t_1]$ *kodu kullanılarak* $\mathbf{m}_1 \in \mathbb{F}_2^{n_1}$ *şeklinde kodlanır. Daha sonra* $\mathbf{m}_1$ *in her* $\mathbf{m}_{1,i}$ *koordinatı* $\mathbb{1}_{n_2}$ *tekrarlı kod ile* $\tilde{\mathbf{m}}_{1,i} \in \mathbb{F}_2^{n_2}$ *yeniden kodlanır. Bu durumda tensör çarpım kodunun uzunluğu* $n_1 n_2$ *(pratikte cebirsel ataklardan kaçınmak için uzunluk n,* $n_1 n_2$ *den büyük olan en küçük primitif asaldır.), boyutu* $k = k_1 \times 1$ *ve kodlanan vektör* $\mathbf{m}G = \tilde{\mathbf{m}} = (\tilde{\mathbf{m}}_{1,0}, \ldots, \tilde{\mathbf{m}}_{1,n_1-1}) \in \mathbb{F}_2^{n_1 n_2}$ *ile ifade edilir.*

$\mathbb{F}_2$ üzerindeki tekrarlı kodlar için kullanılan The Majority Decoding etkili bir kod çözme algoritmasıdır ve

$$\mathbb{1}_{n_2}.\mathsf{Decode}(\tilde{\mathbf{m}}_{1,j}) = \begin{cases} 1 \quad \text{eğer} \quad \sum_{i=0}^{n_2-1} \tilde{\mathbf{m}}_{1,j,i} \geq \left\lceil \frac{n_2+1}{2} \right\rceil \\ 0 \quad \text{diğer durumlar} \end{cases} \tag{3.1}$$

şeklinde ifade edilir.

3.1 Parametreler

- n: kod uzunluğu.
- k: kod boyutu.
- t: hata düzeltme kapasitesi.
- w: $\mathbf{x}$ ve $\mathbf{y}$ nin ağırlıkları.
- w_r: $\mathbf{r}_1$ ve $\mathbf{r}_2$ nin ağırlıkları.
- w_e: $\mathbf{e}$ nin ağırlığı.

3.2 Anahtar Üretimi

Parametreler kullanılarak pk açık anahtarı ve sk gizli anahtarı oluşturulur.

1. $\mathcal{R}$ üzerinde n uzunlukta rastgele $\mathbf{h}$ üretilir.
2. $\mathcal{C}$ kodunun $(k \times n)$ boyutlu $G \in \mathbb{F}_2^{k \times n}$ üreteç matrisi oluşturulur.
3. $\mathcal{R}$ üzerinde $\mathrm{wt}(\mathbf{x}) = \mathrm{wt}(\mathbf{y}) = w$ olacak şekilde n uzunlukta rastgele $\mathbf{x}$ ve $\mathbf{y}$ vektörleri üretilir ve uzunluğu $2n$ olan $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$ gizli anahtarı elde edilir.
4. n uzunlukta $\mathbf{s} = \mathbf{x} + \mathbf{hy}$ vektörü hesaplanır ve uzunluğu $2n$ olan $\mathsf{pk} = (\mathbf{h}, \mathbf{s})$ açık anahtarı bulunur.
5. $(\mathsf{pk}, \mathsf{sk})$ çıktısı verilir.

3.3 Şifreleme

$\mathsf{pk} = (\mathbf{h}, \mathbf{s})$ açık anahtarı, G üreteç matrisi ve diğer rastgele değerler kullanılarak elde edilen $\mathbf{c}$ şifreli metni karşı tarafa gönderilir.

1. $\mathcal{R}$ üzerinde $\mathrm{wt}(\mathbf{e}) = w_e$ ve $\mathrm{wt}(\mathbf{r}_1) = \mathrm{wt}(\mathbf{r}_2) = w_r$ olacak şekilde n uzunlukta rastgele $\mathbf{e}$, $\mathbf{r}_1$ ve $\mathbf{r}_2$, $(\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2))$ elemanları üretilir.

2. Uzunlukları n olan $\mathbf{u} = \mathbf{r}_1 + \mathbf{h}\mathbf{r}_2$ ve $\mathbf{v} = \mathbf{m}G + \mathbf{s}\mathbf{r}_2 + \mathbf{e}$ vektörleri hesaplanır.

3. $2n$ uzunluğunda $\mathbf{c} = (\mathbf{u}, \mathbf{v})$ çıktısı verilir.

3.4 Şifre Çözme

Alınan şifreli metin $\mathbf{c} = (\mathbf{u}, \mathbf{v})$ ve gizli anahtar $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$ kullanılarak $\mathbf{m}$ mesajına ulaşılır.
$\mathbf{v} - \mathbf{u}\mathbf{y}$ hesaplanarak seçilen kod ailesi için uygun bir kod çözme algoritması kullanılır.

$$\begin{aligned} \mathbf{v} - \mathbf{u}\mathbf{y} &= \mathbf{m}G + \mathbf{s}\mathbf{r}_2 + \mathbf{e} - (\mathbf{r}_1 + \mathbf{h}\mathbf{r}_2)\mathbf{y} \\ &= \mathbf{m}G + (\mathbf{x} + \mathbf{h}\mathbf{y})\mathbf{r}_2 + \mathbf{e} - (\mathbf{r}_1 + \mathbf{h}\mathbf{r}_2)\mathbf{y} \\ &= \mathbf{m}G + \mathbf{x}\mathbf{r}_2 + \mathbf{h}\mathbf{y}\mathbf{r}_2 + \mathbf{e} - \mathbf{r}_1\mathbf{y} - \mathbf{h}\mathbf{r}_2\mathbf{y} \\ &= \mathbf{m}G + \mathbf{x}\mathbf{r}_2 + \mathbf{e} - \mathbf{r}_1\mathbf{y} \end{aligned}$$

olduğundan $\mathbf{e}' = \mathbf{x}\mathbf{r}_2 - \mathbf{r}_1\mathbf{y} + \mathbf{e}$ hata vektörü olmak üzere $\mathrm{wt}(\mathbf{e}') \leq t$ sağlandığında kod çözümü yapılarak $\mathbf{m}$ mesajı kurtarılır.

3.5 Kapsülleme

$\mathsf{pk} = (\mathbf{h}, \mathbf{s})$ açık anahtarı ve G üreteç matrisi kullanılarak kapsülleme yapılır.

1. $\mathbb{F}_2$ üzerinde k uzunlukta rastgele $\mathbf{m}$ üretilir.

2. $\theta = \mathcal{G}(\mathbf{m})$ rastgelelik değişkeni $\mathbf{m}$ mesajının özet değeri olarak hesaplanır. Bu θ ya bağlı olarak bir sonraki adımda rastgele elemanlar üretilir.

3. $\mathcal{R}$ üzerinde $\mathrm{wt}(\mathbf{e}) = w_e$ ve $\mathrm{wt}(\mathbf{r}_1) = \mathrm{wt}(\mathbf{r}_2) = w_r$ olacak şekilde uzunlukları n olan rastgele $\mathbf{e}$, $\mathbf{r}_1$ ve $\mathbf{r}_2$ elemanları üretilir.

4. Uzunluğu $2n$ olan şifreli metin $\mathbf{c} = (\mathbf{u}, \mathbf{v}) = (\mathbf{r}_1 + \mathbf{h}\mathbf{r}_2, \mathbf{m}G + \mathbf{s}\mathbf{r}_2 + \mathbf{e})$ hesaplanır.

5. $K = \mathcal{K}(\mathbf{m}, \mathbf{c})$ simetrik anahtarı türetilir.

6. $\mathbf{d} = \mathcal{H}(\mathbf{m})$ hesaplanır.

7. $(\mathbf{c}, \mathbf{d})$ karşı tarafa gönderilir.

3.6 Kapsülden Çıkarma

Alınan $(\mathbf{c}, \mathbf{d})$ ve gizli anahtar $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$ kullanılarak anahtar kapsülden çıkarılır.

1. $\mathcal{C}$.Decode$(\mathbf{v} - \mathbf{uy})$ ile $\mathbf{m}'$ bulunur. Yani $\mathbf{v} - \mathbf{uy}$ hesaplanarak seçilen kod ailesi için uygun bir kod çözme algoritması kullanılır.

$$\begin{aligned} \mathbf{v} - \mathbf{uy} &= \mathbf{m}G + \mathbf{sr}_2 + \mathbf{e} - (\mathbf{r}_1 + \mathbf{hr}_2)\mathbf{y} \\ &= \mathbf{m}G + (\mathbf{x} + \mathbf{hy})\mathbf{r}_2 + \mathbf{e} - (\mathbf{r}_1 + \mathbf{hr}_2)\mathbf{y} \\ &= \mathbf{m}G + \mathbf{xr}_2 + \mathbf{hyr}_2 + \mathbf{e} - \mathbf{r}_1\mathbf{y} - \mathbf{hr}_2\mathbf{y} \\ &= \mathbf{m}G + \mathbf{xr}_2 + \mathbf{e} - \mathbf{r}_1\mathbf{y} \end{aligned}$$

   olduğundan $\mathbf{e}' = \mathbf{xr}_2 - \mathbf{r}_1\mathbf{y} + \mathbf{e}$ hata vektörü olmak üzere $\mathrm{wt}(\mathbf{e}') \leq t$ sağlandığında kod çözümü yapılarak $\mathbf{m}'$ mesajı kurtarılır.

2. $\theta' = \mathcal{G}(\mathbf{m}')$ rastgelelik değişkeni $\mathbf{m}'$ mesajının özet değeri olarak hesaplanır. Bu θ' ya bağlı olarak bir sonraki adımda rastgele elemanlar üretilir.

3. $\mathcal{R}$ üzerinde $\mathrm{wt}(\mathbf{e}) = w_e$ ve $\mathrm{wt}(\mathbf{r}_1) = \mathrm{wt}(\mathbf{r}_2) = w_r$ olacak şekilde rastgele $\mathbf{e}$, $\mathbf{r}_1$ ve $\mathbf{r}_2$ üretilir.

4. Şifreli metin $\mathbf{c}' = (\mathbf{u}, \mathbf{v}) = (\mathbf{r}_1 + \mathbf{hr}_2, \mathbf{m}'G + \mathbf{sr}_2 + \mathbf{e})$ hesaplanır.

5. Eğer $\mathbf{c} \neq \mathbf{c}'$ ya da $\mathbf{d} \neq \mathcal{H}(\mathbf{m}')$ ise hata verir.

6. Diğer durumda $K = \mathcal{K}(\mathbf{m}, \mathbf{c})$ ortak anahtar türetilir.

3.7 Örnek

HQC algoritmasının çalışma adımlarının doğruluğunu gerçeklemek için daha küçük parametreler seçerek bir örnek oluşturduk.

3.7.1 Parametreler

BCH $[n_1, k_1 = k, t_1]$ ve $\mathbb{1}_{n_2}$ $[n_2, 1, t_2]$ kodları olmak üzere,

- Kod uzunluğu: $n_1 = 2^4 - 1 = 15$, $n_2 = 3$ seçilsin. Bu durumda $n_1 n_2 = 45$ dan büyük en küçük primitif asal olan $n = 47$ olur.
- Kod boyutu: $k_1 = k = 7$.
- Hata düzeltme kapasitesi: $t = 2$.
- $\mathbf{x}$ ve $\mathbf{y}$ nin ağırlıkları: $w = 1$.
- $\mathbf{r}_1$ ve $\mathbf{r}_2$ nin ağırlıkları: $w_r = 2$.
- $\mathbf{e}$ nin ağırlığı: $w_e = 2$.

$\mathcal{C} = \text{BCH}\,[15, 7, 2] \otimes \mathbb{1}_3$ tensör çarpım kodu kullanılacaktır. Bu seçimler doğrultusunda $\mathcal{R} = \mathbb{F}_2[x] / \langle x^{47} - 1 \rangle$ bulunur ve bu kümenin elemanları katsayıları $\mathbb{F}_2$ cisminde olan derecesi 47'den küçük polinomlardır.

3.7.2 Anahtar üretimi

Parametreler kullanılarak pk açık anahtarı ve sk gizli anahtarı oluşturulur.

1. $\mathcal{R}$ üzerinde $n = 47$ uzunlukta rastgele $\mathbf{h} = 1 + x^2 + x^3 + x^{12} + x^{30}$ üretilir.
2. $\mathcal{C} = \text{BCH}\,[15, 7, 2] \otimes \mathbb{1}_3$ kodu için Örnek 12'den BCH $[15, 7, 2]$ kodunun üreteç polinomu $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ şeklindedir ve $\mathbb{1}_3$ kodunun üreteç matrisi $G_2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ dir.
3. $\mathcal{R}$ üzerinde $\text{wt}(\mathbf{x}) = \text{wt}(\mathbf{y}) = 1$ olacak şekilde rastgele $\mathbf{x} = x^2$ ve $\mathbf{y} = x^5$ elemanları üretilir ve uzunluğu $2n = 94$ olan $\mathsf{sk} = (\mathbf{x}, \mathbf{y}) = (x^2, x^5)$ gizli anahtarı elde edilir.

