TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KOD TABANLI KUANTUM SONRASI BAZI ŞİFRELEME ALGORİTMALARI VE ANAHTAR KAPSÜLLEME MEKANİZMALARININ İNCELENMESİ

YÜKSEK LİSANS TEZİ

Sevde KARA

Matematik Anabilim Dalı

Tez Danışmanı: Doç. Dr. Zülfükar SAYGI

NİSAN 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 162111006 numaralı Yüksek Lisans öğrencisi **Sevde KARA**'nın ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"KOD TABANLI KUANTUM SONRASI BAZI ŞİFRELEME ALGORİTMALARI VE ANAHTAR KAPSÜLLEME MEKANİZMALARININ İNCELENMESİ"** başlıklı tezi **11.04.2019** tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı:	**Doç. Dr. Zülfükar SAYGI** TOBB Ekonomi ve Teknoloji Üniversitesi	.....
Eş Danışman:	**Doç. Dr. Çetin ÜRTİŞ** TOBB Ekonomi ve Teknoloji Üniversitesi	.....
Jüri Üyeleri:	**Prof. Dr. Emrah KILIÇ (Başkan)** TOBB Ekonomi ve Teknoloji Üniversitesi	.....
	Doç. Dr. Murat CENK Orta Doğu Teknik 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.

Sevde KARA

ÖZET

Yüksek Lisans Tezi

KOD TABANLI KUANTUM SONRASI BAZI ŞİFRELEME ALGORİTMALARI VE ANAHTAR KAPSÜLLEME MEKANİZMALARININ İNCELENMESİ

Sevde KARA

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Matematik Anabilim Dalı

Tez Danışmanı: Doç. Dr. Zülfükar SAYGI

Tarih: NİSAN 2019

Bu tezde NIST çağrısında sunulan kod tabanlı bazı anahtar kapsülleme mekanizmaları ve şifreleme algoritmalarının incelemesi yapılmıştır. Bu algoritmalarda başta McEliece şifreleme sistemi temel alınmış ve bazı iyileştirmeler yapılarak kuantum sonrası dayanıklılıkları sağlanmaya çalışılmıştır. Bu bağlamda öncelikle McEliece şifreleme sisteminden bahsedilmiştir. İlk olarak detayları verilecek olan algoritmalarda kullanılan LRPC ve Goppa kod ailelerinin özelliklerine yer verilmiştir. Algoritmaların anahtar üretimi, şifreleme ve şifre çözme adımları ayrıntılarıyla verilmiş daha sonra tüm süreç örneklerle birlikte gösterilmiştir. Son olarak bu üç algoritmanın parametre uzunlukları karşılaştırılmıştır.

Anahtar Kelimeler: Kuantum sonrası kriptografi, Anahtar kapsülleme mekanizması, Şifreleme algoritması, Kod tabanlı kriptografi.

ABSTRACT

Master of Science

ANALYSING OF SOME CODE BASED POST QUANTUM ENCRYPTION ALGORITHMS AND KEY ENCAPSULATION MECHANISMS

Sevde KARA

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

Supervisor: Doç. Dr. Zülfükar SAYGI

Date: April 2019

In this thesis, some code based key encapsulation mechanisms and encryption algorithms are analysed. In these algorithms, especially McEliece encryption system is based and their post quantum security is tried to be ensure with some improvements. In this sense, McEliece encryption system is mentioned at first. Then, properties of LRPC and Goppa Code families that are used in mentioned algorithms are included. Key generation, encryption and decryption steps of these algoritms are explained in details. Subsequently, whole process of these systems are demonstrated with examples. Finally, parameter sets of these three algorithms are compared.

Keywords: Post quantum cryptography, Key encapsulation mechanism, Encryption algorithm, Code based cryptography.

TEŞEKKÜR

Yüksek lisans eğitimim boyunca yardımları ve tecrübeleriyle beni yönlendiren, desteğini esirgemeyen değerli hocam Doç. Dr. Zülfükar SAYGI'ya ve eş danışman hocam Doç. Dr. Çetin ÜRTİŞ'e; kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Matematik Bölümü öğretim üyelerine teşekkürlerimi sunarım. Eğitim hayatım boyunca maddi ve manevi destekleriyle her zaman yanımda olan aileme ve destek olarak motivasyonumu yükselten tüm arkadaşlarıma teşekkürü bir borç bilirim. Son olarak yüksek lisans eğitimimde sağladığı burstan dolayı TOBB Ekonomi ve Teknoloji Üniversitesi'ne teşekkürlerimi sunarım.

İÇİNDEKİLER

ÇİZELGE LİSTESİ

SEMBOL LİSTESİ

Bu tezde kullanılan simgeler açıklamaları ile birlikte aşağıda yer almaktadır.

Simgeler	**Açıklama**
$\mathbb{F}_{q^m}$	q^m elemanlı sonlu cisim
$\mathbb{F}_q^n$	$\mathbb{F}_q$ üzerinde n boyutlu vektör uzayı
$\mathbb{F}_q^{k\times n}$	$\mathbb{F}_q$ üzerinde $k \times n$ boyutlu matrisler kümesi
$\mathbf{m}$	Açık metin
$\mathbf{c}$	Şifreli metin
$\mathbf{e}$	Hata vektörü
n	Kod uzunluğu
k	Kod boyutu
d	Kod uzaklığı
(n,k,d)	Uzunluğu n, boyutu k, minimum uzaklığı d olan lineer kod

1. GİRİŞ

Açık anahtarlı kriptosistemler şifreleme, imzalama ve anahtar paylaşımı amacıyla kullanılmaktadır. 1970'li yıllardan itibaren kullanılan açık anahtarlı algoritmalardan bazıları ise çarpanlara ayırma problemine dayalı RSA, ayrık logaritma problemine dayalı Diffie-Hellman anahtar değişimi ve DSA (Dijital İmza Algoritması), eliptik eğri ayrık logaritma problemine dayalı ECC (Eliptik Eğri Kriptosistemi) ve kod çözme problemine dayalı McEliece kriptosistemidir [12].

Günümüzde RSA, Diffie-Hellman anahtar değişimi, DSA ve ECC yaygın olarak kullanılmaktadır. Bu algoritmaların kullanım sebebi güvenli olmalarının yanı sıra performansları ve anahtar uzunluklarıdır. McEliece sistemi ise güvenli olmasına karşın anahtar uzunluğunun diğerlerine göre daha büyük olması sebebiyle geçmiş yıllarda yeterince ilgi görmemiştir. Fakat son yıllarda klasik bilgisayarlar için çözülmesi zor olan matematik problemlerini çözebilmek adına kuantum bilgisayarlarla ilgili pek çok çalışma yapılmaktadır. Büyük çapta kuantum bilgisayarlar üretilebilirse, şuan kullanılmakta olan açık anahtarlı kriptosistemlerin dayandığı çarpanlara ayırma ve ayrık logaritma problemleri Shor algoritması [18] ile polinom zamanda kırılabilecektir. Bu durumda birçok güvenlik uygulamasında sorunlar yaşanacaktır. Klasik ve kuantum bilgisayarlarla yapılan ataklara dayanıklı kriptografik sistemlerin geliştirilmesi kuantum sonrası (post-kuantum) kriptografinin temel amacıdır. Bu kapsamda kod tabanlı, kafes tabanlı, çok değişkenli polinom tabanlı, özet fonksiyonları tabanlı ve eliptik eğri tabanlı sistemler geliştirilmeye başlanmıştır. Klasik McEliece algoritması günümüzde hala geliştirilerek, kuantum dayanıklılığı bozulmadan parametre güncelleştirmeleri/iyileştirmeleri yapılmaktadır. Hali hazırda kullanılmakta olan açık anahtarlı algoritmaları kırmak için yeterli büyüklükte kuantum bilgisayarların 20 yıl gibi bir süre içerisinde üretileceği tahmin edilmektedir. Modern açık anahtarlı kriptografi altyapılarının yayılması neredeyse 20 yıl aldığından, kuantum bilgisayarların üretim zamanı tam olarak tahmin edilemese de

kuantum bilgisayarlara dayanıklı bilgi güvenlik sistemlerinin şimdiden hazırlanmaya başlanması gerekmektedir. Bu anlamda NIST (National Institute of Standards and Technology) kuantum dayanıklı bir ya da daha fazla açık anahtarlı kriptografik algoritmayı değerlendirmek ve standartlaştırmak için bir süreç başlatmıştır [20].

NIST tarafından son tarih olarak belirlenen Kasım 2017 tarihine kadar toplam 82 başvuru yapılmıştır. Bu başvurular arasından 69 tanesi uygun başvuru olarak değerlendirilmiştir. 2018'de tamamlanan ilk aşamadaki 69 sistemden 20 tanesi kod tabanlıdır. Çizelge 1.1 de ilk aşamada yer alan 69 kriptosistemin sınıflarına göre isim olarak dağılımı verilmiştir. İkinci aşamaya geçmeye hak kazanan 26 sistemden ise 7 tanesi kod tabanlıdır [1]. Ayrıca ikinci aşamada yer alan LEDAcrypt sistemi LEDAkem ve LEDApkc sistemlerinin birleşimi iken yine ikinci aşamada görülen ROLLO sistemi de LAKE, LOCKER ve Ouroboros-R sistemlerinin birleşimidir. Bu durumda ikinci aşamada 7 olarak görünen kod tabanlı algoritmaların sayısı aslında 10'u bulmaktadır. Çizelge 1.2 de ikinci aşamaya geçmeye hak kazanan 26 kriptosistemin sınıflarına göre isim olarak dağılımı verilmiştir.

Bu çalışmada McEliece'in yanı sıra NIST'in Kuantum Sonrası Kriptografi Standartlaştırma çağrısına sunulan McNie, Klasik McEliece ve NTS-KEM kod tabanlı algoritmalarının detayları sunulacak ve parametre karşılaştırmaları yapılacaktır.

1.1 Bazı Temel Tanımlar

Bu kısımda tez boyunca kullanılacak temel bazı bilgiler kısaca özetlenecektir. Detaylar için [10] ve [19] nolu kaynaklar incelenebilir.

Tanım 1.1. *Bir (n,k) lineer kod için satırları baz olan $k \times n$ boyutundaki G matrisine kodun üreteç matrisi denir.*

Sonuç 1.1. *$G \in \mathbb{F}_q^{k\times n}$, bir (n,k) lineer kod $C \subseteq \mathbb{F}_q^n$ için bir üreteç matristir ancak ve ancak*

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

Çizelge 1.1: Round 1 Aday Algoritmalar

	İmzalama	Anahtar Kapsülleme	Şifreleme
Kod Tabanlı	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
Kafes Tabanlı	CRYSTALS-DILITHIUM DRS FALCON pqNTRUSign qTESLA	CRYSTALS-KYBER Ding Key Exchange FrodoKEM HILA5 KINDI LAC LIMA Lizard LOTUS NewHope NTRUEncrypt NTRU-HRSS-KEM NTRU Prime OKCN/AKCN/CNKE Round2 SABER Three Bears Titanium	Compact LWE EMBLEM and R.EMBLEM Giophantus KINDI LAC LIMA Lizard LOTUS NTRUEncrypt Odd Manhattan OKCN/AKCN/CNKE Titanium
Çok Değişkenli	DualModeMS GeMSS Gui HIMQ-3 MQDSS LUOV Rainbow SRTPI*	CFPKM DME	SRTPI*
Diğer	Gravity SPHINCS SPHINCS+ WalnutDSA Picnic Post Quantum RSA	RVB* HK17* SIKE Lepton Post Quantum RSA Mersenne-756839	Guess Again Post Quantum RSA

* Süreçten çekilen algoritmalar [22]

Çizelge 1.2: Round 2 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	

Tanım 1.2. *Bir (n,k) lineer C kodunun duali her bir elemanı C kodunun tüm elemanlarına dik olan elemanları olarak tanımlanır ve $C^{\perp}$ ile gösterilir. Bu durumda*

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

olur.

Sonuç 1.2. *$\mathbb{F}_q$ cismi üzerindeki bir (n,k) lineer C kodunun duali, $\mathbb{F}_q$ cismi üzerinde bir $(n,n-k)$ lineer koddur.*

Tanım 1.3. *Bir (n,k) lineer C kodunun dualinin üreteç matrisine C kodunun eşlik denetim matrisi denir.*

Sonuç 1.3. *$H \in \mathbb{F}_q^{(n-k)\times n}$, bir (n,k) lineer kod $C \subseteq \mathbb{F}_q^n$ için bir eşlik denetim matristir ancak ve ancak*

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

Tanım 1.4. *$\mathbb{F}_q$ üzerinde n uzunluğunda bir C lineer kodu $\mathbb{F}_q^n$ cisminin bir altuzayıdır.*

Tanım 1.5. $\mathbf{x}$ *ve* $\mathbf{y}$ *bir C kodu üzerinde n uzunluğunda iki kelime olsun.* $\mathbf{x}$ *ve* $\mathbf{y}$ *kelimeleri arasındaki Hamming uzaklığı* $d(\mathbf{x},\mathbf{y})$ *ile gösterilir ve* $\mathbf{x}$ *ile* $\mathbf{y}$ *kelimelerinin farklı olduğu yerlerin sayısı olarak tanımlanır.*
$\mathbf{x} = (x_1,\ldots,x_n)$, $\mathbf{y} = (y_1,\ldots,y_n)$ *ve*

$$d(x_i,y_i) = \begin{cases} 1, & x_i \neq y_i \\ 0, & x_i = y_i \end{cases}$$

olmak üzere,

$$d(\mathbf{x},\mathbf{y}) = d(x_1,y_1) + \cdots + d(x_n,y_n)$$

olur.

Tanım 1.6. *En az iki kelime içeren bir C kodunun en kısa uzaklığı (minimum distance)* $d(C)$ *ile gösterilir ve*

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

olarak ifade edilir.

Tanım 1.7. $\mathbf{x}$, $\mathbb{F}_q^n$ *cisminin bir elemanı olsun.* $\mathbf{x}$ *vektörünün Hamming ağırlığı (weight)* $wt(\mathbf{x})$ *ile gösterilir ve*

$$wt(\mathbf{x}) = d(\mathbf{x},\mathbf{0})$$

olup, $\mathbf{x}$ *vektöründeki sıfırdan farklı elemanların sayısı ile ifade edilir. Burada* $\mathbf{0}$*, sıfır kelimesidir.*

Tanım 1.8. *C bir kod olsun. C kodunun en küçük Hamming ağırlığı* $wt(C)$ *ile gösterilir ve C kodundaki sıfırdan farklı kelimelerin ağırlıklarının en küçüğü olarak ifade edilir.*

Teorem 1.1. *C,* $\mathbb{F}_q$ *üzerinde bir lineer kod olsun. Bu durumda* $d(C) = wt(C)$ *olur.*

2. KOD AİLELERİ

Bu bölümde algoritmalarda kullanılan LRPC (Low Rank Parity Check) ve Goppa kodlarla ilgili bazı temel bilgiler verilecektir. Ayrıntılı bilgi için [2] ve [8] nolu kaynaklar incelenebilir.

2.1 LRPC Kodlar

Tanım 2.1. *$\mathbb{F}_{q^m}$ cismi üzerinde d ranklı, n uzunluğunda ve k boyutunda bir LRPC kod $(n-k)\times n$ boyutunda $H=(h_{ij})$ eşlik denetim matrisine sahiptir. $\mathbb{F}_{q^m}$ cisminin H matrisinin elemanlarıyla üretilen alt vektör uzayının boyutu en fazla d dir ve d, H matrisinin ağırlığı olarak adlandırılır. H matrisinin h_{ij} elemanlarıyla üretilen F alt vektör uzayı, bazlarından biri olan $\{F_1,F_2,\ldots,F_d\}$ ile gösterilir.*

Örnek. $\mathbb{F}_{2^4}$ cismi üzerinde $d=2$ ranklı, $n=6$ uzunluğunda ve $k=4$ boyutunda bir LRPC kodun eşlik denetim matrisi

$$H=\begin{bmatrix} 1 & \alpha^2 & 0 & \alpha^2 & 1 & 0 \\ \alpha^2 & 1 & \alpha^2 & 0 & 0 & 1 \end{bmatrix}$$

olsun. H matrisinin h_{ij} elemanlarıyla üretilen F alt vektör uzayı, $\{1,\alpha^2\}$ bazı ile gösterilebilir.

Örnek. $\mathbb{F}_{2^4}$ cismi üzerinde $\mathbf{e}=(0,0,\alpha,0,\alpha,0)$ hata vektörü olsun. $\mathbf{e}$ vektörünün bileşenlerinden oluşan 1 boyutlu E alt vektör uzayı ise $\{\alpha\}$ bazı ile gösterilebilir.

Tanım 2.2. *d ranklı bir QC-LRPC kod ise düşük d ağırlığında yarı devirli bir H eşlik denetim matrisine sahip yarı devirli bir koddur.*

q bir asal sayının pozitif tam sayı kuvveti, m pozitif bir tam sayı ve V_n ise $\mathbb{F}_{q^m}$ sonlu cismi üzerinde n boyutlu bir vektör uzayı olsun. $B=(\alpha_1,\ldots,\alpha_m)$ ise $\mathbb{F}_{q^m}$ cisminin $\mathbb{F}_q$ üzerinde bir bazı olsun. $\mathcal{F}_i$ ise $\mathbb{F}_{q^m}$ cisminden $\mathbb{F}_q$ cismine $\mathcal{F}_i(x)$, x'in β bazında i.

koordinatı olacak şekilde bir dönüşüm olsun. Herhangi bir $\mathbf{v} = (v_1, \ldots, v_n) \in V_n$ vektörü $V \in \mathbb{F}_q^{m \times n}$ matrisine dönüştürülür. Matrisin elemanları $V_{ij} = \mathcal{F}_i(v_j)$ şeklindedir. Bir $\mathbf{v}$ vektörünün rank ağırlığı ilgili V matrisinin rankı olarak tanımlanabilir. Bu değer rank($\mathbf{v}$) olarak adlandırılırsa, $\mathbf{x}$ ve $\mathbf{y}$ vektörleri arasındaki uzaklık $\mathrm{rd}(\mathbf{x}, \mathbf{y}) = \mathrm{rank}(\mathbf{x} - \mathbf{y})$ şeklinde ifade edilir.

