T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

NTRU KRİPTOSİSTEMİ

SEVCAN TEKİN

YÜKSEK LİSANS TEZİ
MATEMATİK ANABİLİM DALI

DANIŞMAN
YRD. DOÇ. DR. BAYRAM ALİ ERSOY

İSTANBUL, 2011

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

NTRU KRİPTOSİSTEMİ

Sevcan TEKİN tarafından hazırlanan tez çalışması 07.07.2011 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Yrd. Doç. Dr. Bayram Ali ERSOY

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Yrd. Doç. Dr. Bayram Ali ERSOY

Yıldız Teknik Üniversitesi _____

Doç. Dr. Gürsel YEŞİLOT

Yıldız Teknik Üniversitesi _____

Yrd. Doç. Dr. Bülent KÖKLÜCE

Fatih Üniversitesi _____

ÖNSÖZ

Tez çalışmam boyunca gösterdiği her türlü destek ve yardımlarından dolayı çok değerli hocam Yrd. Doç. Dr. Bayram Ali ERSOY'a en içten dileklerimle teşekkür ederim.

Bugüne kadar maddi ve manevi hiçbir desteğini benden esirgemeyen, bugünlere gelmemi sağlayan, hakkını hiçbir zaman ödeyemeyeceğim babama ve anneme sonsuz teşekkürlerimi sunarım.

Bana her zaman destek olan ve sabır gösteren müstakbel eşim Serkan IŞIKAY'a da ayrıca teşekkür ederim.

Mayıs, 2011

Sevcan TEKİN

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ vi

KISALTMA LİSTESİ vii

ŞEKİL LİSTESİ viii

ÇİZELGE LİSTESİ ix

ÖZET x

ABSTRACT xi

BÖLÜM 1

GİRİŞ 1

1.1 Literatür Özeti 1
1.2 Tezin Amacı 1
1.3 Hipotez 2

BÖLÜM 2

KAFES TEORİSİ 3

2.1 Temel Tanımlar ve Özellikler 3
2.2 Kafes Problemleri 8
2.3 Gram-Schmidt Ortogonalleştirme Metodu 10
2.4 LLL Taban İndirgeme Algoritması 10
2.5 Bazı Özel Kafesler 13
2.5.1 Modüler Kafes 13
2.5.2 NTRU Kafesi 13
2.6 Bölüm Halkaları 14

BÖLÜM 3

NTRU 15

3.1 Parametreler 15

3.2 Bir Polinomun Ortalanması ve Küçük Polinomlar 16
3.3 Anahtar Üretimi 17
3.4 Şifreleme 19
3.5 Deşifreleme 20
3.6 Farklı Güvenlik Seviyeleri İçin Önerilen Parametreler 22

BÖLÜM 4

GELİŞTİRİLMİŞ NTRU 23

4.1 *f* Polinomunun Seçimi 23
4.2 *p*'nin Seçimi 24
4.3 *a* Polinomunun Ortalanması 24
4.4 Örnekler 25
4.5 Küçük Hamming Ağırlıklı Polinomlar 29
4.5.1 Küçük Hamming Ağırlıklı Polinomların Üretilmesi 29

BÖLÜM 5

NTRU ÜZERİNE SALDIRILAR 32

5.1 Kaba Kuvvet Saldırısı 32
5.2 Ortada Karşılaşma Saldırısı 33
5.3 Kafes Saldırıları 35
5.3.1 Standart NTRU Kafesi 36
5.3.2 Sıfır Tekrarlı Kafes 38
5.3.3 Sıfır Kuvvetli Kafes 38
5.3.4 Boyutu İndirgenmiş Kafes 39

BÖLÜM 6

SONUÇ VE ÖNERİLER 41

KAYNAKLAR 42

ÖZGEÇMİŞ 45

SİMGE LİSTESİ

c_h	En kısa vektörün beklenen uzunluğunun hedef vektörün uzunluğuna oranı
$det(\Lambda)$	Λ kafesinin determinantı
γ	Yaklaşık çarpanı
δ	Kafes indirgemede kullanılan keyfi bir sabit
$\mathcal{L}(B))$	B tabanı ile üretilen bir kafes
Λ	Bir kafes
λ	NTRU kafesi oluşturulurken seçilen bir sabit
$\sigma(\mathcal{L}(B))$	Bir $\mathcal{L}(B)$ kafesinin en kısa vektörünün beklenen uzunluğu

KISALTMA LİSTESİ

CVP	The Closest Vector Problem
ECCC	Electronic Colloquim on Computational Complexity
EESS	Efficient Embedded Security Standards
LLL	Lenstra-Lenstra-Lovasz Kafes İndirgeme Metodu
SVP	The Shortest Vector Problem