4. $n = 47$ uzunlukta olan

$$\begin{aligned}\mathbf{s} &= \mathbf{x} + \mathbf{hy} \\ &= x^2 + (1 + x^2 + x^3 + x^{12} + x^{30})(x^5) \\ &= x^2 + x^5 + x^7 + x^8 + x^{17} + x^{35}\end{aligned}$$

vektörü hesaplanır ve uzunluğu $2n = 94$ olan $\mathsf{pk} = (\mathbf{h}, \mathbf{s}) = (1 + x^2 + x^3 + x^{12} + x^{30}, x^2 + x^5 + x^7 + x^8 + x^{17} + x^{35})$ açık anahtarı oluşturulur.

5. $(\mathsf{pk}, \mathsf{sk})$ çıktısı verilir.

3.7.3 Kapsülleme

$\mathsf{pk} = (\mathbf{h}, \mathbf{s}) = (1 + x^2 + x^3 + x^{12} + x^{30}, x^2 + x^5 + x^7 + x^8 + x^{17} + x^{35})$ açık anahtarı ve $\mathcal{C} = \text{BCH}\,[15, 7, 2] \otimes \mathbb{1}_3$ tensör çarpım kodunun üreteç matrisi kullanılarak kapsülleme yapılır.

1. $\mathbb{F}_2$ üzerinde $k = 7$ uzunluğunda rastgele $\mathbf{m} = (1110110)$ seçilsin.

2. $\theta = \mathcal{G}(\mathbf{m}) = \mathcal{G}((1110110))$ rastgelelik değişkeni hesaplanır ve θ ya bağlı olarak bir sonraki adımda rastgele elemanlar üretilir.

3. $\mathcal{R}$ üzerinde $\text{wt}(\mathbf{e}) = w_e = 2$, $\text{wt}(\mathbf{r}_1) = \text{wt}(\mathbf{r}_2) = w_r = 2$ ve uzunlukları $n = 47$ olan rastgele $\mathbf{e} = x^2 + x^{12}$, $\mathbf{r}_1 = 1 + x^7$ ve $\mathbf{r}_2 = 1 + x^2$ elemanları verilsin.

4. Uzunluğu $2n = 94$ olan şifreli metin $\mathbf{c} = (\mathbf{u}, \mathbf{v})$ aşağıdaki gibi hesaplanır:

$$\begin{aligned}\mathbf{u} &= \mathbf{r}_1 + \mathbf{hr}_2 \\ &= 1 + x^7 + (1 + x^2 + x^3 + x^{12} + x^{30})(1 + x^2) \\ &= x^3 + x^4 + x^5 + x^7 + x^{12} + x^{14} + x^{30} + x^{32}.\end{aligned}$$

$\mathbf{v} = \mathbf{m}G + \mathbf{sr}_2 + \mathbf{e}$ hesaplanmadan önce Açıklama 3.1 kullanılarak $\mathbf{m}G = \tilde{\mathbf{m}} = (\tilde{\mathbf{m}}_{1,0}, \ldots, \tilde{\mathbf{m}}_{1,14})$ şeklinde hesaplanmalıdır. $\mathbf{m} = (1110110)$

kelimesi için G_1 üreteç matrisi (2.1) matrisi olmak üzere,

$$\begin{aligned}\mathbf{m}G_1 &= \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{1\times 15}.\end{aligned}$$

elde dilir. Yani $\mathbf{m}_1 = \mathbf{m}G_1 = (111000000100010) \in \mathbb{F}_2^{15}$ şeklinde bulunur. Şimdi de $\mathbf{m}_1 = (\mathbf{m}_{1,0}\mathbf{m}_{1,1}\ldots\mathbf{m}_{1,14})$ in her $\mathbf{m}_{1,i}$ koordinatı $\mathbb{1}_{n_2}$ tekrarlı kodu ile aşağıdaki gibi kodlanır:

$$\begin{aligned}
&\tilde{\mathbf{m}}_{1,0} = \mathbf{m}_{1,0}G_2 = [1][111] = [111] \quad \tilde{\mathbf{m}}_{1,8} = \mathbf{m}_{1,8}G_2 = [0][111] = [000] \\
&\tilde{\mathbf{m}}_{1,1} = \mathbf{m}_{1,1}G_2 = [1][111] = [111] \quad \tilde{\mathbf{m}}_{1,9} = \mathbf{m}_{1,9}G_2 = [1][111] = [111] \\
&\tilde{\mathbf{m}}_{1,2} = \mathbf{m}_{1,2}G_2 = [1][111] = [111] \quad \tilde{\mathbf{m}}_{1,10} = \mathbf{m}_{1,10}G_2 = [0][111] = [000] \\
&\tilde{\mathbf{m}}_{1,3} = \mathbf{m}_{1,3}G_2 = [0][111] = [000] \quad \tilde{\mathbf{m}}_{1,11} = \mathbf{m}_{1,11}G_2 = [0][111] = [000] \\
&\tilde{\mathbf{m}}_{1,4} = \mathbf{m}_{1,4}G_2 = [0][111] = [000] \quad \tilde{\mathbf{m}}_{1,12} = \mathbf{m}_{1,12}G_2 = [0][111] = [000] \\
&\tilde{\mathbf{m}}_{1,5} = \mathbf{m}_{1,5}G_2 = [0][111] = [000] \quad \tilde{\mathbf{m}}_{1,13} = \mathbf{m}_{1,13}G_2 = [1][111] = [111] \\
&\tilde{\mathbf{m}}_{1,6} = \mathbf{m}_{1,6}G_2 = [0][111] = [000] \quad \tilde{\mathbf{m}}_{1,14} = \mathbf{m}_{1,14}G_2 = [0][111] = [000]. \\
&\tilde{\mathbf{m}}_{1,7} = \mathbf{m}_{1,7}G_2 = [0][111] = [000]
\end{aligned}$$

Bu durumda $\mathbf{m}G = (111111111000000000000000000011100000000000111000) \in \mathbb{F}_2^{45}$ bulunur. $\mathbf{v}$ ise,

$$\begin{aligned}\mathbf{v} &= \mathbf{m}G + \mathbf{s}\mathbf{r}_2 + \mathbf{e} \\ &= 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + x^{27} + x^{28} + x^{29} + x^{39} \\ &\quad + x^{40} + x^{41} + (x^2 + x^5 + x^7 + x^8 + x^{17} + x^{35})(1 + x^2) + x^2 + x^{12} \\ &= 1 + x + x^2 + x^3 + x^6 + x^7 + x^9 + x^{10} + x^{12} + x^{17} + x^{19} + x^{27} + x^{28} \\ &\quad + x^{29} + x^{35} + x^{37} + x^{39} + x^{40} + x^{41}\end{aligned}$$

şeklinde hesaplanır.

5. $K = \mathcal{K}(\mathbf{m}, \mathbf{c})$ simetrik anahtarı türetilir.

6. $\mathbf{d} = \mathcal{H}(\mathbf{m}) = \mathcal{H}((1110110))$ hesaplanır.

7. $(\mathbf{c}, \mathbf{d})$ karşı tarafa gönderilir.

3.7.4 Kapsülden çıkarma

Alınan $(\mathbf{c}, \mathbf{d})$ ve gizli anahtar $\mathsf{sk} = (\mathbf{x}, \mathbf{y}) = (x^2, x^5)$ kullanılır.

1. $\mathcal{C}$.Decode$(\mathbf{v} - \mathbf{uy})$ ile $\mathbf{m}'$ bulunur.

$$\begin{aligned}\mathbf{v} - \mathbf{uy} &= 1 + x + x^2 + x^3 + x^6 + x^7 + x^9 + x^{10} + x^{12} + x^{17} + x^{19} \\ &\quad + x^{27} + x^{28} + x^{29} + x^{35} + x^{37} + x^{39} + x^{40} + x^{41} \\ &\quad + (x^3 + x^4 + x^5 + x^7 + x^{12} + x^{14} + x^{30} + x^{32})(x^5) \\ &= 1 + x + x^2 + x^3 + x^6 + x^7 + x^8 + x^{27} + x^{28} + x^{29} + x^{39} + x^{40} + x^{41}\end{aligned}$$

şeklinde hsaplanır. Alınan mesajın uzunluğu 47 olduğundan kod ailesi çözüm yapamaz. Bu nedenle sondan 2 birim çıkarılarak uzunluğu $n_1 n_2 = 45$ olan $(111100111000000000000000000111000000000111000)$ vektörüne kod çözme algoritmaları uygulanacaktır. Algoritma 3.1 kullanılarak tekrarlı kod için kod çözümü,

$$\begin{array}{lll} 111 \Rightarrow 1 & 000 \Rightarrow 0 & 000 \Rightarrow 0 \\ 100 \Rightarrow 0 & 000 \Rightarrow 0 & 000 \Rightarrow 0 \\ 111 \Rightarrow 1 & 000 \Rightarrow 0 & 000 \Rightarrow 0 \\ 000 \Rightarrow 0 & 000 \Rightarrow 0 & 111 \Rightarrow 1 \\ 000 \Rightarrow 0 & 111 \Rightarrow 1 & 000 \Rightarrow 0 \end{array}$$

şeklinde yapılır. Bu durumda BCH kodu için ulaşan vektör $\mathbf{r} = (101000000100010)$ dolayısıyla $r(x) = 1 + x^2 + x^9 + x^{13}$ olur. BCH için kod çözümü ise aşağıdaki gibi yapılır.

Sendromların kümesi $S = (S_1, S_2, S_3, S_4)$ ve α primitif eleman olmak üzere,

$$\begin{aligned} S_1 &= r(\alpha) = 1 + \alpha^2 + \alpha^9 + \alpha^{13} = \alpha \\ S_2 &= r(\alpha^2) = 1 + (\alpha^2)^2 + (\alpha^2)^9 + (\alpha^2)^{13} = \alpha^2 \\ S_3 &= r(\alpha^3) = 1 + (\alpha^3)^2 + (\alpha^3)^9 + (\alpha^3)^{13} = \alpha^3 \\ S_4 &= r(\alpha^4) = 1 + (\alpha^4)^2 + (\alpha^4)^9 + (\alpha^4)^{13} = \alpha^4 \end{aligned}$$

şeklinde bulunur.

$$S = (\alpha, \alpha^2, \alpha^3, \alpha^4)$$

sendromları kullanılarak,

$$S(x) = \alpha + \alpha^2 x + \alpha^3 x^2 + \alpha^4 x^3$$

sendrom polinomu elde edilir. $S(x)$ ve $x^{2t} = x^4$ ifadelerine Genişletilmiş Öklid algoritması uygulanarak hata konum polinomu $\sigma(x)$ bulunur.

i	$s_i(x)$	$t_i(x)$	$r_i(x)$	$q_i(x)$
0	1	0	x^4	–
1	0	1	$\alpha + \alpha^2 x + \alpha^3 x^2 + \alpha^4 x^3$	–
2	1	$\alpha^{10} + \alpha^{11} x$	α^{11}	$\alpha^{10} + \alpha^{11} x$

$\deg(r_2) \leq t$ sağlandığından algoritma durur. Bu durumda hata konum polinomu,

$$\sigma(x) = \alpha^{10} + \alpha^{11} x$$

şeklinde elde edilir. Bu polinomun kökü α^{14}, tersi ise α bulunur. O halde hata $e(x) = x$ bulunmuş olur. Böylece $c(x) = r(x) - e(x) = 1 + x + x^2 + x^9 + x^{13}$ kod kelimesine ulaşılır ve buna karşılık gelen mesaj $\mathbf{m}' = (1110110)$ bulunur (Kapsülleme algoritmasının 4. adımına bakılarak kontrol edilebilir).

2. $\theta' = \mathcal{G}((1110110))$ rastgelelik değişkeni hesaplanır ve θ' ya bağlı olarak bir sonraki adımda rastgele elemanlar üretilir. Dikkat edilirse $\mathbf{m}' = \mathbf{m}$ olduğundan $\mathcal{G}$ özet fonksiyonu aynı çıktıları verir ve $\theta' = \theta$ olur. Bu durumda 3. ve 4.

adımlarda kapsülleme algoritmasının 2. adımındaki elemanlarla aynı rastgele elemanlar üretilir. Bu nedenle aynı $\mathbf{c}$ şifreli metni elde edilir.

3. $\mathcal{R}$ üzerinde $\mathrm{wt}(\mathbf{e}) = w_e$ ve $\mathrm{wt}(\mathbf{r}_1) = \mathrm{wt}(\mathbf{r}_2) = w_r$ olacak şekilde rastgele $\mathbf{e}$, $\mathbf{r}_1$ ve $\mathbf{r}_2$ üretilir.

4. Şifreli metin $\mathbf{c}' = (\mathbf{u}, \mathbf{v}) = (\mathbf{r}_1 + \mathbf{h}\mathbf{r}_2, \mathbf{m}'G + \mathbf{s}\mathbf{r}_2 + \mathbf{e})$ hesaplanır.

5. $\mathbf{c} = \mathbf{c}'$ ve $\mathbf{d} = \mathcal{H}(\mathbf{m}')$ olur.

6. $K = \mathcal{K}(\mathbf{m}, \mathbf{c})$ ortak anahtar elde edilir.

4. RQC KUANTUM SONRASI ALGORİTMASI

Bu bölümde ise HQC algoritmasını geliştiren araştırmacıların bir diğer algoritması olan RQC hakkında bilgiler verilecektir. RQC kod tabanlı açık anahtarlı bir kripto sistemdir. Detaylı bilgi için [20] incelenebilir.