$\mathbb{F}_{2^m} = \mathbb{F}_{2^4}$ sonlu cismini, $m = 4$ dereceli $f(x) = x^4 + x + 1$ ilkel polinomunu kullanarak $\mathbb{F}_2$ cisminin bir genişlemesi olarak inşa edelim. α, $f(x)$ polinomunun ilkel bir kökü olsun.

Örnek. $\mathbb{F}_{2^m}$ cisim genişlemesinin elemanlarını $\mathbb{F}_2$ üzerinde göstermek için

$$a_0 + a_1\alpha + \cdots + a_{m-1}\alpha^{m-1} \leftrightarrow a_0 a_1 \ldots a_{m-1}$$

dönüşümü kullanılabilir. $m = 4$ ve $f(x) = x^4 + x + 1$ indirgenemez polinomu için bu dönüşüm aşağıdaki gibidir.

1	[1000]	α	[0100]	α^2	[0010]	α^3	[0001]
α^4	[1100]	α^5	[0110]	α^6	[0011]	α^7	[1101]
α^8	[1010]	α^9	[0101]	α^{10}	[1110]	α^{11}	[0111]
α^{12}	[1111]	α^{13}	[1011]	α^{14}	[1001]	0	[0000]

Örnek. $V_n = V_{10}$, 10 boyutlu bir vektör uzayı ve $\mathbb{F}_{q^m} = \mathbb{F}_{2^4}$ olsun. $B = (\alpha_1, \ldots, \alpha_4)$ ise $\mathbb{F}_{2^4}$ cisminin $\mathbb{F}_2$ üzerinde bir bazı olsun.

$\mathcal{F}_i$: $\mathbb{F}_{2^4} \longrightarrow \mathbb{F}_2$,

$\mathbf{v} = (\alpha^8, \alpha^{12}, 0, \alpha^8, \alpha^8, 0, \alpha^8, 0, \alpha^8, \alpha^8) \in V_{10}$ olmak üzere;

$V_{ij} = \mathcal{F}_i(v_j)$ ile $V = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{F}_2^{4 \times 10}$ elde edilir.

$\mathbf{v}$ vektörünün Hamming ağırlığı sıfırdan farklı elemanlarının sayısına eşit olduğundan $\mathrm{wt}(\mathbf{v}) = 7$ olur. Bu matrisin rankı 2 olduğundan ilgili $\mathbf{v}$ vektörünün rank ağırlığı da 2 dir ve rank($\mathbf{v}$) = 2 şeklinde gösterilir.

Tanım 2.3. *(RSD) (Rank Syndrome Decoding) H, $(n-k) \times n$ boyutunda ve $\mathbb{F}_{q^m}$ üzerinde $k \leq n$ olacak şekilde bir matris olsun. $\mathbf{s} \in \mathbb{F}_{q^m}^k$ ve r bir tam sayı olmak üzere*

rank$(\mathbf{x}) = r$ *ve* $H\mathbf{x}^T = \mathbf{s}$ *koşullarını sağlayan* $\mathbf{x}$ *vektörünü bulma problemi RSD problemi olarak adlandırılır.*

2.2 Goppa Kodlar

$\Gamma(L, g(z))$ Goppa kodu $\mathbb{F}_{q^m}$ cisim genişlemesi üzerinde t dereceli $g(z)$ Goppa polinomu ve $\mathbb{F}_{q^m}$ cisminin L alt kümesiyle tanımlanır.

$$\begin{aligned} g(z) &= g_0 + g_1 z + \cdots + g_t z^t = \sum_{i=0}^{t} g_i z^i, \\ L &= \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}_{q^m} \end{aligned}$$

$g(\alpha_i) \neq 0 \; \forall \; \alpha_i \in L$. $\mathbb{F}_q$ üzerinde bir $\mathbf{c} = (c_1, \ldots, c_n)$ vektörü ile

$$R_{\mathbf{c}}(z) = \sum_{i=1}^{n} \frac{c_i}{z - \alpha_i}$$

fonksiyonu tanımlanır. Burada $\dfrac{1}{z-\alpha_i} = (z-\alpha_i)^{-1}$ polinomu,

$$(z - \alpha_i) \cdot \frac{1}{z - \alpha_i} \equiv 1 \pmod{g(z)}$$

denkliğini sağlayan, yani $z - \alpha_i$ elemanının tersi olan polinomdur.

Tanım 2.4. $\Gamma(L, g(z))$ *Goppa kodu,*

$$R_{\mathbf{c}}(z) \equiv 0 \pmod{g(z)}$$

koşulunnu sağlayan tüm $\mathbf{c}$ *vektörlerinden oluşur.*

Goppa Kod Parametreleri:

Bir Goppa kodun parametreleri n uzunluğu, k boyutu ve minimum d uzaklığıdır. n, k ve d parametrelerine sahip bir Goppa kodu için (n, k, d) gösterimi kullanılır. n parametresi, $\mathbf{c}$ kod kelimelerinin uzunluğunu temsil eder ve L ile sabitlenmiştir. Diğer iki parametre için ise alt sınırlar değiştirilebilir.

Teorem 2.1. *n uzunluğunda bir* $\Gamma(L, g(z))$ *Goppa kodu* $\mathbb{F}_q$ *üzerinde,*

- $k \geq n - mt$
- $d \geq t + 1$

özelliklerini taşıyan bir lineer koddur.

Goppa Kodun Eşlik Denetim Matrisi:

Kod çözmek için koda ait bir eşlik denetim matrisine ihtiyaç duyulur. $\mathbf{c}$ bir kod kelimesidir ancak ve ancak

$$\sum_{i=1}^{n} c_i p_{ij} = 0,\ 1 \leq j \leq t,$$

$\frac{1}{z-\alpha_i} \equiv p_{i1} + p_{i2}z + \cdots + p_{it}z^{t-1} \pmod{g(z)}$. H eşlik denetim matrisi $\mathbf{c}H^T = 0$ eşitliğini sağlar. Dolayısıyla,

$$H = \begin{bmatrix} p_{11} & \cdots & p_{n1} \\ \vdots & \ddots & \vdots \\ p_{1t} & \cdots & p_{nt} \end{bmatrix} \tag{2.1}$$

p_{ij} elemanlarını belirlemek için,

$$p_i(z) \equiv (z-\alpha_i)^{-1} \equiv -\frac{g(z) - g(\alpha_i)}{z-\alpha_i} \cdot g(\alpha_i)^{-1}.$$

hesaplaması yapılır ve $(z-\alpha_i)$ ile çarpılarak bu denklik kontrol edilebilir:

$$-(z-\alpha_i) \cdot g(\alpha_i)z - \alpha_i \cdot g(\alpha_i)^{-1} = -g(z)g(\alpha_i)^{-1} + 1 \equiv 1 \pmod{g(z)}.$$

Şimdi de $h_i := g(\alpha_i)^{-1}$ tanımlanır ve $g(z) = g_0 + g_1 z + \cdots + g_t z^t$ polinomu bir önceki denklemde yerine yazılır:

$$p_i(z) = -\frac{g_t \cdot (z^t - \alpha_i^t) + \cdots + g_1 \cdot (z-\alpha_i)}{z-\alpha_i} \cdot h_i.$$

Buradaki kesir yeniden düzenlenerek aşağıdaki şekilde yazılabilir:
$g_t(z^{t-1} + z^{t-2}\alpha_i + \cdots + \alpha_i^{t-1}) + g_{t-1}(z^{t-2} + z^{t-3}\alpha_i + \cdots + \alpha_i^{t-2}) + \cdots + g_2(z+\alpha_i) +$
g_1.

$p_i(z) = p_{i1} + p_{i2}z + \cdots + p_{it}z^{t-1}$ yerine koyulursa, p_{ij} ler için aşağıdaki ifadeler bulunur:

$$\begin{aligned} p_{i1} &= -(g_t\alpha_i^{t-1} + g_{t-1}\alpha_i^{t-2} + \cdots + g_2\alpha_i + g_1)h_i \\ p_{i2} &= -(g_t\alpha_i^{t-2} + g_{t-1}\alpha_i^{t-3} + \cdots + g_2)h_i \\ &\vdots \\ p_{i(t-1)} &= -(g_t\alpha_i + g_{t-1})h_i \\ p_{it} &= -g_t h_i. \end{aligned} \tag{2.2}$$

$$C = \begin{bmatrix} -g_t & -g_{t-1} & -g_{t-2} & \cdots & -g_1 \\ 0 & -g_t & -g_{t-1} & \cdots & -g_2 \\ 0 & 0 & -g_t & \cdots & -g_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -g_t \end{bmatrix},$$

$$X = \begin{bmatrix} \alpha_1^{t-1} & \alpha_2^{t-1} & \cdots & \alpha_n^{t-1} \\ \alpha_1^{t-2} & \alpha_2^{t-2} & \cdots & \alpha_n^{t-2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ 1 & 1 & \cdots & 1 \end{bmatrix} \text{ ve } Y = \begin{bmatrix} h_1 & 0 & 0 & \cdots & 0 \\ 0 & h_2 & 0 & \cdots & 0 \\ 0 & 0 & h_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & h_n \end{bmatrix}$$

olmak üzere (2.1) ve (2.2) birleştirilerek, $H = CXY$ bulunur.

Goppa Kod Üreteç Matrisi:

H eşlik denetim matrisi hataları düzeltmek için kullanılırken, mesajları kodlamak ve kod çözmek için de üreteç matrisine ihtiyaç duyulur. Bir kod kelimesi $\mathbf{m} = (m_1, \ldots, m_k)$ mesajı G matrisiyle çarpılarak elde edilir. Kodun tüm kelimeleri için geçerli olan $\mathbf{c}H^T = \mathbf{0}$ eşitliği kullanılarak da hatalar düzeltilip kod kelimesine ulaşılır. Bu sebeple,

$$GH^T = \mathbf{0}$$

kullanılarak H matrisinden G matrisi elde edilebilir.

3. MCELIECE

Kod tabanlı kriptografi, kuantum bilgisayara sahip düşmana karşı güvenli açık anahtarlı kriptosistemlerin inşasında kullanılabilecek birkaç matematiksel teknikten biridir. Robert McEliece ilk kod tabanlı kriptosistemi 1978 yılında önermiştir. McEliece algoritması bugüne kadar yapılan tüm kriptanaliz girişimlerine karşı güvenilirliğini korumuştur. McEliece lineer kodlardan bir kelimeyi şifreli metin olarak kullanır. Kodun rastgele bir bazı (üreteç matrisi) şifreleme yapmak için herkesin ulaşabileceği açık anahtar olarak kullanılır. Gizli matematiksel bağlantıyı (kod için hızlı bir kod çözme algoritması) bilen yasal kullanıcılar şifreli metinden hataları uzaklaştırabilir ve açık metne ulaşabilirler. Düşmanlar ise kuantum bilgisayarlar için bile zor olduğu kabul edilen genel kod çözme problemini kullanmalıdırlar.

Güvenlik Varsayımları

Algoritmanın güvenliği, hesaplamaya dayalı iki varsayıma dayanır:

- genel kod çözme probleminin ortalama olarak zorluğu,
- açık anahtarı (üreteç matrisi) rastgele bir matristen ayırt etmenin zorluğu.

Zorluktan kasıt, uygun boyutta seçimler yapıldığında bu problemlerin zor olması ve sınırlı hesaplama gücüne sahip her düşmana karşı sistemin güvenli olmasıdır. Ayırt edilemezlik varsayımının geçerli olduğu kod ailelerini kullanmak ise kod tabanlı kriptografide temel meselelerden biridir. Aksi takdirde sistemin güvenliği bozulur.

Sistemin Tanımı

$\mathbb{F}_{2^m}$ üzerinde t dereceli indirgenemez her polinoma karşılık gelen $n = 2^m$ uzunluğunda, $k \geq n - tm$ boyutunda indirgenemez bir ikili Goppa kodu mevcuttur. Bu kod t ve t den az sayıda olan hataları düzeltme kapasitesine sahiptir. Ayrıca bu kodlar için hızlı çalışan bir kod çözme algoritması vardır [15].

Sistemi tasarlayan kişi n ve t değerlerini seçer. Daha sonra $\mathbb{F}_{2^m}$ üzerinde rastgele t dereceli indirgenemez bir polinom seçer. Rastgele seçilen t dereceli bir polinomun indirgenemez olma ihtimali yaklaşık $1/t$ olmakla birlikte indirgenemezliği test etmek için hızlı bir algoritma var olduğundan bu seçimi yapmak kolaydır [3]. Bir sonraki adımda, kod için $k \times n$ boyutunda G üreteç matrisi oluşturulur. G matrisini gizlemek için $k \times k$ boyutunda rastgele, tekil olmayan, yoğun bir S matrisi ve $n \times n$ boyutunda rastgele bir permütasyon matrisi seçilir. $G' = SGP$ matrisi, G matrisi tarafından üretilen kod ile aynı en kısa uzaklığa ve orana sahip bir lineer kod üretir. Herkes tarafından bilinen bu G' matrisine açık üreteç matrisi adı verilir. Sistemin tasarlayıcısı, kendisiyle güvenli bir şekilde iletişim kurmak isteyenlerin kullanması için oluşturduğu şifreleme algoritmasını yayınlar.

Şifreleme Algoritması

1. Şifrelenecek metin k bitlik bloklar halinde parçalanır.
2. $\mathbf{u}$ bloğundan $\mathbf{x} = \mathbf{u}G' + \mathbf{z}$ şifreli metni elde edilir.
3. G' açık üreteç matrisi ve $\mathbf{z}$ rastgele üretilmiş n uzunluğunda ve t ağırlığında bir vektör.

Şifre Çözme Algoritması

1. $\mathbf{x}' = \mathbf{x}P^{-1}$ hesaplanır. (P^{-1}: P permütasyon matrisinin tersi).
2. $\mathbf{x}'$ seçilen Goppa kodda bir kod kelimesi olur
3. Patterson algoritması kullanılarak $\mathbf{u}S = \mathbf{u}'$ hesaplanır.
4. Son olarak; $\mathbf{u} = \mathbf{u}'S^{-1}$ açık mesajına ulaşılır.

4. KLASİK MCELIECE

Bu bölümde Klasik McEliece kriptosistemiyle ilgili özet bilgilere yer verilecektir. Detaylı bilgi için [4] nolu kaynak incelenebilir.

Klasik McEliece kriptosisteminde kullanıcılar gizli bilgiyi sızdırmamak için dikkatli olmalıdırlar. Klasik McEliece anahtar kapsülleme mekanizmasında başarı ya da başarısızlık arasındaki ayrım gizli tutulmaktadır. 2. adımda $\perp$ çıktısı alındığı an hesaplama durdurulursa bu ayrım zaman içerisinde ortaya çıkacaktır. Bu nedenle sistem uyarlayıcılara 2. adımda mutlaka bir $\mathbf{c} \in \mathbb{F}_2^n$ seçmeleri önerilir.

Esasen anahtar üretimi, kodlama ve kod çözme algoritmaları Klasik McEliece kriptosisteminin bir versiyonu olan Niederreiter [14] kriptosistemi ile aynıdır. Niederreiter tarafından kullanılan GRS (Generalized Reed-Solomon) kod ailesinden farklı olarak bu algoritmada orijinal McEliece sisteminde kullanılan ikili Goppa kod ailesi kullanılmıştır.

2. adımda olduğu gibi 3. ve 5. adımlarda da başarı ile başarısızlık arasındaki ayrımı gizli tutmaya özen gösterilmelidir. Bu bilginin ortaya çıkmaması açısından algoritmadaki adımların her zaman aynı şekilde takip edilmesi önemlidir.

McEliece en eski sistem önerilerinin arasında yer alır ve RSA ile neredeyse aynı tarihte ortaya çıkmıştır. Günümüze kadar RSA ciddi güvenlik kayıpları yaşarken, McEliece sistemi kuantum sonrası rakipleri tarafından erişilemeyen güvenliğini korumayı başarmıştır. 40 yıldır yazılan atak makalelerinin sonucunda anahtar boyutunda yapılması gereken değişim %1'in altındadır. Bu da kalıcı bir güvenlik örneğidir.

Girdi düzgün rastgele bir şekilde seçildiğinde, verilen açık anahtarla şifreli metinden açık metne ulaşmanın zorluğu sistemin tek yönlülüğü olarak ifade edilir. Yani bu durumda girdiden şifreli metne olan dönüşümü ters yöne çevirmek oldukça zordur.