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 $\mathbb{R}^2$ üzerinde bir kafes 3
Şekil 2.2 Bazı kafes tabanları 4
Şekil 2.3 $\mathcal{L}(B)$'nin gerdiği uzay 5
Şekil 2.4 Gram-Schmidt ortogonalleştirme metodu 10

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1 Farklı güvenlik seviyeleri için önerilen parametreler 22
Çizelge 4.1 Bir NTRU kriptosistemi için parametreler kümesi..... 25

ÖZET

NTRU KRİPTOSİSTEMİ

Sevcan Tekin

Matematik Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Bayram Ali ERSOY

Kafes problemlerinin zorluğu açık anahtar kriptografisine yeni bir aday eklemiştir. Bu konuda, Ajtai-Dwork, GGH ve NTRU gibi birçok kriptosistem geliştirilmiştir. NTRU'nun anahtar üretimi, şifreleme ve deşifreleme işlemlerinin daha hızlı olması ve anahtar boyunun daha küçük olması nedeniyle en çok ilgi gören kafes tabanlı kriptosistemdir. Günümüzde NTRU üzerine çalışmalar yoğun bir şekilde devam etmektedir ve şimdiye kadar hiçbir güvenlik açığı bulunamamıştır. Bu tezin amacı, NTRU kriptosisteminin altında yatan matematiksel yapıyı ortaya koymaktır.

Anahtar Kelimeler: NTRU, Kafes Tabanlı Kriptosistem, Kafes Problemleri, Kafes Saldırıları

ABSTRACT

NTRU CRYPTOSYSTEM

Sevcan TEKİN

Department of Mathematics

MSc. Thesis

Advisor: Assist. Prof. Dr. Bayram Ali ERSOY

Hardness of lattice problems has added a new candidate for public key cryptography. On this subject, several cryptosystems have been developed such as Ajtai-Dwork, GGH and NTRU cryptosystems. NTRU is the most attractive lattice based cryptosystem due to the fact that its key generation, encryption and decryption operations are relatively faster and key size is smaller. Today, researches have been extensively going on NTRU and any security issue has not been found so far. The aim of this thesis is to demonstrate the mathematical structure underlying NTRU cryptosystem.

Key words: NTRU, Lattice Based Cryptosystem, Lattice Problems, Lattice Attacks

BÖLÜM 1

GİRİŞ

1.1 Literatür Özeti

Şifreleme, bir gereksinim olarak ortaya çıkmasından 20. Yüzyılın son çeyreğine kadar, hep tek bir anahtar kullanılarak gerçekleştirildi. Yani veriyi şifrelerken ve şifreli veriyi çözerken kullanılan anahtar aynıydı. Simetrik kriptografi olarak adlandırılan bu yöntemin çok hızlı olmasına rağmen, geniş kullanıcı ağlarında anahtar paylaşımı en büyük sorunlarından biridir.

1976'da Diffie ve Hellman iki farklı anahtar kullanan açık anahtar kriptografi kavramını ortaya attı [1]. Bu buluş kriptoloji tarihinde bir devrim niteliğindedir. Yıllardır kullanılan tek anahtarlı şifre yapısı yerine çift anahtarlı şifre yapısı benimseneyerek, anahtar paylaşım problemi çözüldü. Daha sonra birçok açık anahtar kriptosistemi önerildi. 1978'de hala en çok kullanılan RSA açık anahtar kriptosistemi tanıtıldı [2]. RSA'nın güvenliği matematiğin en zor problemlerinden biri olan büyük sayıları çarpanlara ayırma zorluğuna dayanır. Diğer iyi bilinen açık anahtar kriptosistemi ise 1985'te sunulan Eliptik Eğri Kriptografisidir [3]. Bu sistemin güvenliği eliptik eğri logaritmasına dayanır.

1.2 Tezin Amacı

Son yirmi yıl içerisinde, kafesler kriptanalizde önemli bir rol oynamıştır. 1996'ya kadar kafesler ve özellikle kafes tabanı indirgeme algoritmaları bazı sistemlerin güvensizliğini kanıtlamak için kullanılmıştır. Ancak 1996 ve 1997'de Ajtai'nin yayınladığı makaleler sonucunda, bu konunun kriptografide de kullanılabileceği önerilmiştir [4], [5]. Yapılan

araştırmalar sonucunda birkaç kriptosistem sunulmuştur. Bunlarda biri Goldreich, Goldwasser ve Halevi'nin GGH kriptosistemidir [6]. GGH'nin güvenliği CVP'yi çözme zorluğuna dayanır.