Bu algoritmanın kullandığı kod ailesi ve özet fonksiyonları sayısı dışında şifreleme-şifre çözme ve kapsülleme-kapsülden çıkarma algoritmaları HQC ile neredeyse aynıdır. Kodların yapısındaki sendromla kod çözmenin dayanıklılığı sayesinde IND-CPA güvenli olduğunu kanıtlamıştır. Aynı zamanda IND-CCA2 güvenli ve iyi bir verimliliğe sahiptir.

RQC iki çeşit kod kullanır: $G \in \mathbb{F}_{q^m}^{k\times n}$ matrisi tarafından üretilen, etkili bir kod çözme algoritması olan $\mathcal{G}_g$.Decode$(\cdot)$ ile t-hata düzeltebilen $\mathcal{G}_g[n,k,m]$ Gabidulin kodu ve eşlik-denetim matrisi $(\mathbf{1},\mathbf{h})$ olan rastgele $[2n,n]$-ideal kodu. İlk turda RQC algoritması bu durumdan farklı olarak ideal kodlar yerine çift çevrimsel kullanılmasını söylüyordu. Ancak ikinci turda bu durum güncelenerek DFR sıfıra düşürülmüştür.

Kapsülleme ve kapsülden çıkarma algoritmalarında kullanılan $\mathcal{G}$, $\mathcal{H}$ ve $\mathcal{K}$ fonksiyonları özet fonksiyonlardır. NIST'in önerisine göre genellikle SHA512 önerilir. Ancak $\mathcal{G}=\mathcal{H}$ seçilmesi önerilmiyor çünkü ataklarla $\mathbf{s}$ hakkında bilgi sahibi olunmasına sebebiyet verebilir. Bu nedenle RQC algoritması için $\mathcal{G}$ için sözde rastgele fonksiyon (örneğin AES-based seed expander) ve $\mathcal{H}$ için SHA512 kullanılması önerilmektedir.

Ayrıca algoritmalarda,

$$\begin{aligned} S_w^n(\mathbb{F}_{q^m}) &= \left\{\mathbf{x}\in\mathbb{F}_{q^m}^n \mid \text{boy}(\text{Supp}(\mathbf{x}))=w\right\} \\ S_{1,w}^n(\mathbb{F}_{q^m}) &= \left\{\mathbf{x}\in\mathbb{F}_{q^m}^n \mid \text{boy}(\text{Supp}(\mathbf{x}))=w, 1\in\text{Supp}(\mathbf{x})\right\} \end{aligned}$$

notasyonları kullanılacaktır.

Yani $S_w^n(\mathbb{F}_{q^m})$, $\mathbb{F}_{q^m}$ üzerinde uzunluğu n ve rank ağırlığı w olan vektörlerin kümesi, $S_{1,w}^n(\mathbb{F}_{q^m})$ ise $\mathbb{F}_{q^m}$ üzerinde uzunluğu n, rank ağırlığı w olan ve desteği 1 içeren vektörlerin kümesidir.

4.1 Parametreler

- n: kod uzunluğu.
- k: kod boyutu.
- t: hata düzeltme kapasitesi.
- w: $\mathbf{x}$ ve $\mathbf{y}$ nin ağırlıkları.
- w_r: $\mathbf{r}_1$ ve $\mathbf{r}_2$ nin ağırlıkları.
- $p(x)$: $p(x) \in \mathbb{F}_q[x]$ derecesi n olan indirgenemez polinom.

4.2 Anahtar Üretimi

Parametreler kullanılarak pk açık anahtarı ve sk gizli anahtarı oluşturulur.

1. $\mathbb{F}_{q^m}^n$ vektör uzayından rastgele $\mathbf{h}$ seçilir.
2. $\mathbf{g} \in S_n^n(\mathbb{F}_{q^m})$ rastgele vektörü seçilir.
3. $\mathbf{x} \in S_{1,w}^n(\mathbb{F}_{q^m})$ ve $\mathbf{y} \in S_{1,w}^n(\mathbb{F}_{q^m})$ rastgele vektörleri üretilir ve uzunluğu $2n$ olan $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$ gizli anahtarı elde edilir.
4. $\mathcal{G}_g(n,k,m)$ kodunun $k \times n$ boyutlu $G \in \mathbb{F}_{q^m}^{k \times n}$ üreteç matrisi oluşturulur.
5. Uzunluğu $3n$ olan $\mathsf{pk} = (\mathbf{g}, \mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{hy} \mod p(x))$ açık anahtarı bulunur.
6. $(\mathsf{pk}, \mathsf{sk})$ çıktısı verilir.

4.3 Şifreleme

$\mathsf{pk} = (\mathbf{g}, \mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{hy} \mod p(x))$ açık anahtarı, G üreteç matrisi ve diğer rastgele değerler kullanılarak bir $\mathbf{m}$ mesajı $\mathbf{c}$ şifreli metnine dönüştürülür ve karşı tarafa gönderilir.

1. $\mathbf{e} \in S^n_{w_r}(\mathbb{F}_{q^m})$, $\mathbf{r}_1 \in S^n_{w_r}(\mathbb{F}_{q^m})$ ve $\mathbf{r}_2 \in S^n_{w_r}(\mathbb{F}_{q^m})$ rastgele elemanları üretilir.
2. Uzunlukları n olan $\mathbf{u} = \mathbf{r}_1 + \mathbf{hr}_2 \mod p(x)$ ve $\mathbf{v} = \mathbf{m}G + \mathbf{sr}_2 + \mathbf{e} \mod p(x)$ hesaplanır.
3. $2n$ uzunluğunda $\mathbf{c} = (\mathbf{u}, \mathbf{v})$ çıktısı verir.

4.4 Şifre Çözme

Alınan şifreli metin $\mathbf{c} = (\mathbf{u}, \mathbf{v})$ ve gizli anahtar $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$ kullanılarak $\mathbf{m}$ mesajına ulaşılır.
$\mathcal{G}_g$.Decode$(\mathbf{v} - \mathbf{uy} \mod p(x))$ hesaplanır.

$$\begin{aligned}
\mathbf{v} - \mathbf{uy} &= \mathbf{m}G + \mathbf{sr}_2 + \mathbf{e} - (\mathbf{r}_1 + \mathbf{hr}_2)\mathbf{y} \\
&= \mathbf{m}G + (\mathbf{x} + \mathbf{hy})\mathbf{r}_2 + \mathbf{e} - (\mathbf{r}_1 + \mathbf{hr}_2)\mathbf{y} \\
&= \mathbf{m}G + \mathbf{xr}_2 + \mathbf{hyr}_2 + \mathbf{e} - \mathbf{r}_1\mathbf{y} - \mathbf{hr}_2\mathbf{y} \\
&= \mathbf{m}G + \mathbf{xr}_2 + \mathbf{e} - \mathbf{r}_1\mathbf{y}
\end{aligned}$$

olduğundan $\mathbf{e}' = \mathbf{xr}_2 - \mathbf{r}_1\mathbf{y} + \mathbf{e}$ hata vektörü olmak üzere $\mathrm{wt}(\mathbf{e}') \leq t$ olursa kod çözümü yapılarak $\mathbf{m}$ mesajı kurtarılır.

4.5 Kapsülleme

$\mathsf{pk} = (\mathbf{g}, \mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{hy} \mod p(x))$ açık anahtarı ve G üreteç matrisi kullanılarak kapsülleme yapılır.

1. $\mathbb{F}_{q^m}$ cismi üzerinde k uzunlukta rastgele $\mathbf{m}$ üretilir.

2. $\theta = \mathcal{G}(\mathbf{m})$ rastgelelik değişkeni $\mathbf{m}$ mesajının özet değeri olarak hesaplanır. Bu θ ya bağlı olarak bir sonraki adımda rastgele elemanlar üretilir.

3. $\mathbf{e} \in S^n_{w_r}(\mathbb{F}_{q^m})$, $\mathbf{r}_1 \in S^n_{w_r}(\mathbb{F}_{q^m})$ ve $\mathbf{r}_2 \in S^n_{w_r}(\mathbb{F}_{q^m})$ rastgele elemanları üretilir.

4. $2n$ uzunluğunda $\mathbf{c} = (\mathbf{u}, \mathbf{v})$ şifreli metni uzunlukları n olan $\mathbf{u} = \mathbf{r}_1 + \mathbf{h}\mathbf{r}_2 \mod p(x)$ ve $\mathbf{v} = \mathbf{m}G + \mathbf{s}\mathbf{r}_2 + \mathbf{e} \mod p(x)$ hesaplanarak oluşturulur.

5. $K = \mathcal{K}(\mathbf{m}, \mathbf{c})$ simetrik anahtarı türetilir.

6. $\mathbf{d} = \mathcal{H}(\mathbf{m})$ hesaplanır ve $(\mathbf{c}, \mathbf{d})$ karşı tarafa gönderilir.

4.6 Kapsülden Çıkarma

Alınan $(\mathbf{c}, \mathbf{d})$ ve gizli anahtar $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$ kullanılır.

1. $\mathcal{G}_g$.Decode$(\mathbf{v} - \mathbf{u}\mathbf{y} \mod p(x))$ hesaplanır.

$$\begin{aligned}
\mathbf{v} - \mathbf{u}\mathbf{y} \mod p(x) &= \mathbf{m}G + \mathbf{s}\mathbf{r}_2 + \mathbf{e} - (\mathbf{r}_1 + \mathbf{h}\mathbf{r}_2)\mathbf{y} \mod p(x) \\
&= \mathbf{m}G + (\mathbf{x} + \mathbf{h}\mathbf{y})\mathbf{r}_2 + \mathbf{e} - (\mathbf{r}_1 + \mathbf{h}\mathbf{r}_2)\mathbf{y} \mod p(x) \\
&= \mathbf{m}G + \mathbf{x}\mathbf{r}_2 + \mathbf{h}\mathbf{y}\mathbf{r}_2 + \mathbf{e} - \mathbf{r}_1\mathbf{y} - \mathbf{h}\mathbf{r}_2\mathbf{y} \mod p(x) \\
&= \mathbf{m}G + \mathbf{x}\mathbf{r}_2 + \mathbf{e} - \mathbf{r}_1\mathbf{y} \mod p(x)
\end{aligned}$$

   olduğundan $\mathbf{e}' = \mathbf{x}\mathbf{r}_2 - \mathbf{r}_1\mathbf{y} + \mathbf{e} \mod p(x)$ hata vektörü olmak üzere $\mathrm{wt}(\mathbf{e}') \leq t$ olursa kod çözümü yapılarak $\mathbf{m}'$ mesajı kurtarılır.

2. $\theta' = \mathcal{G}(\mathbf{m}')$ rastgelelik değişkeni $\mathbf{m}'$ mesajının özet değeri olarak hesaplanır. Bu θ' ya bağlı olarak bir sonraki adımda rastgele elemanlar üretilir.

3. $\mathbf{e} \in S^n_{w_r}(\mathbb{F}_{q^m})$, $\mathbf{r}_1 \in S^n_{w_r}(\mathbb{F}_{q^m})$ ve $\mathbf{r}_2 \in S^n_{w_r}(\mathbb{F}_{q^m})$ rastgele elemanları üretilir.

4. $2n$ uzunluğunda şifreli metin $\mathbf{c}' = (\mathbf{u}, \mathbf{v}) = (\mathbf{r}_1 + \mathbf{h}\mathbf{r}_2 \mod p(x), \mathbf{m}'G + \mathbf{s}\mathbf{r}_2 + \mathbf{e} \mod p(x))$ hesaplanır.

5. Eğer $\mathbf{c} \neq \mathbf{c}'$ ya da $\mathbf{d} \neq \mathcal{H}(\mathbf{m}')$ ise hata verir.

6. Diğer durumda $K = \mathcal{K}(\mathbf{m}, \mathbf{c})$ ortak anahtar türetilir.

5. BIKE KUANTUM SONRASI ALGORİTMASI

Bu bölümde QC-MDPC kodunu kullanarak, bit-çevirme algoritmalarıyla kod çözebilen anahtar kapsülleme algoritmalarından oluşan BIKE hakkında bilgiler verilecektir. Detaylı bilgi için [2] incelenebilir.

BIKE algoritmasımın ilk turda BIKE-1, BIKE-2 ve BIKE-3 olmak üzere üç farklı çeşidi vardı. Ancak ikinci turda BIKE-1-CCA, BIKE-2-CCA ve BIKE-3-CCA algoritmaları da eklenmiştir, toplam 6 çeşidi vardır. Altısı da McEliece ya da Niederreiter sistemlerini baz alırlar ancak her biri önemli farklılıklara sahiptir. HQC algoritmasına benzer şekilde burada da $\mathcal{R} = \mathbb{F}_2[x]/\langle x^r - 1\rangle$ devirli polinom halkası kullanılmaktadır.

5.1 Kod Çözme Algoritmaları

MDPC kodları için başarılı olan birçok kod çözme algoritması mevcuttur. Bu algoritmalardan bit çevirme algoritması sadeliğinden dolayı oldukça ilgi çekicidir. Ek 2'de 4 farklı kod çözme algoritması verilmiştir. Algoritma 1'de eşik değeri τ nun nasıl seçileceği açıkça belirtilmemiştir. Kod çözme algoritmalarında sabitlenmiş eşik $\tau = \frac{1}{2}$ kabul edilmiştir.