Yukarıda bahsi geçen makalelerin hepsi en etkili atak stratejisi olarak bilinen ISD (information-set decoding) yöntemini kullanmaktadırlar. Bu yöntem üreteç matrisin özel bir yapısından faydalanmak yerine, verilen düzgün rastgele matrisi ve şifreli metni kullanarak düşük ağırlıklı **e** hata vektörünü elde eder. McEliece sisteminde kullanılan G matrislerinin düzgün rastgele matris özellikleri taşıdığı deneylerle de gösterilmiştir. Ayrıca McEliece sistemindeki G açık anahtarından gizli anahtarı elde eden birçok atak makalesi de mevcuttur. Gizli anahtarı elde etmek atağı yapan kişiye aynı zamanda kod çözme algoritmasını kullanma imkanı da sunduğundan bu durum sistemin tek yönlülüğünü de bozar. Fakat bu atak türü ISD ataklarına göre oldukça yavaştır.

McEliece sistemindeki ikili Goppa kodların yerine birçok kod ailesi önerilmiştir. Genellikle daha küçük anahtarlara sahip olmaları amacıyla ortaya çıkmış olsalar da önerilen bu makalelerin çoğunun gizli anahtarın hızlıca ele geçirilmesine, tek yönlü fonksiyonun hızlıca ters çevrilmesine sebep olma gibi zaafiyetleri bulunmuştur. Küçük anahtar üreten sistemlerden bazıları kırılmamış olmakla birlikte, Klasik McEliece sisteminde geleneksel ve üzerinde iyi çalışılmış olan ikili Goppa kodlar tercih edilmiştir.

Sistemin Avantaj ve Dezavantajları

- Bu sistemin en belirgin avantajı güvenliğidir.
- Anahtar üretimi çok hızlı değildir. Anahtarın üretim ve dağılım maliyetini karşılamak için uygulamaların her açık anahtarı yeterince uzun süre kullanması gerekmektedir.
- Buna karşılık ikili vektörlerin ve ikili matris-vektör çarpma işlemlerinin basitliği sayesinde kapsülleme ve kapsülden çıkarma işlemleri yazılımda makul derecede, donanımda ise etkileyici bir şekilde hızlıdır. Anahtar üretimi de donanımda oldukça hızlıdır.
- Ayrıca şifreli metinler kuantum sonrası kriptografi için oldukça küçük boyuttadır. Yüksek güvenlik seviyesi için önerilen parametre setinde bu sayı 256 baytın altındadır (bkz. Ek3). Tüm sistem içerisinde ne kadar sıklıkla gönderildiklerine bağlı olarak, küçük şifreli metin boyutu geniş anahtar

boyutundan daha önemli hale gelebilir. Sistem parametreleri daha küçük şifreli metinler için tekrar düzenlenebilir.

4.1 Gösterimler

n	Kod uzunluğu
k	Kod boyutu
t	Garanti edilen hata düzeltme kapasitesi
q^m	Kullanılan cismin eleman sayısı
m	Pozitif bir tam sayı
$H_\ell(.)$	Kriptografik özet (hash) fonksiyonu
ℓ	Özet fonksiyonunun çıktı uzunluğu (bit)
g	$\mathbb{F}_{q^m}[x]$ polinom halkasında bir polinom
α_i	$\mathbb{F}_{q^m}$ sonlu cisminin bir elemanı
$\mathbf{\Gamma}$	$(g, \alpha_1, \ldots, \alpha_n)$
$\mathbf{s}$	n uzunluğunda bir bit dizgisi
$(\mathbf{s}, \mathbf{\Gamma})$	Klasik McEliece kriptosisteminde bir gizli anahtar
T	Klasik McEliece kriptosisteminde bir açık anahtar
$\mathbf{e}$	n uzunluğunda ve t ağırlığında bir bit dizgisi
$\mathbf{c}$	Bir oturumluk anahtarı kapsülleyen şifreli metin
$\mathbf{c}_0$	$n-k$ uzunluğunda bir bit dizgisi
$\mathbf{c}_1$	ℓ uzunluğunda bir bit dizgisi

4.2 Parametreler

1. Pozitif bir m tam sayısı ve $q = 2$.
2. $n \leq 2^m$ olacak şekilde bir n pozitif tam sayısı.
3. $t \geq 2$ ve $mt < n$ olacak şekilde pozitif bir t tam sayısı.
4. $k = n - mt$.
5. Derecesi m olan monik indirgenemez bir $f(x) \in \mathbb{F}_2[x]$ polinomu. Bu polinom ile $\mathbb{F}_{2^m}$ sonlu cisminin $\mathbb{F}_2[x]/f(x)$ gösterimi belirlenir.

6. Pozitif bir ℓ tam sayısı.

4.3 Anahtar Üretimi

1. t dereceli rastgele monik indirgenemez bir $g(z) \in \mathbb{F}_{q^m}[z]$ polinomu üretilir.

2. $\mathbb{F}_{q^m}$ cisminin n tane farklı elemanından oluşan $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ dizisi seçilir.

3. $i = 1, \ldots, t$ ve $j = 1, \ldots, n$ için $h_{i,j} = \alpha_j^{i-1} / g(\alpha_j)$ hesaplanır.

4. $\mathbb{F}_q$ üzerinde $t \times n$ boyutunda $\tilde{H} = \{h_{i,j}\}$ matrisi üretilir.

5. $\tilde{H}$ matrisinin her bir $c_0 + c_1 z + \cdots + c_{m-1} z^{m-1}$ girdisi t bitlik $c_0, c_1, \ldots, c_{m-1}$ sütunlarıyla değiştirilerek $\mathbb{F}_2$ üzerinde $mt \times n$ boyutunda $\hat{H}$ matrisi üretilir.

6. $\hat{H}$ matrisine Gauss eleme yöntemi uygulanarak $(I_{n-k} \mid T)$ sistematik formuna indirgenir. Gauss eleme yöntemiyle $\hat{H}$ sistematik forma dönüştürülemezse 1. adıma geri dönülür.

7. Rastgele n bitlik bir $\mathbf{s}$ karakter dizisi üretilir.

8. $\mathbf{\Gamma} = (g, \alpha_1, \alpha_2, \ldots, \alpha_n)$ olmak üzere $(\mathbf{s}, \mathbf{\Gamma})$ gizli anahtarı ve T açık anahtarı oluşturulur.

$\mathbf{\Gamma} = (g, \alpha_1, \alpha_2, \ldots, \alpha_n)$, n uzunluğunda ve $k = n - mt$ boyutunda ikili bir Goppa kod tanımlar. $H = (I_{n-k} \mid T)$ ise bu Goppa kodun eşlik denetim matrisidir.

4.4 Kodlama Algoritması

Girdiler: t ağırlığında bir $\mathbf{e} \in \mathbb{F}_2^n$ sütun vektörü ve bir $T \in \mathbb{F}_2^{(n-k) \times k}$ açık anahtarı.

1. $H = (I_{n-k} \mid T)$ tanımlanır.

2. $\mathbf{c}_0 = H\mathbf{e} \in \mathbb{F}_2^{n-k}$ hesaplanır.

4.5 Kod Çözme Algoritması

Kod çözme algoritması, $\mathbf{c}_0 \in \mathbb{F}_2^{n-k}$ şifreli metnini $\mathbf{c}_0 = H\mathbf{e}$ olacak şekilde t ağırlıklı $\mathbf{e}$ hata vektörüne çevirir. Böyle bir vektörün varolmaması halinde hata verir.
Girdiler: $\mathbf{c}_0 \in \mathbb{F}_2^{n-k}$ ve $(\mathbf{s}, \mathbf{\Gamma})$ açık anahtarı.

1. $H = (I_{n-k} \mid T)$ ve $\mathbf{c}_0 = H\mathbf{e}$ olmak üzere t ağırlıklı $\mathbf{e} \in \mathbb{F}_2^n$ hata vektörü mevcut ise kod çözme algoritması $\mathbf{e}$ çıktısını verir.

   (a) $\mathbf{c}_0$, sonuna k tane 0 eklenerek $\mathbf{v} = (\mathbf{c}_0, 0, \ldots, 0) \in \mathbb{F}_2^n$ vektörüne genişletilir.

   (b) $\mathbf{\Gamma}$ ile tanımlanan Goppa kodda $\mathbf{v}$ vektöründen $\leq t$ uzaklığındaki tek kod kelimesi $\mathbf{c}$ bulunur. Eğer böyle bir kod kelimesi mevcut değilse $\perp$ çıktısı verilir.

   - Sendrom aşağıdaki gibi hesaplanır:
   $$s(z) = \sum_{i=1}^{n} \frac{y_i}{z - \alpha_i}.$$
   - Aşağıdaki adımlarla $\sigma(z)$ polinomu elde edilir:
     - Öklid algoritması kullanılarak aşağıdaki denkliği sağlayan $h(z)$ polinomu bulunur:
     $$s(z)h(z) \equiv 1 \pmod{g(z)}.$$
     $h(z) = z$ ise $\sigma(z) = z$ sonucuna ulaşılır.
     - Aşağıdaki denkliği sağlayan $d(z)$ polinomu hesaplanır:
     $$d^2(z) \equiv h(z) + z \pmod{g(z)}.$$
     - $b(z)$ en küçük dereceli olmak üzere aşağıdaki denkliği sağlayan $a(z)$ ve $b(z)$ polinomları bulunur:
     $$d(z)b(z) \equiv a(z) \pmod{g(z)}.$$
     - $\sigma(z) = a^2(z) + b^2(z)z$ olarak belirlenir.
   - Hataların bulunduğu bitlerin kümesi $B = \{i \mid \sigma(\alpha_i) = 0\}$ oluşturulur.
   - $\mathbf{e} = (e_1, \ldots, e_n)$ hata vektörü $i \in B$ için $e_i = 1$ ve geri kalanlar için $e_i = 0$ olacak şekilde belirlenir.
   - $\mathbf{c} = \mathbf{y} - \mathbf{e}$ olarak belirlenir.

2. Eğer $\mathbf{c}_0 = H\mathbf{e}$ olacak şekilde t ağırlıklı bir $\mathbf{e} \in \mathbb{F}_2^n$ mevcut değil ise kod çözme algoritması $\perp$ çıktısı verir.

4.6 Kapsülleme Algoritması

Çıktı olarak tek kullanımlık K anahtarı ve $\mathbf{c}$ şifreli metni elde edilir.

1. t ağırlığında rastgele bir $\mathbf{e} \in \mathbb{F}_2^n$ vektörü üretilir.
2. $\mathbf{e}$ vektörü ve T açık anahtarı ile kodlama algoritması kullanılarak $\mathbf{c}_0$ hesaplanır.
3. $\mathbf{c}_1 = H_\ell(2, \mathbf{e})$ hesaplanır. $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1)$ oluşturulur.
4. $K = H_\ell(1, \mathbf{e}, \mathbf{c})$ hesaplanır.
5. Tek oturumluk K anahtarı ve $\mathbf{c}$ şifreli metni çıktısı alınır.

4.7 Kapsülden Çıkarma Algoritması

Alıcı, $\mathbf{c}$ şifreli metninden tek kullanımlık K anahtarını kapsülden çıkarır.

1. $\mathbf{c}$ şifreli metni $\mathbf{c}_0 \in \mathbb{F}_2^{n-k}$ ve $\mathbf{c}_1 \in \mathbb{F}_2^\ell$ olmak üzere $(\mathbf{c}_0, \mathbf{c}_1)$ şeklinde parçalanır.
2. $b \leftarrow 1$ ayarlanır.
3. $\mathbf{c}_0$ ve $\mathbf{\Gamma}$ özel anahtarı üzerinde kod çözme algoritması kullanılarak $\mathbf{e}$ hesaplanır. Algoritma $\perp$ hata çıktısı verirse $\mathbf{e} \leftarrow s$ ve $b \leftarrow 0$ olarak ayarlanır.
4. $K = H_\ell(b, \mathbf{e}, \mathbf{c})$ hesaplanır.
5. Tek oturumluk K anahtarı çıktısı alınır.

4.8 Örnek.

4.8.1 Parametreler

Bölüm 4.2 de verilen şartlara uygun parametreler seçilir:

1. $q = 2$ ve $m = 4$ olmak üzere $\mathbb{F}_{2^4}$ cismi.
2. $t = 2$, kod uzunluğu $n = 12$, kod boyutu $k = 4$ ve $\ell = 12$ olsun.
3. $f(x) = x^4 + x + 1$ olsun.
4. α ise $f(x)$ polinomunun ilkel kökü olsun.

4.8.2 Anahtar üretimi

Bölüm 4.3 de verilen adımlara uygun şekilde bir anahtar çifti oluşturulur:

1. $g(z) = z^2 + z + 1$ üretilir.
2. $(\alpha_1, \alpha_2, \ldots, \alpha_n) = (\alpha, \alpha^{13}, \alpha^6, \alpha^3, \alpha^2, \alpha^{11}, \alpha^{14}, \alpha^4, \alpha^7, \alpha^9, \alpha^8, \alpha^{12})$ seçilir.
3. $\tilde{H} = \{h_{i,j}\}$ matrisini elde etmek için;
   $h_{1,1} = \frac{(\alpha_1)^0}{g(\alpha_1)} = \frac{1}{\alpha^2+\alpha+1} = \frac{1}{\alpha^{10}} = \alpha^{-10} = \alpha^5$,

   $h_{1,2} = \frac{(\alpha_2)^0}{g(\alpha_2)} = \frac{1}{\alpha^{11}+\alpha^{13}+1} = \frac{1}{\alpha} = \alpha^{-1} = \alpha^{14}$,

   $h_{2,1} = \frac{(\alpha_1)^1}{g(\alpha_1)} = \frac{\alpha}{\alpha^2+\alpha+1} = \frac{\alpha}{\alpha^{10}} = \alpha^{-9} = \alpha^6$,

   $h_{2,2} = \frac{(\alpha_2)^1}{g(\alpha_2)} = \frac{\alpha^{13}}{\alpha^{11}+\alpha^{13}+1} = \frac{\alpha^{13}}{\alpha} = \alpha^{12}$ vb. hesaplamalar yapılır.
4. $\tilde{H}$ matrisi oluşturulur:

$$\tilde{H} = \begin{bmatrix} \alpha^5 & \alpha^{14} & \alpha^{14} & \alpha^7 & \alpha^{10} & \alpha^{13} & \alpha^7 & \alpha^5 & \alpha^{11} & \alpha^{11} & \alpha^{10} & \alpha^{13} \\ \alpha^6 & \alpha^{12} & \alpha^5 & \alpha^{10} & \alpha^{12} & \alpha^9 & \alpha^6 & \alpha^9 & \alpha^3 & \alpha^5 & \alpha^3 & \alpha^{10} \end{bmatrix}$$

5. $\hat{H}$ matrisi oluşturulur:

$$\hat{H} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

6. Gauss eleme yöntemi uygulanarak $\hat{H}$ matrisi;

$$H = (I_{n-k} \mid T) = \left[\begin{array}{cccccccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right]$$

   sistematik formuna dönüştürülür.

7. $\mathbf{s} = (0,0,1,0,1,1,0,1,0,0,1,0)$ rastgele üretilir.

8. $\mathbf{\Gamma}$ aşağıdaki gibi hesaplanır:

$$\begin{aligned} \mathbf{\Gamma} &= (g, \alpha_1, \alpha_2, \ldots, \alpha_n) \\ &= (x^2+x+1, \alpha, \alpha^{13}, \alpha^6, \alpha^3, \alpha^2, \alpha^{11}, \alpha^{14}, \alpha^4, \alpha^7, \alpha^9, \alpha^8, \alpha^{12}). \end{aligned}$$

9. Gizli anahtar,
   $(\mathbf{s}, \mathbf{\Gamma}) = (0,0,1,0,1,1,0,1,0,0,1,0, x^2+x+1, \alpha, \alpha^{13}, \alpha^6, \alpha^3, \alpha^2, \alpha^{11}, \alpha^{14}, \alpha^4, \alpha^7, \alpha^9, \alpha^8, \alpha^{12})$
   ve

$$T = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

açık anahtarı çıktıları oluşturulur.

4.8.3 Kodlama algoritması

Bölüm 4.4 de verilen adımlara uygun şekilde kodlama yapılır:

1. Kodlama algoritması için girdi olarak T açık anahtarı ve $t = 2$ ağırlığında bir $\mathbf{e} = (0,1,0,0,0,0,0,0,0,1,0,0)$ hata vektörü alınır.

2. $\mathbf{c}_0$ aşağıdaki şekilde hesaplanır:

$$\mathbf{c}_0 = H\mathbf{e}$$

$$= \left[\begin{array}{cccccccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right] \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

4.8.4 Kod çözme algoritması

Bölüm 4.5 de verilen adımlara uygun şekilde kod çözülür:

1. $\mathbf{v} = (\mathbf{c}_0, 0, \ldots, 0) = (1,0,0,0,1,0,0,1,0,0,0,0) \in \mathbb{F}_2^n$ olarak belirlenir.

2. $s(z)$ aşağıdaki gibi hesaplanır:

$$\begin{aligned} s(z) &= \sum_{i=1}^{12} \frac{v_i}{z-\alpha_i} \\ &= \frac{1}{z-\alpha} + \frac{1}{z-\alpha^2} + \frac{1}{z-\alpha^4} \\ &\equiv (\alpha^9 + \alpha^3 + \alpha^6) + (\alpha^5 + \alpha^{10} + \alpha^5)z \\ &= \alpha^{11} + \alpha^{10} z \end{aligned}$$

3. $s(z)h(z) \equiv 1 \pmod{g(z)}$ olmak üzere $h(z) = \alpha^{10}z + \alpha^{14}$ alınır.

4. $d^2(z) \equiv h(z) + z \pmod{g(z)} \equiv \alpha^5 z + \alpha^{14} \pmod{g(z)}$ olacak şekilde $d(z) = \alpha^{10}z + \alpha^6$ polinomu hesaplanır.