Bu tezde, diğer önemli kafes tabanlı kriptosistem olan, NTRU kriptosistemi üzerinde çalışılmıştır. Aslında NTRU belirli polinom halkalarının cebirsel yapıları üzerine inşa edilmiştir, fakat güvenliği en kısa kafes vektörünü bulma zorluğuna dayanır. NTRU kriptosistemine karşı bilinen en iyi saldırı temel olarak kafes indirgeme algoritmalarını kullanan kafes saldırılarıdır. NTRU'nun diğer açık anahtar kriptosistemlerinden avantajları şunlardır: açık anahtar ve gizli anahtar çiftini üretmek kolay ve hızlıdır, şifreleme ve deşifreleme işlemleri çok hızlıdır ve anahtar boyutları oldukça küçüktür.

1.3 Hipotez

Bu tez 5 bölümden oluşmaktadır. 2. bölümde kafes teorisi hakkında kısaca bilgi verilmiştir. Kafesler ile ilgili temel tanım ve notasyonlara, ayrıca kafes problemlerine değinilmiştir. Daha sonra NTRU'nun kriptanalizinde kullanılan LLL kafes tabanı indirgeme algoritması verilmiştir. Sonraki bölümde, [7] de gösterilmiş olan ilk versiyonunda olduğu gibi, NTRU'nun şifreleme ve deşifreleme işlemleri verilmiştir. Deşifrelemedeki başarısızlıkları önlemek ve işlemleri hızlandırmak için yapılan geliştirmeler ise 4. bölümde açıklanmıştır. Son bölümde ise NTRU üzerine yapılan saldırılar ele alınmıştır.

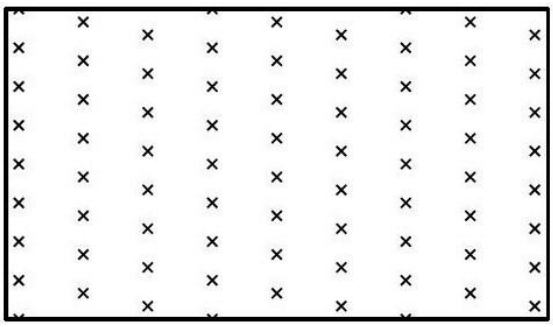
BÖLÜM 2

KAFES TEORİSİ

Bu bölümde kafeslerin temel özellikleri, önemli kafes problemleri ve LLL kafes indirgeme algoritması hakkında kısaca bilgi verilmiştir. Kafes teorisi hakkında daha detaylı bilgi [8], [9] kaynaklarında mevcuttur.

2.1 Temel Tanımlar ve Özellikler

Kafes, Şekil 2.1'de gösterildiği gibi periyodik bir yapıya sahip n-boyutlu uzayda bir noktalar kümesidir. Bu konu ilk olarak 18. Yüzyılın sonlarında Lagrange, Gauss ve Minkowski tarafından incelenmiştir.



Şekil 2.1 $\mathbb{R}^2$ üzerinde bir kafes

Tanım 2.1 $b_1, b_2, \dots, b_n \in \mathbb{R}^m$ n tane lineer bağımsız vektör olsun. Bu vektörlerin tamsayı katsayılı lineer kombinasyonlarının oluşturduğu kümeye kafes denir ve

$$\Lambda = \mathcal{L}(b_1, b_2, \dots, b_n) = \left\{ \sum_{i=1}^{n} x_i b_i \;\middle|\; x_i \in \mathbb{Z} \right\}$$

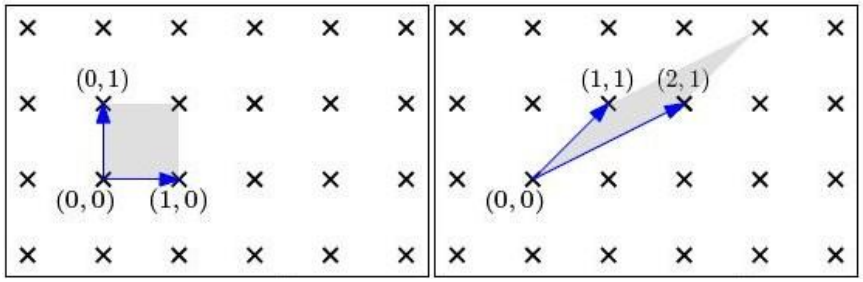
şeklinde gösterilir.