Sendrom ile kod çözmenin bir çeşidi olan gürültülü sendrom ile kod çözme algoritmasının amacı verilen H ve $\mathbf{s}$ için $\mathbf{s} - \mathbf{e}H^T$ ve $\mathbf{e}$ vektörleri küçük ağırlıklara sahip olacak şekilde $\mathbf{e} \in \mathbb{F}_2^n$ hata vektörü bulmaktır. Bit çevirme algoritması gürültülü sendrom için iki durum değiştirilerek kolaylıkla uyarlanabilir. İlki durdurma koşulu, $\mathbf{s} - \mathbf{e}H^T$ eşitliğinin sıfıra eşit olmasına gerek yok, sadece ağırlığının küçük olması yeterlidir. İkinci olarak durdurma koşulundaki ağırlığın ölçülmesi gerekli olduğundan hedef ağırlık u olarak belirlenir. Algoritma 2'de $u = 0$ olduğunda genişletilmiş bit çevirme algoritması ile bit çevirme algoritması aynı görevi yapar.

MDPC kodları için kullanılan bu kod çözme algoritmaları QC-MDPC içinde kullanılabilir. Yarı-devirlilik kod çözme algoritmasını değiştirmez. Kod çözme algoritmaları $[2r,r]$-QC-MDPC kodları için kullanılacaktır. Bu algoritmalar girdi olarak $(\mathbf{s},\mathbf{h}_0,\mathbf{h}_1) \in \mathcal{R}^3$ ve u tam sayısını alarak $\mathrm{wt}(\mathbf{e}_0\mathbf{h}_0+\mathbf{e}_1\mathbf{h}_1+\mathbf{s}) \leq u$ olacak şekilde $(\mathbf{e}_0,\mathbf{e}_1) \in \mathcal{R}^2$ çıktısı verir.

Verilen BIKE parametleri r, w, t ve çeşidi için anahtar özellikleri kod çözme zamanı ve kod çözme hata oranı olacaktır. $\mathcal{R} = \mathbb{F}_2[x]/\langle x^r-1\rangle$ polinom halkası olmak üzere kod çözme algoritmalarının girdisi $(H,\mathbf{s},u)$ aşağıdaki gibi seçilir:

- H eşlik-denetim matrisi 2 indeksli blok-çevrimsel matristir. $H = (\mathbf{h}_0^T\mathbf{h}_1^T) \in \mathcal{R}^{1\times 2}$ öyle ki $\mathrm{wt}(\mathbf{h}_0) = \mathrm{wt}(\mathbf{h}_1) = \frac{w}{2}$.
- u tam sayısı ya 0 (gürültüsüz sendrom ile kod çözme, BIKE-1 ve BIKE-2) ya da $\frac{t}{2}$ (gürültülü sendrom ile kod çözme, BIKE-3).
- $\mathrm{wt}(\mathbf{e}') = u$ ve $\mathrm{wt}(\mathbf{e}_0)+\mathrm{wt}(\mathbf{e}_1) = t$ olcak şekilde bazı $(\mathbf{e}',\mathbf{e}_0,\mathbf{e}_1) \in \mathcal{R}^3$ için $\mathbf{s} = \mathbf{e}'+\mathbf{e}_0\mathbf{h}_0+\mathbf{e}_1\mathbf{h}_1$ sendromu.

Her parametre kümesi ve BIKE çeşidi için kod çözme algoritmaları $\mathbf{h}_0,\mathbf{h}_1,\mathbf{e}',\mathbf{e}_0,\mathbf{e}_1$ ifadelerini girdi olarak alır.

BIKE algoritmasının IND-CCA çeşitlerinde için EK 2'de tanımlanan Algoritma 3, IND-CPA çeşitlerinde için EK 2'de tanımlanan Algoritma 4 kullanılmıştır. Algoritma 3 durduğunda uygun bir hata vektörü verir ancak durmayabilir. Pratikte maksimum çaışma zamanına ulaştığında algoritma hata sonucu vererek durur. Verilen BIKE parametreleri r, w, t için $n = 2r$ ve $k = r$ alınır.

5.2 IND-CCA Çeşitleri

Bu bölümde IND-CCA güvenli olmayı baraşan BIKE çeşitleri BIKE-1, BIKE-2 ve BIKE-3 açıklanacaktır. Bu algoritmalar geçici anahtarlar kullanırlar yani her anahtar değişimi için yeni bir anahtar çifti üretilir. Böylelikle biri anahtarı elinde geçirse dahi

bir sonraki değişimde işine yaramayacağı için güvenlik sağlanmış olur. ℓ_K istenilen simetrik anahtar uzunluğu olmak üzere $\mathbf{K} : \{0,1\}^n \to \{0,1\}^{\ell_K}$ özet fonksiyondur.

5.2.1 Parametreler

BIKE algoritmalarında kullanılacak olan kod ailesi $n = n_0 r$, $k = k_0 r$ olmak üzere $[n,k,w]$-QC-MDPC kodudur.

- n: kod uzunluğu ve $n = 2r$.
- r: asal tam sayı öyle ki $\dfrac{x^r - 1}{x - 1} \in \mathbb{F}_2$ indirgenemez bir polinom.
- w: $[n,k,w]$-QC-MDPC kodunun eşlik-denetim matrisinin her bir satırının ağırlığı.
- t: $[n,k,w]$-QC-MDPC kodunun hata düzeltme kapasitesi.

5.2.2 BIKE-1

Burada hızlı anahtar üretimi için McEliece algoritmasının bir versiyonu kullanılmıştır. İlk olarak, QC-MDPC McEliece [21] algoritmasının aksine, standart formu elde etmek için gizli devirli blokların tersi hesaplanıp tüm gizli matrisle çarpılmaz. Onun yerine gizli kodun yapısını saklamak için seyrek gizli matris, rastgele yoğun devirli blokla çarpılmaktadır. İkincisi de McEliece şifreleme sistemi değiştirilerek kod kelimesi yerine mesaj hata vektörü ile iletilmiştir.

5.2.2.1 Anahtar üretimi

Parametreler kullanılarak seyrek gizli anahtar $(\mathbf{h}_0, \mathbf{h}_1)$ ve yoğun açık anahtar $(\mathbf{f}_0, \mathbf{f}_1)$ çıktısı verilir.

1. $\mathcal{R}$ üzerinde $\mathrm{wt}(\mathbf{h}_0) = \mathrm{wt}(\mathbf{h}_1) = \frac{w}{2}$ olacak şekilde ağırlıkları tek olan $\mathbf{h}_0$ ve $\mathbf{h}_1$ rastgele polinomları seçilir.

2. $\mathcal{R}$ üzerinde $\text{wt}(\mathbf{g}) \approx \frac{r}{2}$ olacak şekilde ağırlığı tek olan $\mathbf{g}$ rastgele elemanı seçilir.

3. $(\mathbf{f}_0, \mathbf{f}_1) = (\mathbf{gh}_1, \mathbf{gh}_0)$ hesaplanır.

5.2.2.2 Kapsülleme

$(\mathbf{f}_0, \mathbf{f}_1)$ yoğun açık anahtarı kullanılarak kapsüllenmiş anahtar K ve şifreli metin $\mathbf{c}$ çıktısı verilir.

1. $(\mathbf{e}_0, \mathbf{e}_1) \in \mathcal{R}^2$ öyle ki $\text{wt}(\mathbf{e}_0) + \text{wt}(\mathbf{e}_1) = t$ hata vektörü seçilir.

2. $\mathcal{R}$ üzerinde $\mathbf{m}$ polinomu rastgele üretilir.

3. $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1) = (\mathbf{mf}_0 + \mathbf{e}_0, \mathbf{mf}_1 + \mathbf{e}_1)$ hesaplanır.

4. $K = \mathbf{K}(\mathbf{e}_0, \mathbf{e}_1)$ anahtarı hesaplanır.

5.2.2.3 Kapsülden çıkarma

Alınan $\mathbf{c}$ şifreli metni, $(\mathbf{h}_0, \mathbf{h}_1)$ seyrek gizli anahtarı kullanılarak kapsülden çıkarılmış K anahtarına ulaşılır.

1. $\mathbf{s} = \mathbf{c}_0\mathbf{h}_0 + \mathbf{c}_1\mathbf{h}_1$ sendromu hesaplanır.

2. $(\mathbf{e}'_0, \mathbf{e}'_1)$ hata vektörünü kurtarmak için $\mathbf{s}$ (gürültüsüz) sendromu ile kod çözme algoritması denenir.

3. Eğer $\text{wt}((\mathbf{e}'_0, \mathbf{e}'_1)) \neq t$ ya da kod çözme başarısız olursa $\perp$ çıktısı verir ve durur.

4. Kod çözme başarılı olursa $K = \mathbf{K}(\mathbf{e}'_0, \mathbf{e}'_1)$ anahtarı hesaplanır.

5.2.3 BIKE-2

BIKE algoritmasının bu çeşidinde ise standart eşlik-denetim matrisi ile Niederreiter sistemi baz alınmıştır. Temel avantajı ise sisteme dahil olan tüm elemanların r

uzunluğunda tekil blok gerektirmesidir. Bu da çok kısa bir formülasyona yol açar. Bu bağlamda tersini alma gerektiren anahtar üretme algoritmaları şifrelemeden önemli ölçüde daha yavaştır. Batch anahtar üretim algoritmasını kullanmak bu duruma bir çözüm olabilir [2].

5.2.3.1 Anahtar üretimi

Parametreler kullanılarak seyrek gizli anahtar $(\mathbf{h}_0, \mathbf{h}_1)$ ve yoğun açık anahtar $(\mathbf{f}_0, \mathbf{f}_1)$ çıktısı verilir.

1. $\mathcal{R}$ üzerinde $\mathrm{wt}(\mathbf{h}_0) = \mathrm{wt}(\mathbf{h}_1) = \frac{w}{2}$ olacak şekilde ağırlıkları tek olan $\mathbf{h}_0$ ve $\mathbf{h}_1$ rastgele polinomları seçilir.
2. $\mathbf{h} = \mathbf{h}_1 \mathbf{h}_0^{-1}$ hesaplanır.

5.2.3.2 Kapsülleme

$(\mathbf{f}_0, \mathbf{f}_1)$ yoğun açık anahtarı kullanılarak kapsüllenmiş anahtar K ve şifreli metin $\mathbf{c}$ çıktısı verilir.

1. $(\mathbf{e}_0, \mathbf{e}_1) \in \mathcal{R}^2$ öyle ki $\mathrm{wt}(\mathbf{e}_0) + \mathrm{wt}(\mathbf{e}_1) = t$ hata vektörü seçilir.
2. $\mathbf{c} = \mathbf{e}_0 + \mathbf{e}_1 h$ hesaplanır.
3. $K = \mathbf{K}(\mathbf{e}_0, \mathbf{e}_1)$ anahtarı hesaplanır.

5.2.3.3 Kapsülden çıkarma

Alınan $\mathbf{c}$ şifreli metni, $(\mathbf{h}_0, \mathbf{h}_1)$ seyrek gizli anahtarı kullanılarak kapsülden çıkarılmış K anahtarına ulaşılır.

1. $\mathbf{s} = \mathbf{c}\mathbf{h}_0$ sendromu hesaplanır.

2. $(\mathbf{e}'_0, \mathbf{e}'_1)$ hata vektörünü kurtarmak için $\mathbf{s}$ (gürültüsüz) sendromu ile kod çözme algoritması denenir.

3. Eğer $\text{wt}((\mathbf{e}'_0, \mathbf{e}'_1)) \neq t$ ya da kod çözme başarısız olursa $\perp$ çıktısı verir ve durur.

4. Kod çözme başarılı olursa $K = \mathbf{K}(\mathbf{e}'_0, \mathbf{e}'_1)$ anahtarı hesaplanır.

5.2.4 BIKE-3

BIKE-3 ise Ouroboros [9] çalışmasını takip etmektedir. BIKE-1 algoritmasına bakıldığında hızlı ve tersini alma içermeyen anahtar üretimi yapar ve açık anahtar iki blok içerir. Aradaki temel farklılık kapsülleme algoritmasının gürültülü sendrom ile kod çözümü yapmasıdır. Bu noktada BIKE-1 ve BIKE-2 den ayrılmaktadır.

5.2.4.1 Anahtar üretimi

Parametreler kullanılarak seyrek gizli anahtar $(\mathbf{h}_0, \mathbf{h}_1)$ ve yoğun açık anahtar $(\mathbf{f}_0, \mathbf{f}_1)$ çıktısı verilir.

1. $\mathcal{R}$ üzerinde $\text{wt}(\mathbf{h}_0) = \text{wt}(\mathbf{h}_1) = \frac{w}{2}$ olacak şekilde ağırlıkları tek olan $\mathbf{h}_0$ ve $\mathbf{h}_1$ rastgele polinomları seçilir.

2. $\mathcal{R}$ üzerinde $\text{wt}(\mathbf{g}) \approx \frac{r}{2}$ olacak şekilde ağırlığı tek olan $\mathbf{g}$ rastgele elemanı seçilir.

3. $(\mathbf{f}_0, \mathbf{f}_1) = (\mathbf{h}_1 + \mathbf{g}\mathbf{h}_0, \mathbf{g})$ hesaplanır.