5. $d(z)b(z) \equiv a(z) \pmod{g(z)}$ olacak şekilde $b(z) = 1$ ve $a(z) = d(z) = \alpha^{10}z + \alpha^6$ alınır.

6. $\sigma(z)$ aşağıdaki gibi hesaplanır:

$$\begin{aligned} \sigma(z) &= a^2(z) + b^2(z)z \\ &= \alpha^5 z^2 + z + \alpha^{12} \end{aligned}$$

7. $\alpha_2 = \alpha^{13}$ ve $\alpha_{10} = \alpha^9$ olduğundan $B = \{i \mid \sigma(\alpha_i) = 0\} = \{2, 10\}$ elde edilir.

8. Hata vektörü $\mathbf{e} = (010000000100)$ olarak bulunur.

5. MCNIE

Bu bölümde McNie kriptosistemiyle ilgili özet bilgilere yer verilecektir. Detaylı bilgi için [7] nolu kaynak incelenebilir.

McNie sisteminde küçük boyutta anahtarlar elde edebilmek için $\ell > n-k$ olma koşulu ile özellikle yarı devirli LRPC kodlar kullanılır. Her kod kelimesinin n_0 kez kaydırılmasıyla yeni bir kod kelimesinin elde edildiği durumlarda, yarı devirli bir (n,k) lineer blok kodu $n = mn_0$ ve $k = mk_0$ olacak şekilde ifade edilir. $\mathbb{F}_q$ üzerinde herhangi bir (n,k) yarı devirli kodu (mn_0, mk_0) koduna denktir ve $m \times m$ boyutunda devirli matrislerden oluşan $mk_0 \times mn_0$ boyutlu G üreteç matrisine sahiptir.

$$G = \begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & \dots & C_{1,n_0} \\ C_{2,1} & C_{2,2} & C_{2,3} & \dots & C_{2,n_0} \\ \vdots & \vdots & \vdots & & \vdots \\ C_{k_0,1} & C_{k_0,2} & C_{k_0,3} & \dots & C_{k_0,n_0} \end{bmatrix}$$

Çevrimsel (circulant) bir C matrisi ilk satırındaki elemanlardan oluşturulan

$$c(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{m-1} x^{m-1}$$

polinomu ile ifade edilir. Çevrimsel C_i matrisleri ile $c_i(x)$ polinomları arasında bire bir eşleme vardır. Devirli iki matrisin toplamı ve çarpımı yine devirlidir.
3 ve 4-yarı devirli LRPC kodlar için özel parametreler aşağıda verilmiştir.

- **2-yarı devirli:** Bu durumda kod uzunluğu n çift olmalıdır. H eşlik denetim matrisi ise çifte çevrimsel (double-circulant) $\frac{n}{2} \times n$ boyutunda bir matristir. $\ell > \frac{n}{2}$ olduğu için $\ell \times n$ boyutunda bir G' matrisi çifte çevrimsel olamaz. Dolayısıyla $\ell \times \frac{n}{2}$ boyutunda $F = G'H^T S$ matrisi de çevrimsel olamaz. Bu durumda G' ve F tek bir vektör olarak tanımlanamaz. Bu ise G' ve F için büyük boyutlarda anahtarlara sebep olacağından 2-yarı devirli durumu gözardı edilir.

- **3-yarı devirli:** Bu durumda n sayısı 3 ile bölünebilmelidir. Dolayısıyla $\ell = k = \frac{2n}{3}$ olur. Açık anahtarlar ise $\frac{2n}{3} \times n$ boyutunda bir G' matrisi ve $\frac{2n}{3} \times \frac{n}{3}$ boyutunda bir F matrisidir.

- **4-yarı devirli:** Bu durumda ise n sayısı 4 ile bölünebilmelidir. Dolayısıyla $k = \frac{n}{2}$ ve $\ell = \frac{3n}{4}$ olur. Açık anahtarlar ise $\frac{3n}{4} \times n$ boyutunda bir G' matrisi ve $\frac{3n}{4} \times \frac{n}{2}$ boyutunda bir F matrisidir.

- **s-yarı devirli:** $s \geq 5$ durumunda 3 ve 4-yarı devirli LRPC kodlardakinden daha kısa boyutta anahtarlar elde edilemediğinden bu duruma ait parametreler dahil edilmemiştir.

Rastgele yapıda ilk rank metrik tabanlı GRSTZ [5] kriptosisteminin hala güvenli olduğu düşünülmektedir. Bu kriptosistem QC-LRPC kodlarla birlikte uygulandığında anahtar boyutları da küçülmektedir. Bu sebeple McNie kriptosistemi QC-LRPC kod tabanlı olacak şekilde tasarlanmıştır.

GRSTZ kriptosistemi gizli anahtar olarak düşük ranklı H eşlik denetim matrisini kullanmaktadır. GRSTZ sistemi üzerinde bilinen yapısal ataklar H matrisine erişebilmek için kodun dualında düşük ağırlıklı bir kod kelimesi bulmaya çalışırlar.
McNie ise GRSTZ kriptosisteminin kod çözme algoritmasından ve çeşitli yarı devirli LRPC kodlardan yararlanır. Bu sayede McNie kriptosistemi de yapısal ve ISD ataklara karşı GRSTZ sistemi kadar güvenli hale gelir. Ayrıca McNie, gizli ve açık anahtarlarında iki farklı matrise yer verdiğinden G' açık anahtarı H gizli anahtarı hakkında hiçbir bilgi içermez ve böylelikle McNie kriptositemini kırmak GRSTZ sistemini kırmaktan daha zor bir hal alır.

McNie şifreleme sistemi McEliece ve Niederreiter kriptosistemlerinin bir kombinasyonudur. $\mathbf{c}_1 = \mathbf{m}G' + \mathbf{e}$ McEliece kriptosistemindeki şifreli metne benzerken, $\mathbf{c}_2 = \mathbf{m}F$ ise Niederreiter kriptosistemindeki şifreli metne benzemektedir.
G üreteç matrisli, H eşlik denetim matrisli ve $G' = SGP$ olacak şekilde bir kod düşünelim. Bu durumda $F = (SGP)P^{-1}H^TS = 0 \in \mathbb{F}_{q^m}^{\ell \times (n-k)}$ elde edilir. Dolayısıyla $\mathbf{c}_2 = \mathbf{0} \in \mathbb{F}_{q^m}^{n-k}$ ve $\mathbf{c}_1 = \mathbf{m}G' + \mathbf{e} = \mathbf{m}SGP + \mathbf{e}$ McEliece kriptosistemindeki şifreli metin olur. Bu sebeple McEliece kriptosistemini kırmak McNie sistemini kırmaktan daha zor değildir.

Sistemin Avantaj ve Dezavantajları

- G' açık anahtarının boyutu ℓ, H eşlik denetim matrisinin boyutu $n-k$ dan daha büyük olmalıdır. Aksi takdirde, atak yapan kişi $\mathbf{c}_2 = \mathbf{m}F$ şifreli metninden gizli mesaj vektörünü elde edebilir. Burada $\mathbf{m} = (m_1, m_2, \ldots, m_\ell) \in \mathbb{F}_{q^m}^{\ell}$ ve F boyutları $\ell \times (n-k)$ olan bir matristir. $\ell \leq n-k$ olduğunda, ℓ bilinmeyenli lineer sistemden tek bir çözüm elde edilir. Bu çözüm ise $\mathbf{m}$ açık mesajıdır. $\ell > n-k$ olduğunda, $\mathbf{m}$ açık mesajının $\mathbf{c}_2 = \mathbf{m}F$ eşitliğinden gelen $q^{m(\ell-n+k)}$ olası çözümü vardır.

- $H_0 = P^{-1}H^T S$ olsun. Dolayısıyla $F = G'P^{-1}H^T S = G'H_0$ olur ve H_0 ın $q^{m(n-\ell)(n-k)}$ olası çözümü vardır.

- Şifrelemede rastgele bir kod kullanıldığından McNie yapısal ve ISD ataklarına karşı güvenlidir.

- McNie aynı parametrelerle GRSTZ kriptosisteminden daha fazla güvenlik sağlamaktadır.

- GRSTZ kriptosistemi McNie sisteminin özel bir durumu olarak gösterilebilir. H eşlik denetim matrisine ve G üreteç matrisine sahip bir yarı devirli LRPC kod olsun. Tersinir bir R matrisi için $G' = RG$ ve P birim matris ise $F = RGH^T S = 0$ olur. Dolayısıyla $\mathbf{c}_2 = 0$ ve $\mathbf{c}_1 = \mathbf{m}RG + \mathbf{e}$ GRSTZ kriptosistemindeki şifreli metin ile aynı olur. McNie, GRSTZ kriptosistemi ile aynı kod çözme algoritmasını kullanmaktadır. Buradan McNie sistemine karşı bir atak yapmanın GRSTZ kriptosistemine saldırmaktan daha zor olduğu sonucuna varılır.

- Diğer şifreleme sistemlerine nazaran McNie, güvenlik seviyesi yükseldikçe daha kademeli olarak artış gösteren küçük boyutta açık anahtarlara sahiptir.

- McNie kriptosistemi, bilinen blok kodların birçok türünü kullanabilir. Bu kodlar tabanlı McEliece kriptosistemi kırılmış olduğundan, McNie yapısal ve (ISD) ataklarına karşı daha dayanıklı görünen rastgele bir kod kullanır. Bu sebeple McNie sistemine çalışmak yakın gelecekte birçok araştırma alanı açacaktır.

- LRPC kod çözümü olasılıksal olduğundan hata ihtimali sıfır değildir. Bu problem parametreler ayarlanarak mümkün olduğu kadar azaltılabilir ancak bu

durum anahtar boyutunun büyümesine sebep olur. McNie kriptosisteminde önerilen parametreler, anahtar boyutu ve düşük başarısızlık oranları en uygun olacak şekilde ayarlanmıştır.

- Aynı mesajın şifrelenmesi farklı şifreli metinlerle sonuçlandığından kod çözme başarısızlığı durumunda alıcı mesajın tekrar gönderilmesini isteyebilir. Bu durum McNie sisteminin güvenliği için herhangi bir tehdit oluşturmaz.

- McNie üçüncü şahsın açık mesajı bilmeden şifreli metni birkaç kez gönderebildiği ve bu sayede kod çözmedeki başarısızlık ihtimalinin düştüğü uygulamalar için kullanışlı olacaktır. Örneğin, 2^{-17} olan bir başarısızlık ihtimali mesajın iki kez gönderilmesiyle birlikte 2^{-34} e kadar düşürülebilir.

5.1 Parametreler

1. $(n-k) \times n$ boyutunda bir H matrisi .
2. $n \times n$ boyutunda bir P matrisi.
3. $(n-k) \times (n-k)$ boyutunda bir S matrisi.
4. $\ell \times n$ boyutunda bir G' matrisi.
5. blk : blok matris boyutu.
6. $\mathbb{F}_{q^m}$ sonlu cismi üzerinde $\ell \times (n-k)$ boyutunda bir F matrisi.
7. q : bir asalın kuvveti.
8. $\ell > n-k$.

5.2 Anahtar Üretimi

1. $\mathbb{F}_{q^m}$ sonlu cismi üzerinde $(n-k) \times n$ boyutunda bir H eşlik denetim matrisi üretilir.
2. Elemanları $\mathbb{F}_{q^m}$ cisminden seçilen $(n-k) \times (n-k)$ boyutunda bir S ve $n \times n$ boyutunda bir P matrisi rastgele üretilir.

3. $\mathbb{F}_{q^m}$ üzerinde $\ell \times n$ boyutunda rastgele bir G' matrisi üretilir.

4. $\ell \times (n-k)$ boyutunda $F = G'P^{-1}H^T S$ matrisi hesaplanır.

5. G' ve F açık anahtar olarak yayınlanır. H, S ve P gizli anahtar olarak tutulur.

5.3 Şifreleme Algoritması

$\mathbb{F}_{q^m}$ üzerinde ℓ uzunluğunda $\mathbf{m}$ mesaj vektörü göndermek için:

1. $\mathbb{F}_{q^m}$ üzerinde n uzunluğunda rastgele bir $\mathbf{e}$ hata vektörü üretilir. (Uygun kod çözme algoritmasıyla çözülebilmesi için ağırlığı en fazla r olmalı).

2. n uzunluğunda $\mathbf{c}_1 = \mathbf{m}G' + \mathbf{e}$ vektörü hesaplanır.

3. $n-k$ uzunluğunda $\mathbf{c}_2 = \mathbf{m}F$ vektörü hesaplanır.

4. $2n-k$ uzunluğunda $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ şifreli metni oluşturulur.

5.4 Şifre Çözme Algoritması

$\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ şifreli metni çözmek için:

1. $\mathbf{s}' = \mathbf{c}_1 P^{-1}H^T - \mathbf{c}_2 S^{-1} = (\mathbf{m}G' + e)P^{-1}H^T - (\mathbf{m}G'P^{-1}H^T S)S^{-1} = \mathbf{e}P^{-1}H^T$ hesaplanır.

2. $\mathbf{s}'$ vektörüne Φ_H kod çözme algoritması uygulanarak $\mathbf{e}' = \mathbf{e}P^{-1}$ hesaplanır.

   (a) $\mathbf{s}' = (s_1, \ldots, s_{n-k})$ sendrom vektörü olmak üzere $S = \langle s_1, \ldots, s_{n-k} \rangle$ sendrom uzayıdır.

   (b) S vektör uzayının tüm üreteçleri F_i^{-1} ile çarpılarak $S_i = F_i^{-1}S$ alt uzayı tanımlanır. Hata vektörünün (support) u olan $E = S_1 \cap S_2 \cap \ldots \cap S_d$ hesaplanır ve E nin bir $\{E_1, E_2, \ldots, E_r\}$ bazı oluşturulur.

   (c) $1 \leq i \leq n$ olmak üzere e_i ler hata (support)unda $e_i = \sum_{i=1}^{n} e_{ij}E_j$ olacak şekilde yazılır ve $H\mathbf{e}^T = \mathbf{s}'$ sistemi çözülür. Burada $H\mathbf{e}^T$ denklemleri ve sendrom vektörünün koordinatları s_i ler $P = \langle E.F \rangle$ çarpım uzayının

elemanları olarak $\{F_1E_1,\ldots,F_1E_r,\ldots,F_dE_1,\ldots,F_dE_r\}$ bazında yazılır. Bu sistemde nr bilinmeyen (e_{ij} ler) ve $(n-k).rd$ denklem vardır.

3. $\mathbf{e}=\mathbf{e}'P$ hata vektörü elde edilir.

4. $\mathbf{m}G'=\mathbf{c}_1-\mathbf{e}$ lineer sistemi çözülerek $\mathbf{m}$ gizli mesajına ulaşılır.

G' standart formda bir matris olduğunda, $\mathbf{c}_1-\mathbf{e}$ nin ilk ℓ elemanına bakmak $\mathbf{m}$ mesajını bulmak için yeterlidir.

5.5 Örnek.

5.5.1 Parametreler

Bölüm 5.1 de verilen şartlara uygun parametreler seçilir:

1. $q=2$ ve $m=1$ olmak üzere $\mathbb{F}_2$ cismi, $\ell=4$, kod uzunluğu $n=6$ ve kod boyutu $k=3$.

5.5.2 Anahtar üretimi

Bölüm 5.2 de verilen adımlara uygun şekilde anahtar üretimi yapılır:

1. $\mathbb{F}_2$ üzerinde $(n-k)\times n=3\times 6$ boyutunda H eşlik denetim matrisi üretilir:

$$H=\begin{bmatrix}1&0&0&1&0&1\\0&0&1&1&0&0\\1&1&0&0&1&0\end{bmatrix}$$

2. Elemanları $\mathbb{F}_2$ cisminden seçilen $(n-k)\times(n-k)=3\times 3$ boyutunda bir

$$S=\begin{bmatrix}0&1&1\\1&0&1\\0&0&1\end{bmatrix}$$

matrisi ve $n\times n=6\times 6$ boyutunda bir

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

matrisi üretilir.

3. $\mathbb{F}_2$ üzerinde $\ell \times n = 4 \times 6$ boyutunda G' matrisi rastgele üretilir:

$$G' = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

4. $\ell \times (n-k) = 4 \times 3$ boyutunda F matrisi aşağıdaki gibi hesaplanır:

$$\begin{aligned} F &= G'P^{-1}H^TS \\ &= \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

5. G' ve F açık anahtar olarak yayınlanır. H, S ve P ise gizli anahtar olarak tutulur.

5.5.3 Şifreleme algoritması

$\ell = 4$ uzunluğunda $\mathbf{m} = (0,1,1,0)$ mesaj vektörünü göndermek için, Bölüm 5.3 de verilen adımlara uygun şekilde şifreleme yapılır:

1. $\mathbf{e} = (0,1,0,1,0,0)$ hata vektörü rastgele üretilir.