Burada $\{b_1, b_2, \dots, b_n\}$ vektörlerine kafesin tabanı (veya bazı) denir. $B = [b_1 \ b_2 \ \dots \ b_n] \in \mathbb{R}^{m \times n}$ taban vektörlerini sütun kabul eden $m \times n$ tipinde matris olmak üzere, B tarafından üretilen kafes

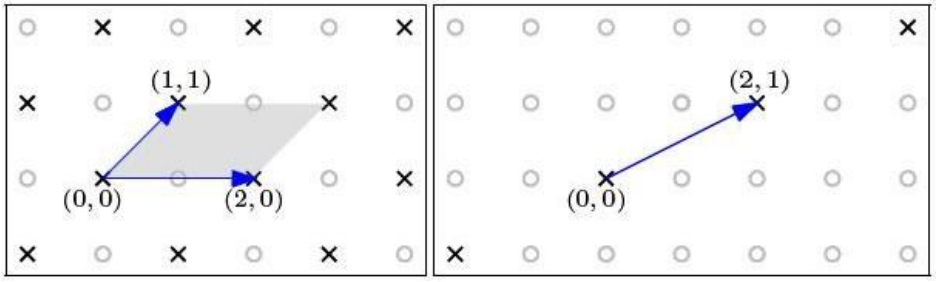
$$\Lambda = \mathcal{L}(B) = \mathcal{L}(b_1, b_2, \dots, b_n) = \{Bx \mid x \in \mathbb{Z}^n\}$$

şeklinde gösterilir.

n tamsayısına kafesin rankı adı verilir ve $rank(B)$ ile gösterilir. m tamsayısına ise kafesin boyutu adı verilir ve $\dim(\mathcal{L}(B))$ ile gösterilir. Rankı ve boyutu birbirine eşit olan kafese tam-rank kafes (full-rank lattice) denir. Tezin geri kalanında, aksi söylenmedikçe kafesler tam-rank olarak düşünülecektir. Ayrıca $b_1, b_2, \dots, b_n \in \mathbb{Z}^n$, yani taban vektörlerinin tamsayı bileşenlere sahip oldukları kabul edilecektir.



(a) $\mathbb{Z}^2$ nin bir tabanı (b) $\mathbb{Z}^2$ nin diğer bir tabanı



(c) $\mathbb{Z}^2$ nin bir tabanı değildir. (d) Tam rank olmayan bir kafes

Şekil 2.2 Bazı kafes tabanları

Şekil 2.2.a'daki $(1,0)^T$ ve $(0,1)^T$ vektörlerinin ürettiği kafes $\mathbb{Z}^2$ dir. Bir kafesin tabanı tek türlü değildir. Örneğin, Şekil 2.2.b'deki $(1,1)^T$ ve $(2,1)^T$ vektörleri de $\mathbb{Z}^2$ yi üretir. $\mathbb{Z}^2$ için başka bir taban $(2005,1)^T$ ve $(2006,1)^T$ olabilir. Öte yandan Şekil 2.2.c'deki $(1,1)^T$ ve $(2,0)^T$ vektörleri $\mathbb{Z}^2$ için bir taban oluşturmaz. Bu vektörler koordinatları toplamı çift sayı olan noktaların oluşturduğu kafesi üretir. Şimdiye kadar olan tüm

örnekler tam-rank kafeslerdir. Ancak Şekil 2.2.d'deki $\mathcal{L}((2,1)^T)$ kafesi tam-rank olmayan bir kafes örneğidir. $\mathcal{L}((2,1)^T)$ kafesinin boyutu 2, rankı 1 dir. 1 boyutlu tam-rank kafese örnek olarak $\mathbb{Z} = \mathcal{L}\big((1)\big)$ kafesi verilebilir.

Tanım 2.2 Bir $\mathcal{L}(B)$ kafesinin gerdiği uzay

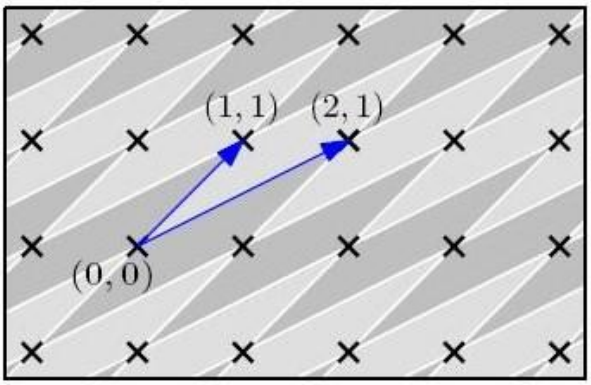
$$span\big(\mathcal{L}(B)\big) = span(B) = \{By \mid y \in \mathbb{R}^n\}$$

şeklinde tanımlanır.