5.2.4.2 Kapsülleme

$(\mathbf{f}_0, \mathbf{f}_1)$ yoğun açık anahtarı kullanılarak kapsüllenmiş anahtar K ve şifreli metin $\mathbf{c}$ çıktısı verilir.

1. $(\mathbf{e}, \mathbf{e}_0, \mathbf{e}_1) \in \mathcal{R}^3$ öyle ki $\text{wt}(\mathbf{e}) = \frac{t}{2}$ ve $\text{wt}(\mathbf{e}_0) + \text{wt}(\mathbf{e}_1) = t$ hata vektörü seçilir.

2. $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1) = (\mathbf{e} + \mathbf{e}_1\mathbf{f}_0, \mathbf{e}_0 + \mathbf{e}_1\mathbf{f}_1)$ hesaplanır.

3. $K = \mathbf{K}(\mathbf{e}_0, \mathbf{e}_1)$ anahtarı hesaplanır.

5.2.4.3 Kapsülden çıkarma

Alınan $\mathbf{c}$ şifreli metni, $(\mathbf{h}_0, \mathbf{h}_1)$ seyrek gizli anahtarı kullanılarak kapsülden çıkarılmış K anahtarına ulaşılır.

1. $\mathbf{s} = \mathbf{c}_0 + \mathbf{c}_1\mathbf{h}_0$ sendromu hesaplanır.
2. $(\mathbf{e}'_0, \mathbf{e}'_1)$ hata vektörünü kurtarmak için $\mathbf{s}$ (en çok $\frac{t}{2}$ gürültü) sendromu ile kod çözme algoritması denenir.
3. Eğer $\mathrm{wt}((\mathbf{e}'_0, \mathbf{e}'_1)) \neq t$ ya da kod çözme başarısız olursa $\perp$ çıktısı verir ve durur.
4. Kod çözme başarılı olursa $K = \mathbf{K}(\mathbf{e}'_0, \mathbf{e}'_1)$ anahtarı hesaplanır.

5.3 IND-CCA Çeşitleri

BIKE algoritmasının bu versiyonu ise statik anahtarları kullanmak yani birçok anahtar değişimini aynı anahtar çiftiyle yapabilmesi için tasarlanmıştır. Bu durum istenen güvenlik seviyesine karşılık gelen ve çok az sayıda kod çözme hatası veren gelişmiş Backflip kod çözme algoritması sayesinde mümkündür. Ayrıca, küçük kod çözme hata oranı GJS saldırısını engeller.

BIKE-1, BIKE-2 ve BIKE-3 algoritmaları IND-CPA güvenliyken, bunlara karşılık gelen BIKE-1-CCA, BIKE-2-CCA ve BIKE-3-CCA algoritmaları ise IND-CCA güvenli algoritmalardır. Bu çeşitleri ikinci turda eklenmiştir. Parametreler bir önceki seçimlerle aynıdır. Paylaşılan anahtarı üreten $\mathbf{K} : \{0,1\}^{2n} \rightarrow \{0,1\}^{\ell_K}$ özet fonksiyonuna ek olarak iki adet özet fonksiyon daha eklenmiştir. Bunlar sistem için rastgelelik oluşturmak üzere $\mathbf{G} : \{0,1\}^n \rightarrow \{0,1\}^r$ ve $\bar{\mathbf{G}} : \{0,1\}^n \rightarrow \{0,1\}^r$ özet fonksiyonlarıdır. $\bar{\mathbf{G}}$ fonksiyonunun değer kümesi $\{0,1\}^r$ kümesinin alt kümesidir ve bu fonksiton ağırlığı $\frac{t}{2}$ olan elemanları döndürür.

5.3.1 BIKE-1-CCA

Aynı BIKE-1 gibi polinomun tersini almaktan kaçınarak çok hızlı anahtar üretimi sağlar. Ayrıca daha önceki gibi, altında yatan kripto sistem mesajı hata vektöründe, kod kelimesini de rastgelete iletildiği şeklinde yorumlanmıştır. Bu şimdi daha uygundur, çünkü bu rastgeleliğin özet fonksiyonu G ile elde edilebileceği anlamına gelir.

5.3.1.1 Anahtar üretimi

Verilen güvenlik seviyesine göre parametreler kullanılarak açık anahtar $(\mathbf{f}_0, \mathbf{f}_1)$ ve gizli anahtar $(\mathbf{h}_0, \mathbf{h}_1, \sigma_1, \sigma_2)$ çıktısı verilir.

1. $\mathcal{R}$ üzerinde $\mathrm{wt}(\mathbf{h}_0) = \mathrm{wt}(\mathbf{h}_1) = \frac{w}{2}$ olacak şekilde ağırlıkları tek olan $\mathbf{h}_0$ ve $\mathbf{h}_1$ rastgele polinomları seçilir.
2. $\mathcal{R}$ üzerinde rastgele σ_1, σ_2 üretilir.
3. $\mathcal{R}$ üzerinde $\mathrm{wt}(\mathbf{g}) \approx \frac{r}{2}$ olacak şekilde ağırlığı tek olan $\mathbf{g}$ rastgele elemanı seçilir.
4. $(\mathbf{f}_0, \mathbf{f}_1) = (\mathbf{g}\mathbf{h}_1, \mathbf{g}\mathbf{h}_0)$ hesaplanır.

5.3.1.2 Kapsülleme

$(\mathbf{f}_0, \mathbf{f}_1)$ açık anahtarı kullanılarak kapsüllenmiş anahtar $\mathbf{K}$ ve şifrelenmiş $\mathbf{c}$ çıktısı verilir.

1. $(\mathbf{e}_0, \mathbf{e}_1) \in \mathcal{R}^2$ öyle ki $\mathrm{wt}(\mathbf{e}_0) + \mathrm{wt}(\mathbf{e}_1) = t$ hata vektörü seçilir.
2. $\mathcal{R}$ üzerinde $\mathbf{m} = \mathbf{G}(\mathbf{e}_0, \mathbf{e}_1)$ polinomu üretilir.
3. $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1) = (\mathbf{m}\mathbf{f}_0 + \mathbf{e}_0, \mathbf{m}\mathbf{f}_1 + \mathbf{e}_1)$ hesaplanır.
4. $K = \mathbf{K}(\mathbf{e}_0, \mathbf{e}_1, \mathbf{c}_0, \mathbf{c}_1)$ anahtarı hesaplanır.

5.3.1.3 Kapsülden çıkarma

Alınan $(\mathbf{h}_0, \mathbf{h}_1, \sigma_1, \sigma_2)$ gizli anahtarı ve şifreli metin $\mathbf{c}$ kullanılarak kapsülden çıkarılmış anahtara ulaşılır.

1. $\mathbf{s} = \mathbf{c}_0\mathbf{h}_0 + \mathbf{c}_1\mathbf{h}_1$ sendromu hesaplanır.
2. $(\mathbf{e}'_0, \mathbf{e}'_1)$ hata vektörünü kurtarmak için $\mathbf{s}$ (gürültüsüz) sendromu ile kod çözme algoritması denenir.
3. $\mathcal{R}$ üzerinde $\mathbf{m}' = \mathbf{G}(\mathbf{e}'_0, \mathbf{e}'_1)$ polinomu üretilir.
4. $\mathbf{c}' = (\mathbf{c}'_0, \mathbf{c}'_1) = (\mathbf{m}'\mathbf{f}_0 + \mathbf{e}'_0, \mathbf{m}'\mathbf{f}_1 + \mathbf{e}'_1)$ hesaplanır.
5. Eğer $\mathrm{wt}((\mathbf{e}'_0, \mathbf{e}'_1)) \neq t$ ya da $\mathbf{c} \neq \mathbf{c}'$ ise kod çözme hata verir, $K = \mathbf{K}(\sigma_1, \sigma_2, \mathbf{c}'_0, \mathbf{c}'_1)$ hesaplanır.
6. Kod çözme başarılı olursa $K = \mathbf{K}(\mathbf{e}'_0, \mathbf{e}'_1, \mathbf{c}'_0, \mathbf{c}'_1)$ anahtarı hesaplanır.

5.3.2 BIKE-2-CCA

BIKE-2 gibi bu çeşit de Niederreiter kripto sistemini baz alır. Bu durum kompakt bir formülasyona yol açar.

5.3.2.1 Anahtar üretimi

Verilen güvenlik seviyesine göre parametreler kullanılarak açık anahtar $\mathbf{h}$ ve gizli anahtar $(\mathbf{h}_0, \mathbf{h}_1, \sigma_1, \sigma_2)$ çıktısı verilir.

1. $\mathcal{R}$ üzerinde $\mathrm{wt}(\mathbf{h}_0) = \mathrm{wt}(\mathbf{h}_1) = \frac{w}{2}$ olacak şekilde ağırlıkları tek olan $\mathbf{h}_0$ ve $\mathbf{h}_1$ rastgele polinomları seçilir.
2. $\mathcal{R}$ üzerinde rastgele σ_1, σ_2 üretilir.
3. $\mathbf{h} = \mathbf{h}_1\mathbf{h}_0^{-1}$ hesaplanır.

5.3.2.2 Kapsülleme

h açık anahtarı kullanılarak kapsüllenmiş anahtar **K** ve şifrelenmiş **c** çıktısı verilir.

1. $(\mathbf{e}_0, \mathbf{e}_1) \in \mathcal{R}^2$ öyle ki $\text{wt}(\mathbf{e}_0) + \text{wt}(\mathbf{e}_1) = t$ hata vektörü seçilir.
2. $\mathbf{c} = \mathbf{e}_0 + \mathbf{e}_1 h$ hesaplanır.
3. $K = \mathbf{K}(\mathbf{e}_0, \mathbf{e}_1, \mathbf{c})$ anahtarı hesaplanır.

5.3.2.3 Kapsülden çıkarma

Alınan $(\mathbf{h}_0, \mathbf{h}_1, \sigma_1, \sigma_2)$ gizli anahtarı ve şifreli metin **c** kullanılarak kapsülden çıkarılmış anahtara ulaşılır.

1. $\mathbf{s} = \mathbf{ch}_0$ sendromu hesaplanır.
2. $(\mathbf{e}'_0, \mathbf{e}'_1)$ hata vektörünü kurtarmak için **s** (gürültüsüz) sendromu ile kod çözme algoritması denenir.
3. Eğer $\text{wt}((\mathbf{e}'_0, \mathbf{e}'_1)) \neq t$ ya da $\mathbf{c} \neq \mathbf{c}'$ ise kod çözme hata verir, $K = \mathbf{K}(\sigma_1, \sigma_2, \mathbf{c}')$ hesaplanır.
4. Kod çözme başarılı olursa $K = \mathbf{K}(\mathbf{e}'_0, \mathbf{e}'_1, \mathbf{c}')$ anahtarı hesaplanır.

5.3.3 BIKE-3-CCA

BIKE-1-CCA algoritmasında olduğu gibi hızlı ve tersini alma içermeyen anahtar üretimi yapar ve açık anahtar ve veriler iki blok içerir. Ayrıca IND-CPA eşinde olduğu gibi Backflip kod çözücünün gürültülü versiyonunu kullanır. Son olarak bir kez daha rastgeleliğe özet fonksiyonu $\bar{G}$ karar verir.

5.3.3.1 Anahtar üretimi

Verilen güvenlik seviyesine göre parametreler kullanılarak açık anahtar $(\mathbf{f}_0, \mathbf{f}_1)$ ve gizli anahtar $(\mathbf{h}_0, \mathbf{h}_1, \sigma_1, \sigma_2)$ çıktısı verilir.

1. $\mathcal{R}$ üzerinde $\text{wt}(\mathbf{h}_0) = \text{wt}(\mathbf{h}_1) = \frac{w}{2}$ olacak şekilde ağırlıkları tek olan $\mathbf{h}_0$ ve $\mathbf{h}_1$ rastgele polinomları seçilir.
2. $\mathcal{R}$ üzerinde rastgele σ_1, σ_2 üretilir.
3. $\mathcal{R}$ üzerinde $\text{wt}(\mathbf{g}) \approx \frac{r}{2}$ olacak şekilde ağırlığı tek olan $\mathbf{g}$ rastgele elemanı seçilir.
4. $(\mathbf{f}_0, \mathbf{f}_1) = (\mathbf{h}_1 + \mathbf{g}\mathbf{h}_0, \mathbf{g})$ hesaplanır.

5.3.3.2 Kapsülleme

$(\mathbf{f}_0, \mathbf{f}_1)$ açık anahtarı kullanılarak kapsüllenmiş anahtar $\mathbf{K}$ ve şifrelenmiş $\mathbf{c}$ çıktısı verilir.

1. $(\mathbf{e}_0, \mathbf{e}_1) \in \mathcal{R}^2$ öyle ki $\text{wt}(\mathbf{e}_0) + \text{wt}(\mathbf{e}_1) = t$ seçilir.
2. $\text{wt}(\mathbf{e}) = \frac{t}{2}$ olacak şekilde $\mathbf{e} = \bar{\mathbf{G}}(\mathbf{e}_0, \mathbf{e}_1)$ hesaplanır.
3. $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1) = (\mathbf{e} + \mathbf{e}_1\mathbf{f}_0, \mathbf{e}_0 + \mathbf{e}_1\mathbf{f}_1)$ hesaplanır.
4. $K = \mathbf{K}(\mathbf{e}_0, \mathbf{e}_1, \mathbf{c}_0, \mathbf{c}_1)$ anahtarı hesaplanır.