2. $\mathbf{c}_1$ hesaplanır:

$$\begin{aligned}
\mathbf{c}_1 &= \mathbf{m}G' + \mathbf{e} \\
&= \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \\
&= (1,1,1,1,0,1) + (0,1,0,1,0,0) \\
&= (1,0,1,0,0,1)
\end{aligned}$$

3. $\mathbf{c}_2$ hesaplanır:

$$\begin{aligned}
\mathbf{c}_2 &= \mathbf{m}F \\
&= \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= (0,1,1)
\end{aligned}$$

4. $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ şifreli metni oluşturulur:

$$\mathbf{c} = (1,0,1,0,0,1,0,1,1)$$

5.5.4 Şifre çözme algoritması

Bölüm 5.4 de verilen adımlara uygun şekilde $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ şifreli metni çözülürek $\mathbf{m}$ mesajına ulaşılır:

1. $\mathbf{s}'$ aşağıdaki gibi hesaplanır:

$$
\begin{aligned}
\mathbf{s}' &= \mathbf{c}_1 P^{-1} H^T - \mathbf{c}_2 S^{-1} \\
&= \mathbf{e} P^{-1} H^T \\
&= \begin{bmatrix} 1\,0\,1\,0\,0\,1 \end{bmatrix} \cdot \begin{bmatrix} 1\,0\,1 \\ 0\,0\,1 \\ 0\,1\,0 \\ 1\,1\,0 \\ 0\,0\,1 \\ 1\,0\,0 \end{bmatrix} - \begin{bmatrix} 0\,1\,1\,0 \end{bmatrix} \cdot \begin{bmatrix} 0\,1\,1\,0\,1\,0 \\ 1\,0\,1\,1\,0\,0 \\ 0\,1\,0\,0\,0\,1 \\ 1\,0\,1\,1\,0\,0 \end{bmatrix} \cdot \begin{bmatrix} 1\,0\,1 \\ 0\,0\,1 \\ 0\,1\,0 \\ 1\,1\,0 \\ 0\,0\,1 \\ 1\,0\,0 \end{bmatrix} \\
&= (0,1,1) - (1,0,0) \\
&= (1,1,1)
\end{aligned}
$$

2. $\mathbf{s}'$ vektörüne Φ_H kod çözme algoritması uygulanarak $\mathbf{e}' = \mathbf{e}P^{-1} = (0,1,0,1,0,0)$ hesaplanır.

3. $\mathbf{e}$ hata vektörü elde edilir:

$$
\begin{aligned}
\mathbf{e} &= \mathbf{e}' P \\
&= (0,1,0,1,0,0)
\end{aligned}
$$

4. $\mathbf{m}G'$ lineer sistemi çözülerek $\mathbf{m}$ mesajına ulaşılır:

$$
\begin{aligned}
\mathbf{m}G' &= \mathbf{c}_1 - \mathbf{e} \\
&= (1,0,1,0,0,1) - (0,1,0,1,0,0) \\
&= (1,1,1,1,0,1)
\end{aligned}
$$

5.6 3-Yarı Devirli LRPC Kodlar ile Açık Anahtarlı Şifreleme Sistemi

5.6.1 Anahtar üretimi

McNie sistemindeki anahtar boyutlarını düşürmek için devirli matrisler kullanılmış ve açık ve gizli anahtarlarında $\mathbb{F}_{q^m}$ üzerinde yarı devirli LRPC kodlar inşa edilmiştir.

1. n, 3'ün bir katı ve $blk = \frac{n}{3}$ olsun.

2. Girdileri $\mathbb{F}_{q^m}$ cisminin altuzayı olan $\mathbb{F}_q$ cisminden alınan $\frac{n}{3}$ uzunluğunda $\mathbf{h}_1$, $\mathbf{h}_2$ ve $\mathbf{h}_3$ vektörleri üretilir.

3. $i = 1,2,3$ için $\frac{n}{3} \times \frac{n}{3}$ boyutlarında, ilk satırları $\mathbf{h}_i$ vektörleri olan H_i çevrimsel matrisleri inşa edilir. Her satır bir önceki satırın döndürülmesiyle oluşturulur.

4. $\frac{n}{3} \times n$ boyutunda $H = \begin{bmatrix} H_1 & H_2 & H_3 \end{bmatrix}$ matrisi d ağırlığında bir LRPC kod için bir eşlik denetim matrisidir. Φ_H ise daha önce belirtildiği üzere LRPC kod çözme algoritmasıdır.

5. Girdileri $\mathbb{F}_{q^m}$ cisminden alınan $\frac{n}{3}$ uzunluğunda $\mathbf{g}_1$ ve $\mathbf{g}_2$ vektörleri üretilir.

6. $\frac{n}{3} \times \frac{n}{3}$ boyutunda G_1 ve G_2 çevrimsel matrisleri sırasıyla $\mathbf{g}_1$ ve $\mathbf{g}_2$ vektörlerinin (cyclic shift) döndürülmesiyle inşa edilir.

7. $G' = \begin{bmatrix} I_{\frac{n}{3}} & 0 & G_1 \\ 0 & I_{\frac{n}{3}} & G_2 \end{bmatrix}$ tanımlanır.

8. P, $n \times n$ birim matrisi olarak alınır.

9. $S = (H_1^T + G_1 H_3^T)^{-1}$ olarak alınır ve $\frac{n}{3} \times \frac{n}{3}$ boyutunda çevrimsel matris oluşturur.

10. $F = G'P^{-1}H^T S$ hesaplanır ve aşağıdaki formda $\frac{2n}{3} \times \frac{n}{3}$ boyutunda matris elde edilir:

$$F = \begin{bmatrix} I_{\frac{n}{3}} & 0 & G_1 \\ 0 & I_{\frac{n}{3}} & G_2 \end{bmatrix} \cdot \begin{bmatrix} H_1^T \\ H_2^T \\ H_3^T \end{bmatrix} \cdot S = \begin{bmatrix} H_1^T + G_1 H_3^T \\ H_2^T + G_2 H_3^T \end{bmatrix} S = \begin{bmatrix} I_{\frac{n}{3}} \\ F' \end{bmatrix},$$

öyle ki $F' = (H_2^T + G_2 H_3^T)(H_1^T + G_1 H_3^T)^{-1}$ dir.

F matrisinin sütunca ingirgenmiş eşelon form dönüştürülememe ihtimali vardır. Bu durumda sütunca ingirgenmiş eşelon form elde edilene kadar yeni vektörler üretilir.

5.6.2 Şifreleme algoritması

$\bar{\mathbf{m}}$ mesaj vektörü $\mathbb{F}_{q^m}$ cismi üzerindedir. $\mathbf{m}$ mesaj dizgisi, $\bar{\mathbf{m}}$ mesaj vektörü ve mesaj uzunluğu bilgisini içeren 4-bayt $\mathbf{a}$ dizgisinden oluşur. $\mathbf{m} = (\mathbf{a} \mid \bar{\mathbf{m}})$ dir. β ise iki bloktaki baytların sayısı olsun. s-bayt $\mathbf{m}$ gizli mesajını göndermek için $s \leq \beta$ olduğunda:

1. $s < \beta$ ise: β-bayt dizgi $\mathbf{x} = (\mathbf{m} \mid \mathbf{v})$, $\mathbf{v}$ düzgün rastgele $(\beta - s)$-bayt dizgi olacak şekilde tanımlanır.

2. $s = \beta$ ise: $\mathbf{x} = \mathbf{m}$ olarak tanımlanır.

3. $\mathbb{F}_{q^m}$ üzerinde n uzunluğunda rastgele bir $\mathbf{e}$ hata vektörü üretilir. (rank ağırlığı en fazla r olan ve uygun bir kod çözme algoritmasıyla çözülebilen)

4. $\mathbf{c}_1 = \mathbf{x}G' + \mathbf{e}$ hesaplanır. $\mathbf{c}_1$ $\frac{nm}{8}$ bayt olur.

5. $\mathbf{c}_2 = \mathbf{x}F$ hesaplanır. $\mathbf{c}_2$ $\frac{nm}{24}$ bayt olur.

6. $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ ise $\frac{nm}{6}$ bayt olur.

5.6.3 Şifre çözme algoritması

$\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ şifreli metni çözmek için:

1. $\mathbf{s}' = \mathbf{c}_1 P^{-1} H^T - \mathbf{c}_2 S^{-1} = (\mathbf{m}G' + \mathbf{e})P^{-1}H^T - (\mathbf{m}G'P^{-1}H^T S)S^{-1} = \mathbf{e}P^{-1}H^T$ hesaplanır.

2. $\mathbf{s}'$ vektörüne Φ_H kod çözme algoritması uygulanarak $\hat{\mathbf{e}} = \mathbf{e}P^{-1}$ elde edilir.

3. $\hat{\mathbf{e}}$, P ile çarpılarak $\mathbf{e}$ hata vektörü elde edilir.

4. $\mathbf{x}G' = \mathbf{c}_1 - \mathbf{e}$ sistemi çözülerek $\mathbf{x}$ vektörü hesaplanır. G' standart formda iken $\mathbf{c}_1 - \mathbf{e}$ nin ilk $\frac{\ell m}{8}$ baytını almak $\mathbf{x}$ vektörünü elde etmek için yeterlidir.

5. $\mathbf{x}$ vektörünün ilk 4 baytından mesaj uzunluğu bilgisine ulaşılır.

6. $\bar{\mathbf{m}}$ mesajına ulaşılır.

5.7 Örnek.

5.7.1 Parametreler

Bölüm 5.6.1 de verilen şartlara uygun parametreler seçilir:

1. $\mathbb{F}_{q^m} = \mathbb{F}_{2^4}$, $d = 2$, kod uzunluğu $n = 6$, $blk = \frac{n}{3} = 2$, kod boyutu $k = 4$ ve $\ell = 4$ olsun.

2. $m = 4$ dereceli $f(x) = x^4 + x + 1 \in \mathbb{F}_2[x]$ indirgenemez polinomu alınır.

5.7.2 Anahtar üretimi

Bölüm 5.6.1 de verilen adımlara uygun şekilde anahtarlar oluşturulur:

1. $\frac{n}{3} = 2$ uzunluğunda,

$$\mathbf{h}_1 = (1, \alpha^2), \mathbf{h}_2 = (0, \alpha^2) \text{ ve } \mathbf{h}_3 = (1, 0)$$

vektörleri üretilir.

2. $\frac{n}{3} \times \frac{n}{3} = 2 \times 2$ boyutlarında,

$$H_1 = \begin{bmatrix} 1 & \alpha^2 \\ \alpha^2 & 1 \end{bmatrix}, H_2 = \begin{bmatrix} 0 & \alpha^2 \\ \alpha^2 & 0 \end{bmatrix} \text{ ve } H_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

matrisleri inşa edilir.

3. $\frac{n}{3} \times n = 2 \times 6$ boyutunda,

$$H = \begin{bmatrix} 1 & \alpha^2 & 0 & \alpha^2 & 1 & 0 \\ \alpha^2 & 1 & \alpha^2 & 0 & 0 & 1 \end{bmatrix}$$

matrisi d ağırlığında bir LRPC kod için bir eşlik denetim matrisidir.

4. $\frac{n}{3} = 2$ uzunluğunda,

$$\mathbf{g}_1 = (\alpha^3, \alpha^5) \text{ ve } \mathbf{g}_2 = (\alpha^2, \alpha^7)$$

vektörleri üretilir.

5. $\frac{n}{3} \times \frac{n}{3} = 2 \times 2$ boyutlarında,

$$G_1 = \begin{bmatrix} \alpha^3 & \alpha^5 \\ \alpha^5 & \alpha^3 \end{bmatrix} \text{ ve } G_2 = \begin{bmatrix} \alpha^2 & \alpha^7 \\ \alpha^7 & \alpha^2 \end{bmatrix}$$

matrisleri inşa edilir.

6. G' matrisi tanımlanır:

$$G' = \begin{bmatrix} 1 & 0 & 0 & 0 & \alpha^3 & \alpha^5 \\ 0 & 1 & 0 & 0 & \alpha^5 & \alpha^3 \\ 0 & 0 & 1 & 0 & \alpha^2 & \alpha^7 \\ 0 & 0 & 0 & 1 & \alpha^7 & \alpha^2 \end{bmatrix}$$

7. $P = I_{6\times 6}$ olarak alınır.

8. $\frac{n}{3} \times \frac{n}{3} = 2 \times 2$ boyutlarında,

$$S = (H_1^T + G_1 H_3^T)^{-1} = \begin{bmatrix} 1 & \alpha^2 \\ \alpha^2 & 1 \end{bmatrix}$$

çevrimsel matrisi oluşturulur.

9. $\frac{2n}{3} \times \frac{n}{3} = 4 \times 2$ boyutlarında,

$$F = G' P^{-1} H^T S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \alpha^{13} & \alpha^6 \\ \alpha^6 & \alpha^{13} \end{bmatrix}$$

matrisi hesaplanır.

5.7.3 Şifreleme algoritması

Bölüm 5.6.2 de verilen adımlara uygun şekilde şifreleme yapılır:

1. $\mathbf{x} = \bar{\mathbf{m}} = (\alpha^2, 0, \alpha^3, \alpha^5)$ alınır.

2. $\mathbb{F}_{q^m}$ cismi üzerinde, $n = 6$ uzunluğunda, ve rank ağırlığı en fazla r olan

$$\mathbf{e} = (0, 0, \alpha, 0, \alpha, 0)$$

hata vektörü rastgele üretilir.

3. $\mathbf{c}_1$ hesaplanır:

$$\begin{aligned}
\mathbf{c}_1 &= \mathbf{x}G' + \mathbf{e} \\
&= \begin{bmatrix} \alpha^2 & 0 & \alpha^3 & \alpha^5 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & \alpha^3 & \alpha^5 \\ 0 & 1 & 0 & 0 & \alpha^5 & \alpha^3 \\ 1 & 0 & 1 & 0 & \alpha^2 & \alpha^7 \\ 1 & 0 & 0 & 1 & \alpha^7 & \alpha^2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \alpha & 0 & \alpha & 0 \end{bmatrix} \\
&= \begin{bmatrix} \alpha^2 & 0 & \alpha^3 & \alpha^5 & \alpha^{12} & \alpha^{10} \end{bmatrix} + \begin{bmatrix} 0 & 0 & \alpha & 0 & \alpha & 0 \end{bmatrix} \\
&= \begin{bmatrix} \alpha^2 & 0 & \alpha^9 & \alpha^5 & \alpha^{13} & \alpha^{10} \end{bmatrix}
\end{aligned}$$

4. $\mathbf{c}_2$ hesaplanır:

$$\begin{aligned}
\mathbf{c}_2 &= \mathbf{x}F \\
&= \begin{bmatrix} \alpha^2 & 0 & \alpha^3 & \alpha^5 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \alpha^{13} & \alpha^6 \\ \alpha^6 & \alpha^{13} \end{bmatrix} \\
&= \begin{bmatrix} \alpha^3 & \alpha \end{bmatrix}
\end{aligned}$$

5. $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ şifreli metni oluşturulur:

$$\mathbf{c} = (\alpha^2, 0, \alpha^9, \alpha^5, \alpha^{13}, \alpha^{10}, \alpha^3, \alpha)$$

5.7.4 Şifre çözme algoritması

Bölüm 5.6.3 de verilen adımlara uygun şekilde $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ şifreli metni çözülerek $\bar{\mathbf{m}}$ açık mesajına ulaşılır:

1. $\mathbf{s}'$ vektörü aşağıdaki gibi hesaplanır:

$$
\begin{aligned}
\mathbf{s}' &= \mathbf{c}_1 P^{-1} H^T - \mathbf{c}_2 S^{-1} \\
&= \mathbf{e} P^{-1} H^T \\
&= \begin{bmatrix} \alpha^2 & 0 & \alpha^9 & \alpha^5 & \alpha^{13} & \alpha^{10} \end{bmatrix} \cdot \begin{bmatrix} 1 & \alpha^2 \\ \alpha^2 & 1 \\ 0 & \alpha^2 \\ \alpha^2 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \alpha^3 & \alpha \end{bmatrix} \cdot \begin{bmatrix} \alpha^{14} & \alpha \\ \alpha & \alpha^{14} \end{bmatrix} \\
&= \begin{bmatrix} \alpha & \alpha^9 \end{bmatrix} - \begin{bmatrix} 0 & \alpha \end{bmatrix} \\
&= \begin{bmatrix} \alpha & \alpha^3 \end{bmatrix}
\end{aligned}
$$

2. $\mathbf{s}'$ vektörüne Φ_H kod çözme algoritması uygulanarak,

$$\hat{\mathbf{e}} = \mathbf{e} P^{-1} = \begin{bmatrix} 0 & 0 & \alpha & 0 & \alpha & 0 \end{bmatrix}$$

elde edilir.

(a) $\mathbf{s}' = (\alpha, \alpha^3)$ sendrom vektörü kullanılarak $S = \langle 0, \alpha, \alpha^3, \alpha^9 \rangle$ sendrom uzayı oluşturulur.

(b) $S_i = F_i^{-1} S$ eşitliği kullanılarak,

$$S_1 = 1^{-1}(0, \alpha, \alpha^3, \alpha^9) = (0, \alpha, \alpha^3, \alpha^9)$$

ve

$$S_2 = (\alpha^2)^{-1}(0, \alpha, \alpha^3, \alpha^9) = (0, \alpha^{14}, \alpha, \alpha^7)$$

hesaplanır. Buradan hata (support)u $E = S_1 \cap S_2 = \alpha$ bulunur. Dolayısıyla E nin bazı $\{E_1\} = \{\alpha\}$ dır.