Tanım 2.3 Herhangi bir B kafes tabanı için temel paralelyüzlü (fundamental parallelepiped)

$$\mathcal{P}(B) = \{\, Bx \mid x \in \mathbb{R}^n, \forall i: 0 \le x_i < 1 \,\}$$

şeklinde tanımlanır. Temel paralelyüzlüye örnek olarak Şekil 2.2'deki gri alanlar gösterilebilir. $\mathcal{P}(B)$, B tabanına bağlıdır. $\mathcal{L}(B)$ kafesindeki her noktaya $\mathcal{P}(B)$'nin bir kopyası yerleştirilirse Şekil 2.3'te gösterildiği gibi $span\big(\mathcal{L}(B)\big)$ elde edilir.



Şekil 2.3 $\mathcal{L}(B)$'nin gerdiği uzay

Şekil 2.2.c'de görüldüğü gibi $\mathbb{Z}^n$ deki her n tane lineer bağımsız vektör kümesi, $\mathbb{Z}^n$ nin bir tabanı olmayabilir. Aşağıdaki önermede, n tane lineer bağımsız kafes vektörünün taban olabilmesi için gerek ve yeter koşul verilmiştir.

Önerme 2.1 Λ, rankı n olan bir kafes olsun. Ayrıca $b_1, b_2, \dots, b_n \in \Lambda$ n tane lineer bağımsız kafes vektörü olsun. $\{b_1, b_2, \dots, b_n\}$ kümesinin Λ kafesinin bir tabanı olması için gerek ve yeter koşul $\mathcal{P}(b_1, b_2, \dots, b_n) \cap \Lambda = \{0\}$ olmasıdır.

İspat. $\{b_1, b_2, \dots, b_n\}$ Λ kafesinin bir tabanı olsun. Tanım gereği Λ, bu vektörlerin tamsayı katsayılı lineer kombinasyonlarının oluşturduğu kümedir. $\mathcal{P}(b_1, b_2, \dots, b_n)$ ise

$b_1, b_2, \dots, b_n$ vektörlerinin, katsayıları $[0,1)$ aralığından olan lineer kombinasyonlarının kümesidir. Bu durumda iki kümenin kesişimi $\{0\}$ dır.

Tersine, $\mathcal{P}(b_1, b_2, \dots, b_n) \cap \Lambda = \{0\}$ olsun. Λ, rankı n olan bir kafes ve $b_1, b_2, \dots, b_n$ lineer bağımsız vektörler olsun. Bu durumda, her $x \in \Lambda$ için $x = \sum y_i b_i$ yazılabilir. Burada $\forall\, i = 1,2, \dots, n$ için $y_i \in \mathbb{R}$ olduğunu fazedelim. $y_i - \lfloor y_i \rfloor \in [0,1)$ olacağı açıktır. Şu halde $x' = \sum (y_i - \lfloor y_i \rfloor)\, b_i \in \mathcal{P}(B)$ dir. Ayrıca bir kafes, toplama işlemi altında kapalı olduğundan $x' \in \Lambda$ 'dır. Varsayım gereği $x' = 0$ olmalıdır. Bu durumda $y_i - \lfloor y_i \rfloor = 0$, dolayısıyla $y_i \in \mathbb{Z}$ dir. Böylece x; $b_1, b_2, \dots, b_n$ vektörlerinin bir tamsayı katsayılı kombinasyonudur.■

Sonuç olarak verilen n tane lineer bağımsız kafes vektörünün taban olabilmesi için gerek ve yeter koşul bu vektörler tarafından üretilen temel paralelyüzlünün orijin hariç hiçbir kafes noktasını içermemesidir. Örneğin Şekil 2.2.c'de görülen temel paralelyüzlü $(0,1)$ kafes noktasını içerir. Bu durumda verilmiş olan vektörler taban oluşturmaz. Oysa Şekil 2.2.a ve 2.2.b'deki temel paralelyüzlüler sıfırdan farklı herhangi bir kafes noktasını içermez.

Tanım 2.4 $U \in \mathbb{Z}^{n\times n}$ olsun. $\det U = \pm 1$ ise U matrisine ünimodüler matris (unimodular matrix) denir.

Örneğin, $\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$ bir ünimodüler matristir.

Aşağıdaki önermede bir ünimodüler matrisin tersinin de ünimodüler matris olduğu ifade edilmiştir. Bu durumda ünimodüler matrisler kümesi matrislerdeki çarpma işlemi altında bir grup oluşturur.

Önerme 2.2 U ünimodüler matris ise U^{-1} de ünimodüler matristir ve $U^{-1} \in \mathbb{Z}^{n\times n}$ dir.

Önerme 2.3 $B_1, B_2 \in \mathbb{R}^{m\times n}$ tabanlarının denk olması için gerek ve yeter koşul $B_2 = B_1 U$ olacak şekilde bir U ünimodüler matrisin olmasıdır.