5.3.3.3 Kapsülden çıkarma

Alınan $(\mathbf{h}_0, \mathbf{h}_1, \sigma_1, \sigma_2)$ gizli anahtarı ve şifreli metin $\mathbf{c}$ kullanılarak kapsülden çıkarılmış anahtara ulaşılır.

1. $\mathbf{s} = \mathbf{c}_0 + \mathbf{c}_1\mathbf{h}_0$ sendromu hesaplanır.
2. $(\mathbf{e}'_0, \mathbf{e}'_1)$ hata vektörünü kurtarmak için $\mathbf{s}$ (en çok $\frac{t}{2}$ gürültü) sendromu ile kod çözme algoritması denenir.

3. $\mathbf{e}' = \bar{\mathbf{G}}(\mathbf{e}'_0, \mathbf{e}'_1)$ hesaplanır.

4. $\mathbf{c}' = (\mathbf{c}'_0, \mathbf{c}'_1) = (\mathbf{e}' + \mathbf{e}'_1\mathbf{f}_0, \mathbf{e}'_0 + \mathbf{e}'_1\mathbf{f}_1)$ hesaplanır.

5. Eğer $\mathrm{wt}((\mathbf{e}'_0, \mathbf{e}'_1)) \neq t$ ya da $\mathbf{c} \neq \mathbf{c}'$ ise kod çözme hata verir, $K = \mathbf{K}(\sigma_1, \sigma_2, \mathbf{c}'_0, \mathbf{c}'_1)$ hesaplanır.

6. Kod çözme başarılı olursa $K = \mathbf{K}(\mathbf{e}'_0, \mathbf{e}'_1, \mathbf{c}'_0, \mathbf{c}'_1)$ anahtarı hesaplanır.

BIKE-CCA çeşitlerini türetmek için kullanılan dönüşümlerle ilgili bazı genel özellikler ve IND-CPA eşlerinden farklı bazı yönleri vardır. Öncelikle artık gizli anahtar her zaman (σ_1, σ_2) ek rastgele dizisini içerir. Bu herhangi bir şekilde bir başarısızlık durumunda, "kesin reddetme (implicit rejection)" olarak bilinen teknikle kapsülden çıkarırken kullanılır. İkinci olarak, paylaşılan anahtar K şimdi sadece $e = (\mathbf{e}_0, \mathbf{e}_1)$ düz metininden değil, aynı zamanda $\mathbf{c}$ şifreli metninden de (BIKE-1-CCA ve BIKE-3-CCA algoritmalarında $(\mathbf{c}_0, \mathbf{c}_1)$) çıkarılır. Bu durum CCA dönüşümünde daha basit bir güvenlik azaltımı (security reduction) elde edilmesini sağlar. Son olarak, BIKE-1-CCA ve BIKE-3-CCA artık "rastgelelik"lerinin, düz metin bir özeti alınarak, deterministik olarak hesaplanmasını gerektiriyor ($\mathbf{G}$ ve $\bar{\mathbf{G}}$ özet fonksiyonları kullanılarak).

6. PERFORMANS ANALİZLERİ

NIST'in Kuantum Sonrası Kriptografi Standartlaştırma çağrısına gönderilen kod tabanlı kuantum sonrası algoritmalardan birinci turdaki 8 adet anahtar kapsülleme mekanizması için performans ölçümleri yapılmıştır. Bu ölçümlerde algoritmaların her biri için 10 anahtar üretimi yapılıp, çalışma sürelerinin (ms) aritmetik ortalaması alınarak hesaplanmıştır. Çalışmada kullanılan bilgisayarlar özellikleriyle beraber aşağıdaki gibidir;

- **SERVER**–POWER8 (architected), altivec supported CPU @ 4116.000000MHz, 107GB SCSI Disk, Linux 64-bit,
- **ASUS**–Intel® Core™ i7 -6500u CPU @ 3.16GHz, 512 GB SSD, Ubuntu 64-bit,
- **FUJITSU**–Intel® Core™ i5-3230M CPU @ 2.60GHz, Ubuntu 64-bit,
- **MAC**–Intel® Core™ i5-5257U CPU @ 2.70GHz, Ubuntu 64-bit.

Asimetrik (açık anahtarlı) kripto sistemleri kullanarak, simetrik kripto sistemler için ortak gizli anahtar oluşturabilen ve aynı zamanda anahtar değişim algoritması olarak da geliştirebilen anahtar kapsülleme mekanizmaları aşağıdaki gibi üç aşamada gerçekleşir:

1. *Anahtar Üretimi (KeyGen "KG") :* Kapsülleme ve kapsülden çıkarma işlemi için açık ve gizli anahtar üretir.
2. *Kapsülleme (Encaps "E"):* Açık anahtarı ve rastgele üretilen başka değerleri kullanarak bir şifreli metin ve K anahtarını oluşturur.
3. *Kapsülden Çıkarma (Decaps "D"):* Şifreli metni ve gizli anahtarı kullanarak K anahtarına ulaşır.

Dolayısıyla bu üç aşama için de çalışma süreleri ayrı ayrı değerlendirilmelidir. Her bir güvenlik seviyesi ve anahtar kapsülleme mekanizmasının üç aşaması için ayrı ayrı çalıştırılan algoritmaların süreleri 128-bit güvenlik seviyesi için Çizelge 6.1'de, 192-bit güvenlik seviyesi için Çizelge 6.2'de ve 256-bit güvenlik seviyesi için Çizelge 6.3'te verilmiştir. Çizelgelerdeki KG, E ve D kısaltmalarının açılımları yukarıdaki maddelerden de görülebileceği gibi anahatar kapsülleme mekanizmasının aşamalarını ifade etmektedir. Çizelgelerdeki algoritmaların detayları için [7], [2], [19], [3], [6], [4], [18] ve [25] çalışmaları incelenebilir.

Algoritmaların çalışma süreleri karşılaştırılırken her güvenlik seviyesi için şu adımlar izlenmiştir:

1. Öncelikle bir bilgisayar seçilmiştir.
2. Daha sonra bu bilgisayardaki anahtar üretim (KG) hızlarına bakılarak en hızlı 6 algoritma seçilerek bulunduğu kutu sarı renge boyanmıştır. Bu adım diğer üç bilgisayar için de tekrar edilmiştir.
3. Son olarak dört bilgisayarda da en hızlı ilk 6 algoritma arasına girenler belirlenerek anahtar üretimini en hızlı yapan algoritmayı ya da algoritmaları seçmek hedeflenmiştir.

Kapsülleme ve kapsülden çıkarma aşamaları içinde aynı adımlar izlenerek kapsülleme (E) için kutular pembe, kapsülden çıkarma (D) için kutular mavi renge boyanmıştır. Daha sonra her aşama için herhangi bir bilgisayarda ilk 6 arasına giren algoritmaların isimleri kalın harflerle yazılmıştır. Burada en hızlı 6 algoritma seçilmesinin sebebi ise ilk kesişimi yani her bilgisayarda da hızlı algoritmanın bu seçimle bulunmasıdır.

128-bit güvenlik seviyesi için Çizelge 6.1 incelenecek olursa BIKE-1 algoritması anahtar üretimi için dört bilgisayarda da en hızlılar arasına girmişken kapsülleme için üç, kapsülden çıkarma için iki bilgisayarda en hızlılar arasındadır. BIKE-2 algoritması sadece kapsüllemede en hızlılar arasına girerek diğer aşamalarda yavaş kalmıştır. BIKE-3 algoritması hem anahtar üretimi hem de kapsülleme aşamalarında dört bilgisayarda da en hızlılar arasında iken kapsülden çıkarma aşamasında sadece

bir bilgisayarda listeye girebilmiştir. BIKE algoritmasının her aşamada hızlı olan BIKE-1 ve BIKE-3 çeşitleriyle performans konusunda iyi olduğu görülebilir. HQC algoritmasına bakılacak olursa yalnızca HQC-I anahtar üretimi ve kapsülden çıkarma için dört bilgisayarda, kapsülleme için bir bilgisayarda en hızlılar arasına girmiştir, HQC-II ve HQC-III ise anahtar üretimi ve kapsülden çıkarma için hızlı iken kapsülleme aşamasında yavaş kalmışlardır. LAKE algoritması kapsülleme ve kapsülden çıkarma aşamalarında dört bilgisayarda, anahtar üretimi içinse iki bigisayarda en hızlılar arasında yer almıştır. Son olarak Ouroboros-R algoritması anahatar üretimi ve kapsülden çıkarma için üç, kapsülleme aşaması içinse dört bilgisayarda da en hızlılar arasındadır. BIG QUAKE, LEDAkem, LOCKER ve RLCE algoritmaları ise bu güvenlik seviyesinde diğerlerine göre çok daha yavaşlardır.

192-bit güvenlik seviyesi için Çizelge 6.2 incelendiğinde ise LAKE algoritması 128-bit güvenlik seviyesinde olduğu gibi kapsülleme ve kapsülden çıkarma aşamalarında dört bilgisayarda, anahtar üretimi içinse iki bigisayarda en hızlılar arasında yer almıştır. LOCKER algoritmasına bakılacak olursa LOCKER II anahtar üretimi için bir, kapsülleme ve kapsülden çıkarma aşamaları için dört bilgisayarda en hızlı 6 algoritmadan biri olmuştur. Ouroboros-R algoritması anahatar üretimi, kapsülleme ve kapsülden çıkarma için dört bilgisayarda da en hızlılar arasındadır. Aynı 128-bit güvenlik seviyesinde olduğu gibi BIG QUAKE, LEDAkem, RLCE ve bir önceki güvenlik seviyesinden farklı olarak BIKE ve HQC algoritmaları bu güvenlik seviyesinde diğerlerine göre çok daha yavaşlardır.

Son olarak 256-bit güvenlik seviyesi için Çizelge 6.3 incelendiğinde LAKE algoritması 128-bit ve 192-bit güvenlik seviyelerinde olduğu gibi kapsülleme ve kapsülden çıkarma aşamalarında dört bilgisayarda, anahtar üretimi içinse iki bigisayarda en hızlılar arasındadır. LOCKER algoritmasına bakılacak olursa bu sefer LOCKER VI anahtar üretimi için iki, kapsülleme ve kapsülden çıkarma aşamaları için dört bilgisayarda en hızlı 6 algoritmadan biri olmuştur. Ouroboros-R algoritması ise 192-bit güvenlik seviyesinde olduğu gibi anahatar üretimi, kapsülleme ve kapsülden çıkarma için dört bilgisayarda da en hızlılar arasında yer almıştır. Aynı 128-bit ve 192-bit güvenlik seviyesinde olduğu gibi BIG QUAKE, LEDAkem, RLCE

Çizelge 6.1: 128-bit güvenlik seviyesi için algoritma çalışma süreleri (ms)

Algoritma			Server	Asus	Fujitsu	Mac
BIG QUAKE		KG	556,059	346,654	423,731	353,484
		E	2,314	1,509	1,801	1,609
		D	2,879	1,745	2,259	1,745
BIKE	**BIKE-1**	KG	**1,873**	**0,113**	**0,355**	**0,148**
		E	1,936	**0,117**	**0,409**	**0,159**
		D	13,659	**1,507**	2,476	**1,693**
	BIKE-2	KG	4,357	2,399	2,942	2,569
		E	**0,382**	**0,081**	**0,158**	**0,122**
		D	3,358	1,531	2,279	1,642
	BIKE-3	KG	**0,238**	**0,086**	**0,232**	**0,147**
		E	**0,439**	**0,123**	**0,351**	**0,138**
		D	**2,612**	1,780	2,870	1,812
HQC	**HQC-I**	KG	**0,388**	**1,231**	**0,557**	**1,539**
		E	**0,674**	1,371	1,111	1,609
		D	**1,367**	**0,920**	**1,770**	**1,301**
	HQC-II	KG	**0,405**	**1,349**	**0,378**	**1,547**
		E	0,689	1,386	0,792	1,980
		D	**1,437**	**0,813**	**1,369**	**1,091**
	HQC-III	KG	**0,427**	1,575	**0,808**	1,731
		E	0,716	1,565	1,259	1,727
		D	**1,379**	**0,900**	**1,944**	**1,019**
LAKE		KG	2,320	**1,079**	2,324	**1,321**
		E	**0,359**	**0,197**	**0,441**	**0,253**
		D	**1,183**	**0,829**	**0,862**	**0,903**
LEDAkem		KG		50,165	59,831	53,078
		E		2,172	3,706	3,260
		D		2071,975	20,624	2932,986
LOCKER	LOCKER I	KG	4,720	1,732	4,688	1,921
		E	**0,651**	**0,349**	**0,590**	**0,432**
		D	2,480	1,658	2,411	1,983
	LOCKER IV	KG	6,620	11,477	7,468	12,015
		E	0,860	1,904	**0,770**	2,194
		D	2,884	7,612	**1,911**	9,521
	LOCKER VII	KG	14,88	19,096	15,457	23,509
		E	1,598	1,733	0,933	2,094
		D	4,952	5,454	3,155	5,998
Ouroboros-R		KG	**0,431**	**0,262**	**0,670**	**0,298**
		E	**0,654**	**0,334**	0,946	**0,320**
		D	**1,029**	**0,711**	**1,782**	**0,699**
RLCE	RLCE-A	KG	611,140	216,439	249,944	219,541
		E	3,548	1,182	3,376	1,341
		D	7,319	2,705	334,215	3,405
	RLCE-B	KG	1263,650	457,284	581,987	510,340
		E	5,516	1,872	5,337	2,301
		D	10,892	4,010	528,394	5,399