(c) $\hat{e}_i = \sum_{i=1}^{n} \hat{e}_{ij} E_j$ kullanılarak $\hat{\mathbf{e}} = (\hat{e}_{11}\alpha, \hat{e}_{21}\alpha, \hat{e}_{31}\alpha, \hat{e}_{41}\alpha, \hat{e}_{51}\alpha, \hat{e}_{61}\alpha)$ bulunur. $H\hat{\mathbf{e}}^T = \mathbf{s}'$ sisteminde yerine yazılırsa,

$$\begin{bmatrix} 1 & \alpha^2 & 0 & \alpha^2 & 1 & 0 \\ \alpha^2 & 1 & \alpha^2 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \hat{e}_{11}\alpha \\ \hat{e}_{21}\alpha \\ \hat{e}_{31}\alpha \\ \hat{e}_{41}\alpha \\ \hat{e}_{51}\alpha \\ \hat{e}_{61}\alpha \end{bmatrix} = \begin{bmatrix} \alpha \\ \alpha^3 \end{bmatrix} \text{ elde edilir.}$$

$\hat{e}_{11}\alpha + \hat{e}_{21}\alpha^3 + \hat{e}_{41}\alpha^3 + \hat{e}_{51}\alpha = \alpha$

$\hat{e}_{11}\alpha^3 + \hat{e}_{21}\alpha + \hat{e}_{31}\alpha^3 + \hat{e}_{61}\alpha = \alpha^3$

$\hat{e}_{11} + \hat{e}_{51} = 1$

$\hat{e}_{21} + \hat{e}_{41} = 0$

$\hat{e}_{21} + \hat{e}_{61} = 0$

$\hat{e}_{11} + \hat{e}_{31} = 1$ denklemleri bulunur.

Buradan, $\hat{e}_{11} = \hat{e}_{21} = \hat{e}_{41} = \hat{e}_{61} = 0$ ve $\hat{e}_{31} = \hat{e}_{51} = 1$ elde edilerek

$\hat{\mathbf{e}} = (0,0,\alpha,0,\alpha,0)$ hata vektörüne ulaşılır.

3. **e** hata vektörü elde edilir:

$$\begin{aligned} \mathbf{e} &= \hat{\mathbf{e}}P \\ &= \begin{bmatrix} 0 & 0 & \alpha & 0 & \alpha & 0 \end{bmatrix} \end{aligned}$$

4. $\mathbf{x}G'$ hesaplanır:

$$\begin{aligned} \mathbf{x}G' &= \mathbf{c}_1 - \mathbf{e} \\ &= \begin{bmatrix} \alpha^2 & 0 & \alpha^9 & \alpha^5 & \alpha^{13} & \alpha^{10} \end{bmatrix} - \begin{bmatrix} 0 & 0 & \alpha & 0 & \alpha & 0 \end{bmatrix} \\ &= \begin{bmatrix} \alpha^2 & 0 & \alpha^3 & \alpha^5 & \alpha^{12} & \alpha^{10} \end{bmatrix} \end{aligned}$$

5. $\mathbf{x}G' = \begin{bmatrix} \alpha^2 & 0 & \alpha^3 & \alpha^5 & \alpha^{12} & \alpha^{10} \end{bmatrix}$ sistemi çözülerek $\mathbf{x} = (\alpha^2, 0, \alpha^3, \alpha^5)$ elde edilir ve $\bar{\mathbf{m}}$ mesajına ulaşılır.

6. NTS-KEM

Bu bölümde NTS-KEM kriptosistemiyle ilgili özet bilgilere yer verilecektir. Detaylı bilgi için [2] nolu kaynak incelenebilir.

NTS-KEM kriptosistemi, McEliece ve Niederreiter açık anahtarlı şifreleme algoritmalarının bir başka şeklidir. Fakat NTS-KEM şifreli bir mesajın iletilmesiyle değil rastgele bir anahtarın güvenli iletilmesiyle ilgilenir.

McEliece ve Niederreiter kriptosistemleri orijinal halleriyle seçili açık metin ve seçili şifreli metin atakları karşısında ayırt edilemez değildir. NTS-KEM ise bir anahtar kapsülleme mekanizması olarak seçili şifreli metin güvenliğine sahiptir.

NTS-KEM sisteminin seçili şifreli metin güvenliği McElice sisteminin tek yönlülüğünü kırma zorluğuna dayanır. Bu ise rastgele lineer kodların bilinen kod çözme problemiyle ilgilidir. NIST tarafından yapılan çağrıda tanımlanan üç ayrı güvenlik kategorisi için NTS-KEM sisteminde üç parametre seti önerilmiştir. Bu sayede NTS-KEM kuantum öncesi ve sonrası güvenlik gerektiren uygulamaların her ikisinde de kullanılabilir.

NTS-KEM etkin anahtar kapsülleme ve kapsülden çıkarma işlemlerine yer verir. Anahtarı kapsülden çıkarma işlemleri Goppa kodların etkin kod çözme algoritmaları sayesinde etkin bir şekilde gerçekleşir. Şifreli metin boyutu daha kısa olan bu sistemin kullanımı iletişim bandı genişliği sınırlı olan uygulamalara elverişlidir. Fakat çoğu kod tabanlı sistem gibi NTS-KEM de büyük boyutta açık anahtarlara sahiptir. Bunun sebebi ise devirli ya da yarı devirli yapılar yerine daha klasik kodların seçilmesidir. Bununla birlikte geniş boyutta açık anahtarların bir dezavantaj olarak görülmediği, hızlı işlemlere ve küçük boyutlarda şifreli metinlere daha çok önem verildiği uygulama örnekleri mevcuttur. Uzun vadeli sabit açık anahtarlar kullanan herhangi bir uygulama bu duruma örnektir.

NTS-KEM sisteminin tasarımında kapsamlı olarak çalışılmış ve yaklaşık 40 yıldır

güvenilir olan hata düzeltme kodları tercih edilmiştir. Bu sayede basit, esnek ve etkili bir yaklaşımla uzun vadede kuantum sonrası güvenliği sağlamak hedeflenmiştir. Bu güvenlik ise iyi bilinen bir matematik problemine ve klasik kod çözme algoritmalarının karmaşıklığının oldukça yakın tahminlerine dayanmaktadır. Seçili şifreli metin ataklarına karşı uygun güvenlik seviyelerini sağlamak için bu karmaşıklık ayarlanabilir.

Yapısı bilindiğinde ikili Goppa kodlar Patterson metodu [15] veya Berlekamp-Massey algoritması [3, 11] ile etkili bir şekilde çözülebilir. Fakat kod yapısı gizlendiğinde, ikili Goppa kodları çözmenin rastgele lineer kodları çözmek kadar zor hale gelmesi beklenir. Bu durumda en iyi bilinen algoritmalar orijinali Prange [16] tarafından önerilmiş ISD tekniğine dayanmaktadır.

Sistemin Avantaj ve Dezavantajları

- NTS-KEM güçlü güvenlik dayanaklarına sahiptir. Kriptografi toplumu tarafından 40 yıldır büyük ilgi gören McEliece ve Niederreiter sistemlerinin bir versiyonudur. NTS-KEM sisteminin kendi yapısı ise seçili şifreli metin ataklarına karşı dayanıklı bir anahtar kapsülleme mekanizmasıdır. Bu güvenlik ise McEliece açık anahtarlı şifreleme sisteminin ters çevirme problemi ile olan güçlü bağlantısına dayanmaktadır.

- Bu sebeple NTS-KEM basit ve iyi anlaşılmış bir matematik problemine dayanmaktadır. Bu problemi çözmeye karşı en temel yaklaşım olan ISD algoritmaları ise kapsamlı bir şekilde çalışılmış ve karmaşıklık tahminleri yapılmıştır.

- Bu tahminler amaçlanan güvenlik kategorilerine göre uygun parametrelerin ayarlanmasında kullanılmıştır.

- NTS-KEM için ihtiyatlı parametreler verilmesine rağmen sistem, parametrelerin esnetilmesine uygundur. Kod uzunluğu ve hata vektörünün ağırlığı gibi parametrelerde, (ISD) algoritmalarının ileride gelişmesi ya da şu anki tahminlerin çok iyimser olduğunun kanıtlanması durumunda değişiklikler yapılabilir. Yani parametre seçimleri değiştirilerek güvenlik seviyesi tahminlerinde, anahtarlar ve şifreli metinlerin boyutlarında kolaylıkla

oynamalar yapılabilir. Bu sayede performans ve güvenlik arasındaki olası ödünleşmede daha kolay ayarlamalar yapılabilir. Ayrıca parametreler kasıtlı olarak düşük ayarlanıp yeni önerilmiş kriptanalitik teknikler pratikte test edilebilir.

- NTS-KEM kriptosisteminin bir diğer avantajı ise uzun vadeli anahtarlar sağlamasıdır. Sistemde kapsülden çıkarma işlemi sırasında kararlı bir kod çözme algoritması kullanılır. Dolayısıyla açık anahtardan gizli anahtarı bulmanın pratikte bilinen tek yolu kaba kuvvettir. Bu sebeple açık ve gizli anahtar ikilileri uzun süre boyunca kullanılabilir.

- NTS-KEM kısa şifreli metinlere sahiptir. En yüksek güvenlik seviyesinde şifreli metin boyutu 2000 bit civarındadır. Bu da sistemi özellikle bant genişliği düşük, uzun vadeli anahtarlar kullanan uygulamalar için uygun hale getirir.

- Ayrıca NTS-KEM yazılım uygulamalarını makul şekilde hızlandıran başta kapsülleme olmak üzere etkili ve basit işlemlere sahiptir.

- NTS-KEM kriptosisteminin göze çarpan dezavantajı ise açık anahtar boyutudur. Önerilen en yüksek güvenlik seviyesinde, NTS-KEM açık anahtarı yaklaşık olarak 1.39 MB boyutundadır. 128-bit güvenlik seviyesinde ise 312 KB tır. Geniş anahtar boyutu tüm Goppa kod tabanlı sistemlerin ortak özelliği iken geniş anahtar boyutunu kaygılanacak bir özellik olmaktan çıkaran uygulamalar mevcuttur. NTS-KEM ise bu uygulamalar arasında uzun vadeli güvenliği, küçük boyutta şifreli metinleri ve etkili açık ve gizli anahtar işlemleri gibi özellikleri ile yerini alır.

6.1 Parametreler

1. $n = 2^m$: kod uzunluğu
2. τ: kodun düzeltebileceği hata sayısı
3. $f(x)$: $\mathbb{F}_2$ üzerinde m dereceli indirgenemez bir polinom
4. $\ell = 256$: kapsüllenecek rastgele anahtar uzunluğu

Ayrıca $k = n - \tau m$ olacak şekilde $\log_2 \binom{n}{\tau} \geq \ell$ ve $\ell < k < n$ olmalıdır.
NTS-KEM sisteminde ℓ bit uzunluğunda ikili dizgi üretmek için $H_\ell(.)$ sözde rastgele bit üreteci kullanılır. NTS-KEM kriptosisteminde $\ell = 256$ olarak alınmıştır ve $H_\ell(.)$ olarak SHA3-256 özet fonksiyonu kullanılmıştır.

6.2 Anahtar Üretimi

1. $g_i \in \mathbb{F}_{2^m} \cong \mathbb{F}_2[x]/f(x)$ ve $g_0 \neq 0$ olmak üzere τ dereceli monik bir $G(z) = g_0 + g_1 z + \cdots + g_{\tau-1} z^{\tau-1} + z^\tau$ Goppa polinomu rastgele üretilir. $G(z)$ polinomu $n = 2^m$ uzunluğunda, $k = n - \tau m$ boyutunda ve τ ya kadar hata düzeltme kapasitesine sahip ikili $\overline{C_\mathcal{G}}$ Goppa kodunu tanımlar. $G(z)$ Goppa polinomu aşağıdaki şekilde üretilir:

   (a) $i = \{0, 1, \ldots, \tau - 1\}$ için g_i katsayıları, $g_0 \neq 0$ olmak üzere $\mathbb{F}_{2^m}$ cisminden rastgele seçilir.

   (b) $g_\tau = 1$ alınır ve $G(z) = \sum_{i=0}^{\tau} g_i z^i$ oluşturulur.

   (c) $G(z)$ polinomunun uygunluğu kontrol edilir. Aşağıdaki şartlar sağlanıyorsa $G(z)$ polinomu uygundur:

   - $g_0 \neq 0$
   - $G(z)$ polinomunun $\mathbb{F}_{2^m}$ cisminde kökü yoktur. Bu durum $n = 2^m$ olduğunu garanti eder.
   - $G(z)$ polinomunun herhangi bir cisim genişlemesinde tekrar eden bir kökü yoktur. Bu durum ise ikili Goppa kodun τ ya kadar hata düzeltebilir olduğunu garanti eder. Bu durumun sağlanması için $\left(G(z), \frac{d}{dz} G(z)\right) = 1$ olmalıdır (bkz. Ek1).

   Yukarıdaki şartlardan biri sağlanmadığında (a) maddesine geri dönülür.

2. n uzunluğunda rastgele bir $\mathbf{p}$ permütasyon vektörü üretilir. n elemanlı kümede $\pi_\mathbf{p}$ permütasyonunu temsil eder.

   - n elemanlı sıralı bir dizi üzerindeki π permütasyonu, her satır ve sütununda yalnız bir tane 1 bulunan ve diğer tüm elemanları 0 olan $P \in \mathbb{F}_2^{n \times n}$ permütasyon matrisi tarafından temsil edilebilir. π

permütasyonu aynı zamanda $\mathbf{p} = (p_0, p_1, \ldots, p_{n-1})$ permütasyon vektörüyle de temsil edilebilir. Burada p_i ile i. sütununda 1 bulunan satır ifade edilir.

- $\mathbf{b} = (b_0, b_1, \ldots, b_{n-1})$ dizisi verildiğinde, sıralı dizi $\mathbf{b}' = \mathbf{b}P = \pi_{\mathbf{p}}(\mathbf{b})$, $b'_i = b_{p_i}$ ve ters sıralı dizi $\mathbf{b} = \mathbf{b}'P^{-1} = \pi_{\mathbf{p}}^{-1}(\mathbf{b}')$, $b_{p_i} = b'_i$ şeklinde hesaplanır.

3. Değiştirilmiş $C_{\mathcal{G}}$ kodunun satırca indirgenmiş eşelon biçiminde $G = [I_k \mid Q]$ üreteç matrisi aşağıdaki şekilde oluşturulur:

   (a) $\mathbf{a}'$, $\mathbb{F}_{2^m}$ cisminin tüm elemanlarının sıralı bir dizisi olmak üzere $\mathbf{a} = \pi_{\mathbf{p}}(\mathbf{a}') = (a_{p_0}, a_{p_1}, \ldots, a_{p_{n-1}}) \in \mathbb{F}_{2^m}^n$ ise $\mathbf{a}'$ dizisinin elemanlarının $\pi_{\mathbf{p}}$ permütasyonuna göre tekrar düzenlenmesiyle elde edilen dizidir.

   - $\alpha \in \mathbb{F}_{2^m}$, $\mathbb{F}_{2^m} \cong \mathbb{F}_2[x]/f(x)$ olmak üzere $f(x)$ polinomunun bir kökü olsun. $B = \left\langle \alpha^{(m-1)}, \ldots, \alpha, 1 \right\rangle$ ise $\mathbb{F}_{2^m}$ cisminin bir bazı olsun. $\mathbb{F}_{2^m}$ cisminin i. elemanı B bazında aşağıdaki şekilde tanımlanır.

     $$B[i] = \{b_0\alpha^{(m-1)} + b_1\alpha^{(m-2)} + \cdots + b_{m-2}\alpha + b_{m-1} : b_j \in \{0,1\}\},$$

     öyle ki $i = \sum_{j=0}^{m-1} b_j 2^j$.
   - $\mathbf{a}'$ aşağıdaki şekilde tanımlanır:

     $$\begin{aligned} \mathbf{a}' &= (a_0, a_1, a_2, \ldots, a_{n-2}, a_{n-1}) \\ &= (B[0], B[1], B[2], \ldots, B[n-2], B[n-1]) \end{aligned}$$

   (b) $\mathbf{a}$ dizisi kullanılarak $H_m \in \mathbb{F}_{2^m}^{\tau \times n}$ eşlik denetim matrisi inşa edilir.

   $\mathbf{h} = (h_{p_0}, h_{p_1}, \ldots, h_{p_{n-1}}) \in \mathbb{F}_{2^m}^n$ ise H_m matrisinin ilk satırıdır.

   - $\bar{\mathbf{h}} = (G(B[0]), G(B[1]), \ldots, G(B[n-1]))$ vektörü elde edilir.
   - $\mathbf{h}' = (\bar{h}_0^{-2}, \bar{h}_1^{-2}, \ldots, \bar{h}_{n-1}^{-2})$ vektörü oluşturulur.
   - $\mathbf{h} = \pi_p(\mathbf{h}') = (h_{p_0}, h_{p_1}, \ldots, h_{p_{n-1}})$ hesaplanır.
   - H_m eşlik denetim matrisi hesaplanır:

     $$H_m = \begin{bmatrix} h_{p_0} & h_{p_1} & \ldots & h_{p_{n-1}} \\ a_{p_0}h_{p_0} & a_{p_1}h_{p_1} & \ldots & a_{p_{n-1}}h_{p_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p_0}^{\tau-1}h_{p_0} & a_{p_1}^{\tau-1}h_{p_1} & \ldots & a_{p_{n-1}}^{\tau-1}h_{p_{n-1}} \end{bmatrix}$$

   (c) $B(.)^T$ operatörü kullanılarak $\mathbf{H}_m$ matrisi $\mathbf{H} \in \mathbb{F}_2^{m\tau \times n}$ matrisine dönüştürülür.

   - $B(a_i) = (b_{i0}, b_{i1}, \ldots, b_{i(m-1)})$, a_i nin $\mathbb{F}_2$ üzerinde,

$$a_i = b_{i0} + b_{i1}\alpha + b_{i2}\alpha^2 + \cdots + b_{i(m-1)}\alpha^{m-1}$$

ve $b_{ij} \in \mathbb{F}_2$ olacak şekilde bir gösterimi olsun.