İspat. İlk olarak $B_1, B_2 \in \mathbb{R}^{m\times n}$ tabanlarının denk olduğunu yani $\mathcal{L}(B_1) = \mathcal{L}(B_2)$ olduğunu kabul edelim. B_2 tabanının her b_i $(1 \le i \le n)$ sütunu için $b_i \in \mathcal{L}(B_1)$ dir. Bu durumda $B_2 = B_1 U$ olacak şekilde bir $U \in \mathbb{Z}^{n\times n}$ vardır. Benzer şekilde $B_1 = B_2 V$

olacak şekilde $V \in \mathbb{Z}^{n\times n}$ vardır. Böylece $B_2 = B_1U = B_2VU$ elde edilir. Ayrıca $B_2^T = (B_2VU)^T$ olduğundan elde edilen bu iki eşitlik taraf tarafa çarpılırsa

$B_2^T B_2 = (VU)^T B_2^T B_2 VU$

ve determinant alınırsa

$\det(B_2^T B_2) = \left(det(VU)\right)^2 det(B_2^T B_2)$

elde edilir. Böylece

$det(VU) = det(V)det(U) = \pm 1$

bulunur. U ve V tamsayı matrisler olduğundan $det(U) = \pm 1$ elde edilir.

Tersine, U ünimodüler matris olmak üzere $B_2 = B_1U$ olsun. B_2'nin her sütunu $\mathcal{L}(B_1)$ içinde olacağından $\mathcal{L}(B_2) \subseteq \mathcal{L}(B_1)$ dir. Ayrıca $B_1 = B_2U^{-1}$ ve U^{-1} ünimodüler matris olduğundan benzer şekilde $\mathcal{L}(B_1) \subseteq \mathcal{L}(B_2)$ elde edilir. Dolayısıyla $\mathcal{L}(B_1) = \mathcal{L}(B_2)$ bulunur.■

Sonuç olarak, B matrisinin $\mathbb{Z}^n$ kafesinin bir tabanı olması için gerek ve yeter koşul B matrisinin ünimodüler olmasıdır. (Şekil 2.2'deki örnekler bunu sağlar.)

Tanım 2.5 $\Lambda = \mathcal{L}(B)$, rankı n olan bir kafes olsun. $det(\Lambda)$ ile gösterilen Λ kafesinin determinantı

$\det(\Lambda) = \sqrt{det(B^T B)}$

dir.

Λ tam-rank kafes ise B matrisi kare matris olacağından $det(\Lambda) = |det(B)|$ dir.

Bir kafesin determinantı, B tabanının seçiminden bağımsızdır. Gerçekten B_1 ve B_2, Λ kafesinin iki tabanı ise Önerme 2.3'ten $B_2 = B_1U$ olacak şekilde bir U unimodular matris vardır. Böylece

$$\sqrt{det(B_2^T B_2)} = \sqrt{det(U^T B_1^T B_1 U)} = \sqrt{det(B_1^T B_1)}$$

dir.

2.2 Kafes Problemleri

Kafesler ile ilgili birçok problem vardır. En kısa vektör problemi (the shortest vector problem, SVP), verilen bir kafeste sıfırdan farklı en kısa kafes vektörünü bulma problemidir. Farklı algoritmik zorlukları ele almak için bu problemin farklı versiyonları tanımlanmıştır: en kısa vektörü bulmak, en kısa vektörün uzunluğunu bulmak ve verilen bir sayıdan daha kısa olup olmadığına karar vermek. Bir $\mathcal{L}(B)$ kafesinin en kısa vektörünün beklenen uzunluğu $\sigma(\mathcal{L}(B))$ ile gösterilir.

- Arama SVP: Verilen bir $B \in \mathbb{Z}^{m\times n}$ kafes tabanı için $\|v\| = \sigma(\mathcal{L}(B))$ olacak şekilde $v \in \mathcal{L}(B)$ yi bulma problemidir.
- Optimizasyon SVP: Verilen bir $B \in \mathbb{Z}^{m\times n}$ kafes tabanı için $\sigma(\mathcal{L}(B))$'yi bulma problemidir.
- Karar SVP: Verilen bir $B \in \mathbb{Z}^{m\times n}$ kafes tabanı ve $r \in \mathbb{Q}$ rasyonel sayısı için $\sigma(\mathcal{L}(B)) \leq r$ olup olmadığına karar verme problemidir.

Burada kafes tabanı reel vektörler yerine tamsayı vektörlerden oluşmasıyla kısıtlanmıştır. Bunun amacı girdiyi sonlu sayıda bit ile gösterilebilir yapmaktır. Böylece SVP standart bir bilişimsel problem olarak düşünülebilir.