Çizelge 6.2: 192-bit güvenlik seviyesi için algoritma çalışma süreleri (ms)

Algoritma			Server	Asus	Fujitsu	Mac
BIG QUAKE		KG	4829,134	3081,125	3829,313	3423,600
		E	5,107	3,738	4,324	5,734
		D	11,211	12,187	22,382	14,133
BIKE	BIKE-1	KG	23,451	**0,214**	**0,892**	**0,294**
		E	**1,014**	**0,226**	1,064	**0,278**
		D	12,723	3,573	6,024	4,201
	BIKE-2	KG	15,225	3,095	11,281	3,154
		E	**0,951**	**0,128**	**0,453**	**0,152**
		D	9,127	3,288	6,251	4,009
	BIKE-3	KG	**0,603**	**0,242**	**0,552**	**0,318**
		E	1,161	**0,227**	**0,915**	**0,259**
		D	5,685	3,662	6,249	4,267
HQC	HQC-I	KG	**0,750**	2,616	**1,066**	3,198
		E	1,324	1,461	1,673	1,810
		D	**2,463**	**1,650**	**1,730**	2,750
	HQC-II	KG	**0,776**	2,867	**1,098**	**2,980**
		E	1,462	1,312	2,335	1,791
		D	**2,490**	**1,687**	3,418	**2,135**
	HQC-III	KG	**0,840**	**1,945**	**1,293**	**2,345**
		E	1,514	1,395	2,774	1,792
		D	**2,538**	1,857	3,948	**2,272**
LAKE		KG	**2,929**	**1,176**	4,604	**1,261**
		E	**0,404**	**0,189**	**0,549**	**0,221**
		D	**1,918**	**1,206**	**1,341**	**1,529**
LEDAkem		KG		209,324	251,610	400,031
		E		7,928	14,612	8,301
		D		5863,974	59,055	5953,342
LOCKER	**LOCKER II**	KG	5,566	**2,177**	3,324	3,201
		E	**0,686**	**0,375**	**0,505**	**0,428**
		D	**2,638**	**1,685**	**2,093**	**2,193**
	LOCKER V	KG	7,238	13,011	7,753	13,948
		E	**0,938**	2,482	**0,612**	3,491
		D	**3,900**	**0,189**	**2,573**	**0,311**
	LOCKER VIII	KG	16,309	22,162	13,621	29,806
		E	1,646	1,579	0,923	1,971
		D	5,087	4,530	**3,009**	4,870
Ouroboros-R		KG	**0,489**	**0,266**	**0,655**	**0,271**
		E	**0,773**	**0,344**	**0,827**	**0,339**
		D	**1,865**	**1,188**	**2,950**	**0,990**
RLCE	RLCE-A	KG	2454,471	861,411	1126,866	893,012
		E	8,413	2,804	8,131	3,500
		D	16,099	5,828	804,940	6,318
	RLCE-B	KG	4971,158	1982,791	2188,167	2004,059
		E	12,943	4,777	11,909	5,317
		D	24,228	9,618	1178,949	9,977

Çizelge 6.3: 256-bit güvenlik seviyesi için algoritma çalışma süreleri (ms)

Algoritma			**Server**	**Asus**	**Fujitsu**	**Mac**
BIG QUAKE		KG	8733,030	6027,456	6742,165	6587,916
		E	7,156	5,340	5,844	6,241
		D	16,618	16,618	24,873	19,315
BIKE	BIKE-1	KG	**1,334**	**0,315**	**1,633**	**0,389**
		E	2,595	**0,332**	1,566	**0,402**
		D	13,611	7,806	14,110	7,931
	BIKE-2	KG	23,450	6,970	25,391	7,090
		E	**1,015**	**0,178**	**0,600**	**0,199**
		D	12,724	7,889	12,867	9,230
	BIKE-3	KG	**1,334**	**0,329**	**1,256**	**0,329**
		E	2,592	**0,502**	1,801	**0,704**
		D	13,608	9,606	15,690	10,021
HQC	HQC-I	KG	**1,170**	3,024	**1,853**	**3,753**
		E	2,248	1,732	3,981	2,108
		D	**3,745**	**2,641**	**3,481**	**2,729**
	HQC-II	KG	**1,270**	**2,876**	**2,114**	4,201
		E	2,409	2,336	4,518	3,431
		D	**3,877**	**2,927**	**2,841**	**3,921**
	HQC-III	KG	**1,342**	3,626	**2,080**	4,114
		E	2,619	2,539	4,114	3,021
		D	4,190	**3,741**	3,974	4,250
	HQC-IV	KG	1,389	6,379	2,196	6,920
		E	2,690	9,420	4,662	9,812
		D	4,417	7,311	4,477	8,021
LAKE		KG	3,015	**1,200**	4,104	**2,298**
		E	**0,446**	**0,222**	**0,614**	**0,281**
		D	**2,512**	**1,751**	**2,013**	**2,790**
LEDAkem		KG		642,918	817,088	697,170
		E		21,954	44,102	28,910
		D		11140,809	116,214	11998,120
LOCKER	LOCKER III	KG	6,235	11,756	6,352	13,860
		E	**0,783**	1,788	**0,419**	1,901
		D	**3,653**	7,942	**2,545**	8,492
	LOCKER VI	KG	8,103	**2,551**	9,929	**2,832**
		E	**0,935**	**0,413**	**0,513**	**0,510**
		D	**3,980**	**2,133**	**2,559**	**2,392**
	LOCKER IX	KG	18,118	6,524	18,738	6,901
		E	**1,822**	0,858	**1,248**	1,078
		D	6,490	3,855	4,889	5,908
Ouroboros-R		KG	**0,597**	**0,355**	**1,163**	**0,402**
		E	**0,951**	**0,568**	**1,491**	**0,570**
		D	**2,597**	**2,062**	**1,885**	**2,001**
RLCE	RLCE-A	KG	5900,639	2379,420	2788,513	3003,130
		E	24,730	7,598	24,800	7,999
		D	48,427	17,528	2455,193	19,420
	RLCE-B	KG	11682,897	4723,123	5107,598	5099,072
		E	40,834	12,565	36,413	13,932
		D	76,103	28,202	3604,934	31,000

ve 128-bit güvenlik seviyesinden farklı olarak BIKE ve HQC algoritmaları bu güvenlik seviyesinde diğerlerine göre çok daha yavaş bulunmuşlardır.

Bu çalışma sonucunda öne çıkan algoritmalar Çizelge 6.4'te verilmiştir. Her güvenlik seviyesinde de hızlarıyla ön planda olan algoritmalar LAKE ve Ouroboros-R dir.

Çizelge 6.4: Çalışma sürelerine göre öne çıkan algoritmalar

Algoritma	**128-bit**	**192-bit**	**256-bit**
BIKE	✓		
HQC	✓		
LAKE	✓	✓	✓
LOCKER		✓	✓
Ouroboros-R	✓	✓	✓

Bu sonuçlara paralel olarak çağrı sürecinde LAKE, LOCKER ve Ouroboros-R algoritmaları birleştirilerek ROLLO [17] adı altında ikinci tura devam etmiştir. Aynı şekilde BIKE ve HQC algoritmaları da ikinci tura devam eden aday algoritmalar arasındadır. Son olarak LEDAkem algoritması da LEDApkc şifreleme-şifre çözme algoritması ile birleşerek LEDAcrypt ismiyle ikinci tura geçmiştir.

7. SONUÇ ve ÖNERİLER

Bu tezde, NIST'in düzenlediği Kuantum Sonrası Kriptografi Standartlaştırma çağrısı kapsamında aday gösterilen kod tabanlı kuantum sonrası algoritmalardan BIG QUAKE, BIKE, HQC, LAKE, LEDAkem, LOCKER, Ouroboros-R ve RLCE incelenerek 128-bit, 192-bit, 256-bit güvenlik seviyeleri için 4 farklı bilgisayarda çalıştırılmış ve performansları ölçülmüştür. Bu çalışma kapsamında öne çıkan algoritmalar Çizelge 6.4'te verilmiştir.

Bu çalışma sonucunda öne çıkan kuantum sonrası algoritmalardan HQC ve BIKE algoritmalarının kullandıkları sistemler ve kod aileleri, sırasıyla BCH ve QC-MDPC, incelenmiştir. Bunların yanında HQC algoritması ile neredeyse aynı sisteme sahip olan ve son zamanlarda öne çıkan Gabidulin kodlarını kullanan RQC kod tabanlı kuantum sonrası algoritmasının da detayları ve kod ailesi incelenmiştir. BCH kod ailesini kullanan HQC algoritmasının çalışma adımları örnek üzerinde gerçeklenmiştir.

Bu algoritmaların parametre karşılaştırmalarını yapmak ve analizlerini gerçekleştirmek; buradan elde edilecek sonuçlar ışığında algoritmaların performanslarını ve güvenliklerini artırmak amacıyla çalışmalarımız devam etmektedir.

KAYNAKLAR

[1] **Aguilar-Melchor, C., Blazy, O., Deneuville, J.-C., Gaborit, P., and Zémor, G.** (2018). Efficient encryption from random quasi-cyclic codes. *IEEE Transactions on Information Theory*, 64(5):3927–3943.

[2] **Aragon, N., Barreto, P., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.-C., Gaborit, P., Gueron, S., Guneysu, T., Melchor, C. A., et al.** (2017a). Bike: Bit flipping key encapsulation. *NIST Submission*.

[3] **Aragon, N., Blazy, O., Deneuville, J.-C., Gaborit, P., Hauteville, A., Ruatta, O., Tillich, J.-P., and Zémor, G.** (2017b). Lake-low rank parity check codes key exchange. *NIST Submission*.

[4] **Aragon, N., Blazy, O., Deneuville, J.-C., Gaborit, P., Hauteville, A., Ruatta, O., Tillich, J.-P., and Zémor, G.** (2017c). Locker-low rank parity check codes encryption. *NIST Submission*.

[5] **Augot, D., Loidreau, P., and Robert, G.** (2018). Generalized gabidulin codes over fields of any characteristic. *Designs, Codes and Cryptography*, 86(8):1807–1848.

[6] **Baldi, M., Barenghi, A., Chiaraluce, F., Pelosi, G., and Santini, P.** (2018). Ledakem: a post-quantum key encapsulation mechanism based on qc-ldpc codes. In *International Conference on Post-Quantum Cryptography*, pages 3–24. Springer.

[7] **Bardet, M., Barelli, E., Blazy, O., Torres, R. C., Couvreur, A., Gaborit, P., Otmani, A., Sendrier, N., and Tillich, J.-P.** (2017). Big quake binary goppa quasi–cyclic key encapsulation. *NIST Submission*.

[8] **Bringer, J., Chabanne, H., Cohen, G., Kindarji, B., and Zemor, G.** (2008). Theoretical and practical boundaries of binary secure sketches. *IEEE Transactions on Information Forensics and Security*, 3(4):673–683.

[9] **Deneuville, J.-C., Gaborit, P., and Zémor, G.** (2017). Ouroboros: A simple, secure and efficient key exchange protocol based on coding theory. In *International Workshop on Post-Quantum Cryptography*, pages 18–34. Springer.

[10] **Gabidulin, E. M.** (1985). Theory of codes with maximum rank distance. *Problemy Peredachi Informatsii*, 21(1):3–16.

[11] **Gadouleau, M. and Yan, Z.** (2006). Properties of codes with the rank metric. *arXiv preprint cs/0610099*.

[12] **Joiner, L. L. and Komo, J. J.** (1995). Decoding binary bch codes. In *Proceedings IEEE Southeastcon'95. Visualize the Future*, pages 67–73. IEEE.

[13] **Lidl, R. and Niederreiter, H.** (1996). *Finite Fields*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2 edition.

[14] **Lin, S. and Costello, D.** (2004). *Error Control Coding: Fundamentals and Applications*. Pearson-Prentice Hall.

[15] **Ling, S. and Xing, C.** (2004). *Coding Theory: A First Course*. Cambridge University Press.

[16] **Loidreau, P.** (2005). A welch–berlekamp like algorithm for decoding gabidulin codes. In *International Workshop on Coding and Cryptography*, pages 36–45. Springer.

[17] **Melchor, C. A., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.-C., Gaborit, P., Hauteville, A., Ruatta, O., Tillich, J.-P., et al.** (2018a). Rollo-rank-ouroboros, lake & locker. *NIST Submission*.

[18] **Melchor, C. A., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.-C., Gaborit, P., Hauteville, A., Zémor, G., and Bourges, I.-C.** (2017a). Ouroboros-r. *NIST Submission*.

[19] **Melchor, C. A., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.-C., Gaborit, P., Persichetti, E., Zémor, G., and Bourges, I.-C.** (2018b). Hamming quasi-cyclic (hqc). Technical report, Technical report, National Institute of Standards and Technology.

[20] **Melchor, C. A., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.-C., Gaborit, P., and Zémor, G.** (2017b). Rank quasi cyclic (rqc).