(d) H matrisi satırca indirgenmiş eşelon biçimine dönüştürülür ve eğer gerekliyse sütunlarının yerleri değiştirilerek son $n-k$ sütununun I_{n-k} birim matrisi oluşturması sağlanır.

(e) Sütunların yer değiştirmesini temsil eden permütasyon ρ ise aynı düzenleme $\mathbf{a}$, $\mathbf{h}$ ve $\mathbf{p}$ vektörlerine de uygulanır. $\mathbf{a} = \rho(\mathbf{a})$, $\mathbf{h} = \rho(\mathbf{h})$ ve $\mathbf{p} = \rho(\mathbf{p})$ halini alır.

(f) Değiştirilmiş $C_{\mathcal{G}}$ kodunun $G = [I_k \mid Q] \in \mathbb{F}_2^{k\times n}$ üreteç matrisi, $H = [Q^T \mid I_{n-k}]$ eşlik denetim matrisinden elde edilir.

4. $\mathbf{a}$ ve $\mathbf{h}$ vektörleri, $\mathbf{a}_a$, $\mathbf{h}_a \in \mathbb{F}_{2^m}^{k-\ell}$, $\mathbf{a}_b$, $\mathbf{h}_b \in \mathbb{F}_{2^m}^{\ell}$ ve $\mathbf{a}_c$, $\mathbf{h}_c \in \mathbb{F}_{2^m}^{n-k}$ olmak üzere $\mathbf{a} = (\mathbf{a}_a \mid \mathbf{a}_b \mid \mathbf{a}_c)$ ve $\mathbf{h} = (\mathbf{h}_a \mid \mathbf{h}_b \mid \mathbf{h}_c)$ şeklinde parçalara ayrılırlar. Son olarak;

$$\mathbf{a}^* = (\mathbf{a}_b \mid \mathbf{a}_c) \text{ ve } \mathbf{h}^* = (\mathbf{h}_b \mid \mathbf{h}_c)$$

tanımlanır.

NTS-KEM açık ve gizli anahtarları aşağıdaki gibi belirlenir.

- $Q \in \mathbb{F}_2^{k\times(n-k)}$ ve τ, ℓ pozitif tam sayılar olmak üzere, açık anahtar pk$= (Q, \tau, \ell)$.
- $\mathbf{a}^*, \mathbf{h}^* \in \mathbb{F}_{2^m}^{n-k+\ell}$ ve $\mathbf{p} \in \mathbb{F}_{2^m}^{n}$ olmak üzere gizli anahtar sk=$(\mathbf{a}^*, \mathbf{h}^*, \mathbf{p})$.

6.3 Kapsülleme Algoritması

pk $= (Q, \tau, \ell)$ açık anahtarı verildiğinde kapsülleme sonucunda $\mathbb{F}_2$ üzerinde iki vektör oluşturulur. Biri ℓ uzunluğunda rastgele bir $\mathbf{k}_r$ vektörü, diğeri ise $\mathbf{k}_r$ yi kapsülleyen $\mathbf{c}^*$ şifreli metnidir.
NTS-KEM kapsülleme algoritması ise aşağıdaki şekildedir:

1. τ ağırlıklı $\mathbf{e} \in \mathbb{F}_2^n$ hata vektörü rastgele üretilir.
2. $\mathbf{e}$ vektörü, $\mathbf{e}_a \in \mathbb{F}_2^{k-\ell}$, $\mathbf{e}_b \in \mathbb{F}_2^{\ell}$ ve $\mathbf{e}_c \in \mathbb{F}_2^{n-k}$ olmak üzere $\mathbf{e} = (\mathbf{e}_a \mid \mathbf{e}_b \mid \mathbf{e}_c)$ olarak parçalara ayrılır.

3. $\mathbf{k}_e = H_\ell(\mathbf{e}) \in \mathbb{F}_2^\ell$ hesaplanır.

4. $\mathbf{m} = (\mathbf{e}_a \mid \mathbf{k}_e) \in \mathbb{F}_2^k$ mesaj vektörü inşa edilir.

5. $\mathbf{m}$ mesajı Q ile birlikte kodlanır:

$$\begin{aligned}\mathbf{c} &= (\mathbf{m} \mid \mathbf{m}Q) + \mathbf{e} \\ &= (\mathbf{e}_a \mid \mathbf{k}_e \mid (\mathbf{e}_a \mid \mathbf{k}_e)Q) + (\mathbf{e}_a \mid \mathbf{e}_b \mid \mathbf{e}_c) \\ &= (\mathbf{0}_a \mid \mathbf{k}_e + \mathbf{e}_b \mid (\mathbf{e}_a \mid \mathbf{k}_e)Q + \mathbf{e}_c) \\ &= (\mathbf{0}_a \mid \mathbf{c}_b \mid \mathbf{c}_c).\end{aligned}$$

   Burada $\mathbf{c}_b = \mathbf{k}_e + \mathbf{e}_b$ ve $\mathbf{c}_c = (\mathbf{e}_a \mid \mathbf{k}_e)Q + \mathbf{e}_c$ dir. $\mathbf{c}$ vektöründen ilk $k-\ell$ koordinatı uzaklaştırılırsa, $\mathbf{c}^* = (\mathbf{c}_b \mid \mathbf{c}_c) \in \mathbb{F}_2^{n-k+\ell}$ elde edilir.

6. $\mathbf{k}_r = H_\ell(\mathbf{k}_e \mid \mathbf{e}) \in \mathbb{F}_2^\ell$ olmak üzere $(\mathbf{k}_r, \mathbf{c}^*)$ çıktısı alınır.

6.4 Kapsülden Çıkarma Algoritması

$\mathbf{c}^* = (\mathbf{c}_b \mid \mathbf{c}_c)$ şifreli metni $\text{sk} = (\mathbf{a}^*, \mathbf{h}^*, \mathbf{p})$ gizli anahtarıyla birlikte aşağıdaki şekilde kapsülden çıkarılır:

1. Değiştirilmiş $\mathbf{e}'$ hata vektörünü elde etmek üzere, $\mathbf{c} = (\mathbf{0}_a \mid \mathbf{c}_b \mid \mathbf{c}_c) \in \mathbb{F}_2^n$ vektörüne gizli anahtar kullanılarak kod çözme algoritması uygulanır.

   (a) $\mathbf{a}^*$ ve $\mathbf{h}^*$ vektörleri $r = m\tau = n-k$ olmak üzere aşağıdaki şekilde parçalanır:
   $\mathbf{a}^* = (\mathbf{a}_b \mid \mathbf{a}_c) = (a_{b,0}, a_{b,1}, \ldots, a_{b,\ell-1} \mid a_{c,0}, a_{c,1}, \ldots, a_{c,r-1})$
   $\mathbf{h}^* = (\mathbf{h}_b \mid \mathbf{h}_c) = (h_{b,0}, h_{b,1}, \ldots, h_{b,\ell-1} \mid h_{c,0}, h_{c,1}, \ldots, h_{c,r-1})$

   (b) $H_m^* \in \mathbb{F}_{2^m}^{2\tau \times (\ell + m\cdot\tau)}$ eşlik denetim matrisi aşağıdaki şekilde inşa edilir:

$$H_m^* = \left[\begin{array}{ccc|ccc} h_{b,0} & \cdots & h_{b,\ell-1} & h_{c,0} & \cdots & h_{c,r-1} \\ a_{b,0}h_{b,0} & \cdots & a_{b,\ell-1}h_{b,\ell-1} & a_{c,0}h_{c,0} & \cdots & a_{c,r-1}h_{c,r-1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{b,0}^{\tau-1}h_{b,0} & \cdots & a_{b,\ell-1}^{\tau-1}h_{b,\ell-1} & a_{c,0}^{\tau-1}h_{c,0} & \cdots & a_{c,r-1}^{\tau-1}h_{c,r-1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{b,0}^{2\tau-1}h_{b,0} & \cdots & a_{b,\ell-1}^{2\tau-1}h_{b,\ell-1} & a_{c,0}^{2\tau-1}h_{c,0} & \cdots & a_{c,r-1}^{2\tau-1}h_{c,r-1} \end{array}\right]$$

   (c) $\mathbf{c}^* = (\mathbf{c}_b \mid \mathbf{c}_c)$ şifreli metninin sendromları hesaplanır:

$$\begin{aligned}\mathbf{s} &= (\mathbf{c}_b \mid \mathbf{c}_c) \cdot (H_m^*)^T \\ &= (s_0, s_1, \ldots, s_{2\tau-1})\end{aligned}$$

(d) Berlekamp Massey algoritması (bkz. Ek1) ile σ^* fonksiyonu hesaplanır.

(e) $\mathbf{\Lambda} = \{\sigma^*(B[0]), \sigma^*(B[1]), \ldots, \sigma^*(B[n-1])\}$ hesaplanır.

(f) $\mathbf{e}'$ hata vektörü aşağıdaki şekilde elde edilir:

- $\mathbf{e}' = \mathbf{0} \in \mathbb{F}_2^n$ alınır.
- $\sigma^*(B[j] = 0$ için $e'_j = 1$ olarak değiştirilir.

2. $\mathbf{e} = \pi_{\mathbf{p}}(\mathbf{e}')$ hata vektörü hesaplanır.

3. $\mathbf{e}_a \in \mathbb{F}_2^{k-\ell}$, $\mathbf{e}_b \in \mathbb{F}_2^{\ell}$ ve $\mathbf{e}_c \in \mathbb{F}_2^{n-k}$ olacak şekilde $\mathbf{e} = (\mathbf{e}_a \mid \mathbf{e}_b \mid \mathbf{e}_c)$ hata vektörü parçalanır ve $\mathbf{k}_e = \mathbf{c}_b - \mathbf{e}_b$ hesaplanır.

4. $H_\ell(\mathbf{e}) = \mathbf{k}_e$ ve wt($\mathbf{e}$)= τ olup olmadığı kontrol edilir. Eğer öyleyse $\mathbf{k}_r = H_\ell(\mathbf{k}_e \mid \mathbf{e}) \in \mathbb{F}_2^{\ell}$ çıktısı alınır, aksi takdirde $\perp$ çıktısı alınır.

6.5 Örnek.

6.5.1 Parametreler

Bölüm 6.1 de verilen şartlara uygun parametreler seçilir:

1. $q = 2$ ve $m = 4$ olmak üzere $\mathbb{F}_{2^4}$ cismi
2. $n = 2^4 = 16$: kod uzunluğu $\tau = 2$, $k = 8$: kod boyutu, $\ell = 6$
3. $m = 4$ dereceli $f(x) = x^4 + x + 1 \in \mathbb{F}_2[x]$ indirgenemez polinomu
4. α: $f(x)$ polinomunun ilkel kökü

6.5.2 Anahtar üretimi

Bölüm 6.2 de verilen adımlara uygun şekilde anahtarlar oluşturulur:

1. $G(z) = \alpha^8 + \alpha z + z^2 \in \mathbb{F}_{2^4}[z]$ Goppa polinomu rastgele üretilir.

2. $n = 16$ uzunluğunda $\mathbf{p} = (3,5,0,7,2,11,9,6,13,10,8,14,1,4,12,15)$ permütasyon vektörü rastgele üretilir.

3. $\mathbf{a}' = (0,\alpha^3,\alpha^2,\alpha^6,\alpha,\alpha^9,\alpha^5,\alpha^{11},1,\alpha^{14},\alpha^8,\alpha^{13},\alpha^4,\alpha^7,\alpha^{10},\alpha^{12})$ olmak üzere
$\mathbf{a} = \pi_{\mathbf{p}}(\mathbf{a}') = (\alpha^6,\alpha^9,0,\alpha^{11},\alpha^2,\alpha^{13},\alpha^{14},\alpha^5,\alpha^7,\alpha^8,1,\alpha^{10},\alpha^3,\alpha,\alpha^4,\alpha^{12})$ oluşturulur.

4. $\bar{\mathbf{h}} = (\alpha^8,\alpha^9,\alpha^{11},1,\alpha^8,\alpha^9,\alpha^{11},1,\alpha^5,\alpha^{14},\alpha^{13},\alpha,\alpha^5,\alpha^{14},\alpha^{13},\alpha)$ vektörü hesaplanır.

5. $\mathbf{h}' = (\alpha^{14},\alpha^{12},\alpha^8,1,\alpha^{14},\alpha^{12},\alpha^8,1,\alpha^5,\alpha^2,\alpha^4,\alpha^{13},\alpha^5,\alpha^2,\alpha^4,\alpha^{13})$ vektörü hesaplanır.

6. $\mathbf{a} = \pi_{\mathbf{p}}(\mathbf{a}') = (\alpha^6,\alpha^9,0,\alpha^{11},\alpha^2,\alpha^{13},\alpha^{14},\alpha^5,\alpha^7,\alpha^8,1,\alpha^{10},\alpha^3,\alpha,\alpha^4,\alpha^{12})$
ve
$\mathbf{h} = \pi_{\mathbf{p}}(\mathbf{h}') = (1,\alpha^{12},\alpha^{14},1,\alpha^8,\alpha^{13},\alpha^2,\alpha^8,\alpha^2,\alpha^4,\alpha^5,\alpha^4,\alpha^{12},\alpha^{14},\alpha^5,\alpha^{13})$
elde edilir.

7. H_m eşlik denetim matrisi oluşturulur:

$$H_m = \begin{bmatrix} 1 & \alpha^{12} & \alpha^{14} & 1 & \alpha^8 & \alpha^{13} & \alpha^2 & \alpha^8 & \alpha^2 & \alpha^4 & \alpha^5 & \alpha^4 & \alpha^{12} & \alpha^{14} & \alpha^5 & \alpha^{13} \\ \alpha^6 & \alpha^6 & 0 & \alpha^{11} & \alpha^{10} & \alpha^{11} & \alpha & \alpha^{13} & \alpha^9 & \alpha^{12} & \alpha^5 & \alpha^{14} & 1 & 1 & \alpha^9 & \alpha^{10} \end{bmatrix}$$

8. $B(.)^T$ operatörü ile H_m matrisi;

$$H = \left[\begin{array}{cccccccc|cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{array}\right]$$

matrisine dönüştürülür.

9. H matrisi;

$$H = \left[\begin{array}{cccccccc|cccccccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

matrisine dönüştürülür ve bu işlem sırasında sütunların yerlerini değiştirmek için kullanılan ρ permütasyonuyla,

$$\mathbf{a} = \rho(\mathbf{a}) = (\alpha^7, \alpha^8, 1, \alpha^{10}, \alpha^3, \alpha, \alpha^4, \alpha^{12}, \alpha^6, \alpha^9, 0, \alpha^{11}, \alpha^2, \alpha^{13}, \alpha^{14}, \alpha^5),$$

$$\mathbf{h} = \rho(\mathbf{h}) = (\alpha^2, \alpha^4, \alpha^5, \alpha^4, \alpha^{12}, \alpha^{14}, \alpha^5, \alpha^{13}, 1, \alpha^{12}, \alpha^{14}, 1, \alpha^8, \alpha^{13}, \alpha^2, \alpha^8),$$

$$\mathbf{p} = \rho(\mathbf{p}) = (13, 10, 8, 14, 1, 4, 12, 15, 3, 5, 0, 7, 2, 11, 9, 6)$$

vektörleri elde edilir.

10. G üreteç matrisi aşağıdaki şekilde oluşturulur:

$$G = \left[\begin{array}{cccccccc|cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{array}\right]$$

11. **a** ve **h** vektörleri

$$\mathbf{a} = (\alpha^7, \alpha^8 \mid 1, \alpha^{10}, \alpha^3, \alpha, \alpha^4, \alpha^{12} \mid \alpha^6, \alpha^9, 0, \alpha^{11}, \alpha^2, \alpha^{13}, \alpha^{14}, \alpha^5),$$

$$\mathbf{h} = (\alpha^2, \alpha^4 \mid \alpha^5, \alpha^4, \alpha^{12}, \alpha^{14}, \alpha^5, \alpha^{13} \mid 1, \alpha^{12}, \alpha^{14}, 1, \alpha^8, \alpha^{13}, \alpha^2, \alpha^8)$$

şeklinde parçalandıktan sonra,

$$\mathbf{a}^* = (1, \alpha^{10}, \alpha^3, \alpha, \alpha^4, \alpha^{12} \mid \alpha^6, \alpha^9, 0, \alpha^{11}, \alpha^2, \alpha^{13}, \alpha^{14}, \alpha^5),$$

$$\mathbf{h}^* = (\alpha^5, \alpha^4, \alpha^{12}, \alpha^{14}, \alpha^5, \alpha^{13} \mid 1, \alpha^{12}, \alpha^{14}, 1, \alpha^8, \alpha^{13}, \alpha^2, \alpha^8)$$

vektörleri tanımlanır.

12. Açık anahtar $\mathrm{pk} = \left(\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}, 2, 6 \right)$ ve

gizli anahtar $\mathrm{sk} = (1, \alpha^{10}, \alpha^3, \alpha, \alpha^4, \alpha^{12} \mid \alpha^6, \alpha^9, 0, \alpha^{11}, \alpha^2, \alpha^{13}, \alpha^{14}, \alpha^5, \alpha^5,$
$\alpha^4, \alpha^{12}, \alpha^{14}, \alpha^5, \alpha^{13} \mid 1, \alpha^{12}, \alpha^{14}, 1, \alpha^8, \alpha^{13}, \alpha^2, \alpha^8, 1, 13,$
$10, 8, 14, 1, 4, 12, 15, 3, 5, 0, 7, 2, 11, 9, 6)$

olarak belirlenir.