Bu problemlerin zorluğu nedeniyle yaklaşık versiyonları tanımlanmıştır. Burada en kısa vektörü bulmak yerine, bu en kısa vektörün bir yaklaşığını bulmak ile ilgilenilir. Yaklaşık çarpanı $\gamma \geq 1$ parametresi ile gösterilir.

- Arama SVP_γ: Verilen bir $B \in \mathbb{Z}^{m\times n}$ kafes tabanı için $v \neq 0$ ve $\|v\| \leq \gamma \cdot \sigma(\mathcal{L}(B))$ olacak şekilde $v \in \mathcal{L}(B)$ vektörünü bulma problemidir.
- Optimizasyon SVP_γ: Verilen bir $B \in \mathbb{Z}^{m\times n}$ kafes tabanı için $d \leq \sigma(\mathcal{L}(B)) \leq \gamma \cdot d$ olacak şekilde d'yi bulma problemidir.
- Söz SVP_γ: Bir $B \in \mathbb{Z}^{m\times n}$ kafes tabanı ve $r \in \mathbb{Q}$ rasyonel sayısı için bir problem örneği olan (B, r) çifti verilsin.

  $\sigma(\mathcal{L}(B)) \leq r$ ise (B, r) bir EVET örneği

  $\sigma(\mathcal{L}(B)) > \gamma \cdot r$ ise (B, r) bir HAYIR örneğidir.

Son versiyon genellikle GapSVP_γ ile gösterilir. Bu bir söz verme problemidir. Bunun anlamı bir problem evet örneği ve hayır örneği gibi iki ayrık girdiler kümesi ile tanımlanır. Amaç girdinin hangi kümeden alındığını belirlemektir. Karar probleminin tersine, bu kümelerin birleşimi tüm olası girdileri içermek zorunda değildir. Yani, algoritmanın davranışını tanımsız yapan illegal girdiler de vardır.

Diğer temel kafes problemi en yakın vektör problemi (the closest vector problem,CVP) dir. Burada amaç, uzayda verilen bir noktaya en yakın kafes noktasını bulmaktır. Daha önce olduğu gibi bir yaklaşık çarpanı $\gamma \geq 1$ için CVP'nin üç versiyonu tanımlanmıştır.

- Arama CVP_γ: Verilen bir $B \in \mathbb{Z}^{m\times n}$ kafes tabanı ve $t \in \mathbb{Z}^m$ vektörü için $\|v - t\| \leq \gamma \cdot dist\big(t, \mathcal{L}(B)\big)$ olacak şekilde $v \in \mathcal{L}(B)$ vektörünü bulma problemidir.
- Optimizasyon CVP_γ: Verilen bir $B \in \mathbb{Z}^{m\times n}$ kafes tabanı ve $t \in \mathbb{Z}^m$ vektörü için $d \leq dist\big(t, \mathcal{L}(B)\big) \leq \gamma \cdot d$ olacak şekilde d'yi bulma problemidir.
- Söz CVP_γ: Bir $B \in \mathbb{Z}^{m\times n}$ kafes tabanı ve $t \in \mathbb{Z}^m$ vektörü için bir problem örneği olan (B, t, r) üçlüsü verilsin.

  $dist\big(t, \mathcal{L}(B)\big) \leq r$ ise (B, t, r) bir EVET örneği

  $dist\big(t, \mathcal{L}(B)\big) > \gamma \cdot r$ ise (B, t, r) bir HAYIR örneğidir.

$\gamma = 1$ olduğu zaman SVP_γ ve CVP_γ sırasıyla SVP ve CVP ye denk olur. Ayrıca CVP'nin SVP'den daha kolay olmadığı [10] da kanıtlanmıştır. Bu problemin karmaşıklık sonuçları Cai tarafından [11] de verilmiştir.

Minimum taban problemi ise verilen bir kafeste taban vektörlerinin uzunluklarının çarpımı minimum olan tabanı bulma problemidir. Λ kafesinin bir tabanı $B = \{b_1, b_2, \dots, b_n\}$ olmak üzere taban vektörlerinin uzunluklarının çarpımı

$$\prod_{i=1}^{n} \|b_i\|$$

ile gösterilir ve B tabanının ağırlığı olarak da adlandırılır.