[21] **Misoczki, R., Tillich, J.-P., Sendrier, N., and Barreto, P. S.** (2013). Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. In *2013 IEEE international symposium on information theory*, pages 2069–2073. IEEE.

[22] **Reed, I. S. and Solomon, G.** (1960). Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, 8(2):300–304.

[23] **Shor, P. W.** (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332.

[24] **Vanstone, S. A. and Van Oorschot, P. C.** (2013). *An introduction to error correcting codes with applications*, volume 71. Springer Science & Business Media.

[25] **Wang, Y.** (2017). Rlcekey encapsulation mechanism (rlce-kem) specifcation. *NIST Submission*.

[26] **Url-1.** `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization`. [alındığı tarih: 31 Temmuz 2019].

[27] **Url-2.** https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions. [alındığı tarih: 31 Temmuz 2019].

[28] **Url-3.** https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions. [alındığı tarih: 31 Temmuz 2019].

EKLER

EK 1 : Türkçe-İngilizce Matematik Terimleri Sözlüğü

EK 2 : BIKE Algoritmasında Kullanılan Kod Çözme Algoritmaları

EK 1: Türkçe-İngilizce Matematik Terimleri Sözlüğü

Türkçe Terim	İngilizce Terim
Anahtar Kapsülleme Mekanizması	Key Encapsulation Mekanizm
Çevrimsel Matris	Circulant Matrix
Devirli Kod	Cyclic Code
Eşlik-Denetim Matrisi	Parity-Check Matrix
Kafes Tabanlı	Lattice-Based
Kapsülden Çıkarma	Decapsulation
Kapsülleme	Encapsulation
Kod	Code
Kod Çözme	Encoding
Kod Kelimesi	Codeword
Kod Tabanlı	Code-Based
Kodlama	Coding
Kriptografi	Cryptography
Kriptoloji	Cryptology
Lineer Kod	Linear Code
Mesaj	Message
Özet Fonksiyon	Hash Function
Şifre Çözme	Decryption
Şifreleme	Encryption
Şifreli Metin	Ciphertext
Tekrarlı Kod	Repetition Code
Üreteç Matris	Generator Matrix
Yarı-Devirli Kod	Quasi-Cyclic Code

EK 2: BIKE Algoritmasında Kullanılan Kod Çözme Algoritmaları

Algoritma 1 Bit Çevirme Algoritması

Girdi: $H \in \mathbb{F}_2^{(n-k)\times n}$ eşlik-denetim matrisi, $\mathbf{s} \in \mathbb{F}_2^{n\times k}$ gürültüsüz sendrom
Koşul: $\mathbf{e}H^T = \mathbf{s}$

$\mathbf{e} \leftarrow 0$
$\mathbf{s}' \leftarrow \mathbf{s}$
while $\mathbf{s}' \neq 0$ **do**
    $\tau \leftarrow$ eşik $\in [0,1]$
    for $j = 0$ to $n-1$ **do**
        if $\mathrm{wt}(h_j \star \mathbf{s}') \geq \tau\ \mathrm{wt}(h_j)$ **then**
            $e_j \leftarrow e_j + 1 \mod 2$
        $\mathbf{s}' \leftarrow \mathbf{s} - \mathbf{e}H^T$
return e

h_j, H matrisinin j inci sütununun satır vektörü şeklindeki ifadesidir. $\mathrm{wt}(h_j \star \mathbf{s})$, j yi içeren konrtol edilmemiş eşlik denklemlerinin sayısıdır.

Algoritma 2 Genişletilmiş Bit Çevirme Algoritması

Girdi: $H \in \mathbb{F}_2^{(n-k)\times n}$eşlik-denetim matrisi, $\mathbf{s} \in \mathbb{F}_2^{n\times k}$ gürültülü sendrom, $u \geq 0$ tam sayısı
Koşul: $\mathrm{wt}(\mathbf{s} - \mathbf{e}H^T) \leq u$

$\mathbf{e} \leftarrow 0$
$\mathbf{s}' \leftarrow \mathbf{s}$
while $\mathrm{wt}(\mathbf{s}') > u$ **do**
    $\tau \leftarrow$ eşik $\in [0,1]$
    for j **from** 0 **to** $n-1$ **do**
        if $\mathrm{wt}(h_j \star \mathbf{s}') \geq \tau\ \mathrm{wt}(h_j)$ **then**
            $e_j \leftarrow e_j + 1 \mod 2$
        $\mathbf{s}' \leftarrow \mathbf{s} - \mathbf{e}H^T$
return e

Eşik Seçim Kuralı. $\mathbf{s} = \mathbf{e}H^T$ sendrom ve $\mathbf{e}$ hata olsun. H matrisinin sütun ağırlığı $d = \frac{w}{2}$ şeklinde tanımlansın.

$$\pi_1 = \frac{\mathrm{wt}(s) + X}{td} \text{ ve } \pi_0 = \frac{w\ \mathrm{wt}(s) - X}{(n-t)d} \text{ öyle ki } X = \sum_{\ell \text{ tek}} (\ell - 1) \frac{r\binom{w}{l}\binom{n-w}{t-l}}{\binom{n}{t}}$$

olsun ($\pi_1 \geq \pi_0$).

$$T = \left\lceil \frac{\log \dfrac{n-t}{t} + d \log \dfrac{1-\pi_0}{1-\pi_1}}{log\dfrac{\pi_1}{\pi_0} + log\dfrac{1-\pi_0}{1-\pi_1}} \right\rceil \tag{7.1}$$

Bu değer sadece $n = 2r$, $w = 2d$, $t = \mathrm{wt}(\mathbf{e})$ hata ağırlığı ve $\mathrm{wt}(\mathbf{s})$ sendrom ağırlığına bağlıdır. Herhangi parametre seti içi bu değer önceden hesaplanabilir.

Algoritma 3 One-Round Bit Çevirme Algoritması

Girdi: $H \in \mathbb{F}_2^{(n-k)\times n}$ eşlik-denetim matrisi, $\mathbf{s} \in \mathbb{F}_2^{n\times k}$ sendrom, $u \geq 0$ tam sayısı
Koşul: $\mathrm{wt}(\mathbf{s} - \mathbf{e}H^T) \leq u$

$T \leftarrow \textsc{threshold}(\mathrm{wt}(\mathbf{s}))$
for $j = 0, \ldots, n-1$ **do**
    $\ell \leftarrow \min\{(\textsc{ctr}(H, \mathbf{s}, j), T\}$
    $J_\ell \leftarrow J_\ell \cup \{j\}$
$\mathbf{e} \leftarrow J_T$ ▷ $(**)$
$\mathbf{s}' \leftarrow \mathbf{s} - \mathbf{e}H^T$
while $\mathrm{wt}(\mathbf{s}') > S$ **do** ▷ $(***)$
    for $\ell = 0, \ldots, \delta$ **do** ▷ $(***)$
        $\mathbf{e}' \leftarrow \textsc{check}(H, \mathbf{s}', J_{T-\ell}, \frac{d}{2})$
        $(\mathbf{e}, \mathbf{s}') \leftarrow (\mathbf{e} + \mathbf{e}', \mathbf{s}' - \mathbf{e}'H^T)$
$\mathbf{e}' \leftarrow \textsc{check}(H, \mathbf{s}', \mathbf{e}, \frac{d}{2})$ ▷ $(**)$
$(\mathbf{e}, \mathbf{s}') \leftarrow (\mathbf{e} + \mathbf{e}', \mathbf{s}' - \mathbf{e}'H^T)$
while $\mathrm{wt}(\mathbf{s}') > u$ **do**
    $j \leftarrow \textsc{guess\_error\_pos}(H, \mathbf{s}', \frac{d}{2})$
    $(\mathbf{e}_j, \mathbf{s}') \leftarrow (\mathbf{e}_j + 1, \mathbf{s}' + h_j)$ ▷ $(*)$
return e

---

function $\textsc{check}(H, \mathbf{s}, J, T)$
    $\mathbf{e} \leftarrow 0$
    for $j \in J$ **do**
        if $\textsc{ctr}(H, \mathbf{s}, j) \geq T$ **then**
            $e_j \leftarrow 1$
    return e

function $\textsc{guess\_error\_pos}(H, \mathbf{s}, T)$
    loop
        $i \xleftarrow{\$} \mathbf{s}$ ▷ $(**)$
        for $j \in \mathbf{e}q_i$ **do** ▷ $(*), (**)$
            if $\textsc{ctr}(H, \mathbf{s}, j) \geq T$ **then**
                return j

function $\textsc{ctr}(H, \mathbf{s}, j)$
    return $\mathrm{wt}(h_j \cap \mathbf{s})$ ▷ $(*), (**)$

function $\textsc{threshold}(S)$
    return r, w, t ve S nin fonksiyonu

$(*)$ h_j, H matrisinin j inci sütununun satır vektörü şeklindeki ifadesidir. eq_i ise H matrisinin i nci satırıdır.
$(**)$ İkilik vektörler sıfırdan farklı pozisyonları ile tutulmaktadır.
$(***)$ Algoritma r, w, t ye bağlı olan S ve δ parametrelerini kullanır.

Eşik Seçim Kuralı THRESHOLD(S,t'). Daha önce olduğu gibi,

$$\pi_1 = \frac{S+X}{t'd} \text{ ve } \pi_0 = \frac{w\,S-X}{(n-t')d} \text{ öyle ki } X = \sum_{\ell \text{ tek}} (\ell-1)\frac{r\binom{w}{l}\binom{n-w}{t'-l}}{\binom{n}{t'}}$$

şeklinde tanımlansın. O halde THRESHOLD(S,t'),

$$t'\binom{d}{T}\pi_1^T(1-\pi_1)^{d-T} \geq (n-t')\binom{d}{T}\pi_0^T(1-\pi_0)^{d-T}$$

eşitsizliğini sağlayan en küçük T dir. t' iyimser olarak tahmin edildiğinden (algoritmada yalnızca hataların çevrildiği varsayılıyor: $t' = t - \mathrm{wt}(F)$), $\pi_1 > 1$ iken olabilir. Bu durumda THRESHOLD(S,t'),

$$1 \geq (n-t')\binom{d}{T}\pi_0^T(1-\pi_0)^{d-T} \tag{7.2}$$

eşitsizliğini sağlayan en küçük T dir.

Algoritma 4 Backflipping Algoritması

Girdi: $H \in \mathbb{F}_2^{(n-k)\times n}$ eşlik-denetim matrisi, $\mathbf{s} \in \mathbb{F}_2^{n\times k}$ sendrom, $u \geq 0$ tam sayısı
Koşul: $\mathrm{wt}(\mathbf{s} - \mathbf{e}H^T) \leq u$

$\mathbf{e} \leftarrow 0$; time $\leftarrow 1$; $F \leftarrow 0$ ▷ $F_j = j$ nin ölüm zamanı
while $\mathrm{wt}(\mathbf{s} - \mathbf{e}H^T) \leq u$ **do**
  for j öyle ki $F_j =$ time **do**
    $e_j \leftarrow 1 - e_j$; $F_j \leftarrow 0$
  $\mathbf{s}' \leftarrow \mathbf{s} - \mathbf{e}H^T$
  $T \leftarrow$ THRESHOLD$(\mathrm{wt}(\mathbf{s}'), t - \mathrm{wt}(F))$
  time $\leftarrow$ time $+ 1$
  for $j = 0, \ldots, n-1$ **do**
    if $\mathrm{wt}(\mathbf{s}' \cap h_j) \geq T$ **then** ▷ h_j, H matrisinin j. sütunu
      $e_j \leftarrow 1 - e_j$; $F_j \leftarrow 0$
      if $F_j \geq$ time **then**
      else time $+$ TTL$(\mathrm{wt}(\mathbf{s}' \cap h_j) - T)$
return e

function THRESHOLD(S,t')
  return r, w, t' ve S nin fonksiyonu

function TTL(δ)

ÖZGEÇMİŞ

Ad-Soyad : Burcu Ecem YILMAZ

Uyruğu : T.C.

Doğum Tarihi ve Yeri : 05.01.1994, Ankara

E-posta : burcuecemyilmaz@gmail.com

ÖĞRENİM DURUMU:

- **Yüksek Lisans** : 2019, TOBB Ekonomi ve Teknoloji Üniversitesi, Matematik Anabilim Dalı, Cebir, Kriptografi
- **Lisans** : 2016, TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Edebiyat Fakültesi, Matematik Bölümü

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	**Yer**	**Görev**
2016-2019	TOBB Ekonomi ve Teknoloji Üniversitesi	Tam Burslu Yüksek Lisans Öğrencisi

YABANCI DİL: İngilizce, Almanca

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- Bingöl G.B., Çolak B., Harma S.B., Kara S., **Yılmaz B.E.**, Yüksel M.B., 2018. Kuantum Sonrası Kod ve Kafes Tabanlı Bazı Algoritmaların Performans Analizleri, 13. Ankara Matematik Günleri-AMG 2018, 27-28 Nisan, TOBB Ekonomi ve Teknoloji Üniversitesi, Ankara, Türkiye.
- **Kara S.**, Yılmaz B.E., 2018. Kod Tabanlı Kuantum Sonrası Algoritmaların Parametrelerinin Karşılaştırılması, 13. Ankara Matematik Günleri-AMG 2018, 27-28 Nisan, TOBB Ekonomi ve Teknoloji Üniversitesi, Ankara, Türkiye.