6.5.3 Kapsülleme algoritması

Bölüm 6.3 de verilen adımlara uygun şekilde $(\mathbf{k}_r, \mathbf{c}^*)$ ikilisi oluşturulur:

1. $\mathbf{e} = (0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0)$ hata vektörü rastgele üretilir.
2. $\mathbf{e} = (\mathbf{e}_a, \mathbf{e}_b, \mathbf{e}_c) = (0,0 \mid 0,0,1,0,0,0 \mid 0,0,0,0,0,0,1,0)$ olmak üzere parçalanır.
3. $\mathbf{k}_e = H_\ell(\mathbf{e}) = (0,0,0,0,1,0)$ hesaplanır.
4. $\mathbf{m} = (\mathbf{e}_a \mid \mathbf{k}_e) = (0,0 \mid 0,0,0,0,1,0)$ mesaj vektörü inşa edilir.
5. $\mathbf{c}$ mesajı aşağıdaki şekilde hesaplanır:

$$
\begin{aligned}
\mathbf{c} &= (\mathbf{m} \mid \mathbf{m}Q) + \mathbf{e} \\
&= \left[0,0 \mid 0,0,0,0,1,0 \mid \left[0,0 \mid 0,0,0,0,1,0\right] \cdot \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} + \mathbf{e}\right] \\
&= [0,0 \mid 0,0,1,0,1,0 \mid 0,1,0,0,1,1,0,1]
\end{aligned}
$$

ve bu kodlamadan sonra $\mathbf{c}^* = (0,0,1,0,1,0 \mid 0,1,0,0,1,1,0,1)$ olarak belirlenir.

6. $\mathbf{k}_r = H_\ell(\mathbf{k}_e \mid \mathbf{e}) = (0,0,0,0,1,0)$ olmak üzere;

$$(\mathbf{k}_r, \mathbf{c}^*) = (0,0,0,0,1,0,0,0,1,0,1,0 \mid 0,1,0,0,1,1,0,1)$$

ikilisi çıktı olarak verilir.

6.5.4 Kapsülden çıkarma algoritması

Bölüm 6.4 de verilen adımlara uygun şekilde $\mathbf{c}^* = (\mathbf{c}_b \mid \mathbf{c}_c)$ şifreli metni kapsülden çıkarılır:

1. $\mathbf{a}^*$ ve $\mathbf{h}^*$ vektörleri;

$$\mathbf{a}^* = (\mathbf{a}_b \mid \mathbf{a}_c) = (1, \alpha^{10}, \alpha^3, \alpha, \alpha^4, \alpha^{12} \mid \alpha^6, \alpha^9, 0, \alpha^{11}, \alpha^2, \alpha^{13}, \alpha^{14}, \alpha^5)$$

ve

$$\mathbf{h}^* = (\mathbf{h}_b \mid \mathbf{h}_c) = (\alpha^5, \alpha^4, \alpha^{12}, \alpha^{14}, \alpha^5, \alpha^{13} \mid 1, \alpha^{12}, \alpha^{14}, 1, \alpha^8, \alpha^{13}, \alpha^2, \alpha^8)$$

olmak üzere parçalanır.

2. $\mathbf{a}^*$ ve $\mathbf{h}^*$ vektörlerinden,

$$H_m^* = \left[\begin{array}{cccccc|cccccccc} \alpha^5 & \alpha^4 & \alpha^{12} & \alpha^{14} & \alpha^5 & \alpha^{13} & 1 & \alpha^{12} & \alpha^{14} & 1 & \alpha^8 & \alpha^{13} & \alpha^2 & \alpha^8 \\ \alpha^5 & \alpha^{14} & 1 & 1 & \alpha^9 & \alpha^{10} & \alpha^6 & \alpha^6 & 0 & \alpha^{11} & \alpha^{10} & \alpha^{11} & \alpha & \alpha^{13} \\ \alpha^5 & \alpha^9 & \alpha^3 & \alpha & \alpha^{13} & \alpha^7 & \alpha^{12} & 1 & 0 & \alpha^7 & \alpha^{12} & \alpha^9 & 1 & \alpha^3 \\ \alpha^5 & \alpha^4 & \alpha^6 & \alpha^2 & \alpha^2 & \alpha^4 & \alpha^3 & \alpha^9 & 0 & \alpha^3 & \alpha^{14} & \alpha^7 & \alpha^{14} & \alpha^8 \end{array}\right]$$

matrisi inşa edilir.

3. **s** sendrom vektörü aşağıdaki şekilde hesaplanır:

$$\begin{aligned} \mathbf{s} &= (\mathbf{c}_b \mid \mathbf{c}_c) \cdot (H_m^*)^T \\ &= [0,0,1,0,1,0 \mid 0,1,0,0,1,1,0,1] \cdot \left[\begin{array}{cccc} \alpha^5 & \alpha^5 & \alpha^5 & \alpha^5 \\ \alpha^4 & \alpha^{14} & \alpha^9 & \alpha^4 \\ \alpha^{12} & 1 & \alpha^3 & \alpha^6 \\ \alpha^{14} & 1 & \alpha & \alpha^2 \\ \alpha^5 & \alpha^9 & \alpha^{13} & \alpha^2 \\ \alpha^{13} & \alpha^{10} & \alpha^7 & \alpha^4 \\ 1 & \alpha^6 & \alpha^{12} & \alpha^3 \\ \alpha^{12} & \alpha^6 & 1 & \alpha^9 \\ \alpha^{14} & 0 & 0 & 0 \\ 1 & \alpha^{11} & \alpha^7 & \alpha^3 \\ \alpha^8 & \alpha^{10} & \alpha^{12} & \alpha^{14} \\ \alpha^{13} & \alpha^{11} & \alpha^9 & \alpha^7 \\ \alpha^2 & \alpha & 1 & \alpha^{14} \\ \alpha^8 & \alpha^{13} & \alpha^3 & \alpha^8 \end{array}\right] \\ &= (\alpha^7, \alpha^4, \alpha^{14}, \alpha^8) \end{aligned}$$

4. Berlekamp Massey algoritması ile $\sigma^*(x) = 1 + \alpha^{13}x + \alpha^{13}x^2$ polinomu hesaplanır.

5. $\mathbf{\Lambda}$ aşağıdaki gibi hesaplanır:

$$\begin{aligned} \mathbf{\Lambda} &= \{\sigma^*(B[0]), \sigma^*(B[1]), \ldots, \sigma^*(B[n-1])\} \\ &= \{1,0,\alpha^2,\alpha^8,\alpha^{14},\alpha^3,\alpha^6,\alpha^{13},1,0,\alpha^2,\alpha^8,\alpha^{14},\alpha^3,\alpha^6,\alpha^{13}\} \end{aligned}$$

6. $e'_1 = 1$ ve $e'_9 = 1$ olur.

7. $\mathbf{e} = \pi_{\mathbf{p}}(\mathbf{e}') = (0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0)$ hesaplanır.

8. $\mathbf{e} = (0,0 \mid 0,0,1,0,0,0 \mid 0,0,0,0,0,0,1,0)$ olacak şekilde parçalanır ve

$$\mathbf{k}_e = \mathbf{c}_b - \mathbf{e}_b = (0,0,1,0,1,0) - (0,0,1,0,0,0) = (0,0,0,0,1,0)$$

hesaplanır.

9. $H_\ell(\mathbf{e}) = \mathbf{k}_e$ ve $\mathrm{wt}(\mathbf{e}) = \tau = 2$ olduğu doğrulandıktan sonra

$$\mathbf{k}_r = H_\ell(\mathbf{k}_e \mid \mathbf{e}) = (0,0,0,0,1,0) \in \mathbb{F}_2^\ell$$

çıktısı verilir.

7. SONUÇ ve ÖNERİLER

Bu çalışmada NIST tarafından başlatılan süreçte yer alan bazı kod tabanlı anahtar kapsülleme mekanizmaları ve şifreleme sistemleri incelenmiştir. Bu sistemlerin dayandığı kod aileleri hakkında bilgi verilmiştir. Detayları verilen anahtar üretimi, şifreleme ve şifre çözme algoritmaları örneklerle desteklenmiştir. Bu algoritmaların performanslarını incelemek ve analizleriyle ilgili çalışmalar yapmak başka bir araştırma konusu olarak önerilebilir.

KAYNAKLAR

[1] **Akleylek S., Seyhan K.,** (2019). Kuantum Bilgisayarlar Sonrası Güvenilir Kafes Tabanlı Kriptosistem Temellerine Giriş. *Siber Güvenlik ve Savunma: Farkındalık ve Caydırıcılık (Cilt II)*

[2] **Albrecht, M., Cid, C., Paterson, K.G., Tjhai, C.J., Tomlinson, M.,** (2018). NTS-KEM. *NIST submissions*.

[3] **Berlekamp E.R.,** *Algebraic coding theory,* McGraw-Hill series in systems science, McGraw-Hill, (1968).

[4] **Bernstein, D.J., Chou, T., Lange, T., von Maurich, I., Misoczki, R., Niederhagen, R., ... & Szefer, J.,** (2017). Classic McEliece: conservative code-based cryptography. *NIST submissions*.

[5] **Gaborit, P., Murat, G., Ruatta, O., Zémor, G.,** (2013). Low rank parity check codes and their application to cryptography. *In Proceedings of the Workshop on Coding and Cryptography WCC (Vol. 2013).*

[6] **Gaborit P., Ruatta, O., Schrek, J., Tillich, J. P., Zémor, G.,** (2015). Rank based cryptography: a credible post-quantum alternative to classical cryptography. *In NIST 2015: Workshop on Cybersecurity in a Post-Quantum World 2015*.

[7] **Galvez L., Kim J.L., Kim, M.J., Kim, Y., Lee, N.,** (2018). McNie: Compact McEliece-Niederreiter Cryptosystem. *NIST submissions*.

[8] **Jochemsz, E.,** Goppa Codes & the McEliece Cryptosystem *Vrije Universiteit Amsterdam*, 63. (2002).

[9] **Kolman, B., Hill, D.R.,** *Elementary Linear Algebra with Applications*, Pearson Education, Upper Saddle River, New Jersey, (2008).

[10] **Ling, S., Xing, C.,** *Coding theory: a first course*, Cambridge University Press, (2004).

[11] **Massey J.L.,** (1969). Shift-register synthesis and BCH decoding. *IEEE Trans. Information Theory*, 15(1):122-127.

[12] **McEliece R.J.,** (1978). A Public-Key Cryptosystem Based on Algebraic Coding Theory, *Deep Space Network Progress Report, Jet Propulsion Lab.*, 114-116.

[13] **Menezes, A.J., Blake, I.F., Gao, X., Mullin, R.C., Vanstone, S A., Yaghoobian, T.,** *Applications of finite fields*, Vol. 199. Springer Science & Business Media, (2013).

[14] **Niederreiter, H.,** (1986). Knapsack-type cryptosystems and algebraic coding theory, *Prob. Control and Inf. Theory*, 15(2), 159-166.

[15] **Patterson, N.J.,** (1975). The Algebraic Decoding of Goppa Codes. *IEEE Transactions on Information Theory,* 21(2):203-207.

[16] **Prange, E.,** (1962). The use of information sets in decoding cyclic codes. *IRE Trans. Information Theory,* 8(5):5-9.

[17] **Sendrier, N.,** (2017). Code-Based Cryptography: State of the Art and Perspectives. *IEEE Security & Privacy*, 44-50.

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

[19] **Yamada, A., Eaton, E., Kalach, K., Lafrance, P., Parent, A.,** (2017). QC-MDPC KEM: A Key Encapsulation Mechanism Based on the QC-MDPC McEliece Encryption Scheme. *NIST submissions.*

[20] **Url-1** https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions, alındığı tarih: 04 Kasım 2018.

[21] **Url-2** https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions, alındığı tarih: 04 Mart 2019.

[22] **Url-3** http://arxiv.org/abs/1802.06157, alındığı tarih: 03 Eylül 2018.

EKLER

EK 1 : Bazı Temel Algoritmalar

EK 2 : Türkçe-İngilizce Matematik Terimleri Sözlüğü

EK 3 : Algoritmaların Parametre Uzunlukları

EK 1: Bazı Temel Algoritmalar

Berlekamp Massey Algoritması:

Algorithm 1 Berlekamp Massey Algoritması

Require: $\mathbf{s} = (s_0, s_1, \ldots, s_{2\tau-1})$

Require: $\sigma(x) = \sum \sigma_i x^i = 1$

Require: $\beta(x) = \sum \beta_i x^i = x$

Require: $\delta = 1$

Require: $L = R = \xi = 0$

for $i \leftarrow 0$ to $2\tau - 1$ **do**

$$d \leftarrow \sum_{j=0}^{\min\{i,\tau\}} \sigma_j s_{i-j}$$

$\varphi(x) \leftarrow \delta\sigma(x) - d\beta(x)$

if $d == 0$ veya $i < 2L$ **then**

$R \leftarrow R + 1$

$\beta(x) \leftarrow x\beta(x)$

else

$R \leftarrow 0$

$\beta(x) \leftarrow x\sigma(x)$

$L \leftarrow i - L + 1$

$\delta \leftarrow d$

end if

$\sigma(x) \leftarrow \varphi(x)$

end for

if (degree of) $\sigma(x) < (\tau - \frac{R}{2})$ **then**

$\xi \leftarrow 1$

end if

$\sigma^*(x) \leftarrow x^{\tau-\xi}\sigma(x^{-1})$

return $(\sigma^*(x), \xi)$

G(z) Polinomunun Türevi ve İki Polinomun EBOB'u:
$G(z) = \sum_{i=0}^{\tau} g_i z^i$ olsun. Bu durumda $G(z)$ polinomunun türevi;

$$\begin{aligned} \frac{d}{dz}G(z) &= \sum_{i=1}^{\tau} i g_i z^{i-1} \\ &= \sum_{j=0}^{\lceil \tau/2 \rceil - 1} g_{2j+1} z^{2j} \end{aligned}$$

Karakteristiği 2 olan cisimde çalışıldığı için $\frac{d}{dz}G(z)$ sadece çift kuvvetleri içerir ve tam karedir.

$\mathbb{F}_{2^m}[z]$ deki iki polinomun ebobu aşağıdaki algoritmada gösterildiği gibi hesaplanır.

Algorithm 2 $a(z)$ ve $b(z)$'nin en büyük ortak böleni

Require: $\deg a(z) \geq \deg b(z)$
  while $\deg b(z) \geq 0$ **do**
    $t(z) \leftarrow b(z)$
    $b(z) \leftarrow a(z) \pmod{b(z)}$
    $a(z) \leftarrow t(z)$
  end while
  return $a(z)$

EK 2: Türkçe-İngilizce Matematik Terimleri Sözlüğü

Türkçe terim	İngilizce Terim
Ağırlık	Weight
Yarı Devirli	Quasi Cyclic
Düzgün Rastgele	Uniformly at Random
Üreteç Matrisi	Generator Matrix
Eşlik Denetim Matrisi	Parity Check Matrix
Anahtar Kapsülleme Mekanizması	Key Encapsulation Mechanism
Hata Vektörü	Error Vector
Şifreli Metin	Ciphertext
Açık Metin	Plaintext
Gizli Anahtar	Private Key
Açık Anahtar	Public Key
Şifreleme	Encryption
Şifre Çözme	Decryption
Kapsülleme	Encapsulation
Kapsülden Çıkarma	Decapsulation
Kriptoloji	Cryptology
Kod kelimesi	Codeword
Permütasyon Matrisi	Permutation Matrix
Permütasyon Vektörü	Permütation Vector
Karmaşa	Complexity
İkili	Binary
Dizgi	String
Cisim Genişlemesi	Extension Field
Tek Oturumluk	One Session
Gauss Eleme Yöntemi	Gauss Elimination Method
Kodlama	Encoding
Kod Çözme	Decoding
Anahtar Üretimi	Key Generation
Primitif Kök	Primitive Root

EK 3: Algoritmaların Parametre Uzunlukları

Çizelge 7.1: McNie parametre uzunlukları (bit)

Güvenlik	Açık A.	Gizli A.	Şifreli M.
128-bit	431	194	579
	486	218	653
	347	340	422
	417	401	505
192-bit	569	247	764
	631	274	846
	487	465	590
	539	512	651
256-bit	819	337	1097
	829	348	1110
	630	584	761
	647	601	781

Çizelge 7.2: NTS-KEM ve Klasik McEliece parametre uzunlukları (bit)

Güvenlik	Algoritmalar		Açık A.	Gizli A.	Şifreli M.
128-bit	NTS-KEM		319488	9216	128
192-bit	NTS-KEM		929760	17524	162
256-bit	NTS-KEM		1419704	19890	253
256-bit	Klasik McEliece	me6960119	1047319	13908	226
		me8192128	1357824	14080	240

ÖZGEÇMİŞ

Ad-Soyad : Sevde KARA

Uyruğu : T.C.

Doğum Tarihi ve Yeri : 21.08.1992, Sakarya

E-posta : svdekara92@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2016, İhsan Doğramacı Bilkent Üniversitesi, Fen Fakültesi, Matematik Bölümü
- **Yüksek Lisans** : 2019, TOBB Ekonomi ve Teknoloji Üniversitesi, Matematik Anabilim Dalı

MESLEKİ DENEYİM VE ÖDÜLLER:

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

YABANCI DİL: İngilizce, Fransızca

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **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.

- 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 Parametrelerinin Karşılaştırılması, 13. Ankara Matematik Günleri-AMG 2018, 27-28 Nisan, TOBB Ekonomi ve Teknoloji Üniversitesi, Ankara, Türkiye.