2.3 Gram-Schmidt Ortogonalleştirme Metodu

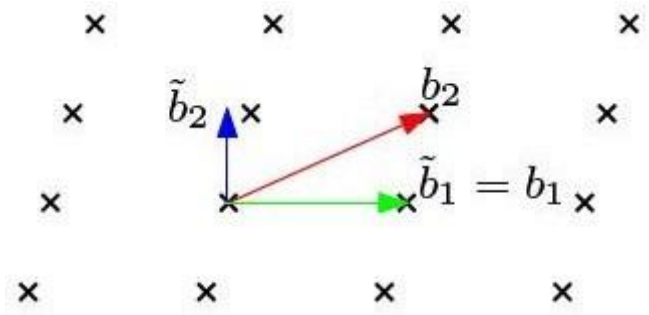
n tane lineer bağımsız vektör kümesinden n tane ortogonal(dik) vektör kümesi elde etme metodudur.

Tanım 2.6 $b_1, b_2, \dots, b_n$ n tane lineer bağımsız vektör dizisi için, bu vektörleri ortogonalleştirme işlemi şu şekilde tanımlanır:

$$\tilde{b}_i = b_i - \sum_{j=1}^{i-1} \mu_{i,j}\, \tilde{b}_j \qquad \mu_{i,j} = \frac{\langle b_i, \tilde{b}_j \rangle}{\langle \tilde{b}_j, \tilde{b}_j \rangle}$$

Gram-Schmidt ortogonalleştirme metodunun bazı temel özellikleri şunlardır:

- $\forall i \neq j$ için $\langle \tilde{b}_i, \tilde{b}_j \rangle = 0$
- $\forall\, 1 \leq i \leq n$ için $span(b_1, b_2, \dots, b_n) = span(\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_n)$
- $\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_n$ vektörleri $\mathcal{L}(b_1, b_2, \dots, b_n)$ kafesinin bir tabanı olmak zorunda değildir. Hatta genellikle kafesin içinde bulunmazlar (Şekil 2.4).



Şekil 2.4 Gram-Schmidt ortogonalleştirme metodu

2.4 LLL Taban İndirgeme Algoritması

Bir kafesin sonsuz sayıda tabanı vardır ancak oldukça kısa ve dik vektörler içeren tabanlar diğerlerinden daha önemlidir. Kafes indirgeme algoritmalarının amacı, kafesin kısa vektörlerden oluşan bir tabanını ve en kısa vektörünü bulmaktır.

Kafesin boyutu küçük ise en kısa vektör sıralama teknikleriyle ayrıntılı arama kullanılarak bulunabilir [12]. Ancak büyük boyutlu kafeslerde ayrıntılı aramanın çalışma süresi üsteldir. 1982'ye kadar, yüksek boyutlu kafeslerde kısa vektörleri bulan bir algoritma yoktu. 1982'de Lenstra, Lenstra ve Lovasz kafes indirgeme metodu olan LLL

algoritmasını sundu [13]. LLL polinom zamanda yaklaşık olarak kısa vektörleri bulan tek algoritmadır.

Kafeslerde kısa vektörleri bulmak için bir yaklaşım, dik bir taban elde etmektir. Eğer bir kafes dik bir tabana sahipse en kısa vektörü taban vektörlerinden biridir.

Önerme 2.4 Λ bir kafes ve $B = \{b_1, b_2, \dots, b_n\}$ bu kafesin dik bir tabanı olsun. Kafesin en kısa vektörü, taban vektörlerinden biridir.

İspat. $b \in \Lambda$ olsun. $B = \{b_1, b_2, \dots, b_n\}$ taban olduğundan $x_i \in \mathbb{Z}$ için

$$b = x_1 b_1 + x_2 b_2 + \dots + x_n b_n$$

ve buradan

$$\|b\| = \|x_1 b_1 + x_2 b_2 + \dots + x_n b_n\|$$

yazılabilir.

B tabanı dik olduğundan,

$$\|b\|^2 = {x_1}^2 \|b_1\|^2 + \dots + {x_n}^2 \|b_n\|^2$$

$$\forall\, i \in \{1,2,..,n\}\ için \qquad \|b\|^2 \geq \|b_i\|^2$$

$$\|b\| \geq \|b_i\| \geq \min_{1 \leq i \leq n} \|b_i\|$$

elde edilir. ■

LLL algoritmasına geçmeden önce LLL indirgenmiş tabanı tanımlayalım.

Tanım 2.7 $B = \{b_1, b_2, \dots, b_n\}$, Λ kafesinin bir tabanı olsun. Aşağıdakiler sağlanırsa B tabanına LLL indirgenmiş denir.

i) $1 \leq i < j \leq n$ için $\left|\mu_{i,j}\right| \leq \frac{1}{2}$ (2.1)

ii) $\forall\, 1 < i \leq n$ için $\left\|\tilde{b}_{i+1} + \mu_{i+1,i}\tilde{b}_i\right\|^2 \geq \delta \left\|\tilde{b}_i\right\|^2$ (2.2)

δ parametresi $\left(\frac{1}{4}, 1\right)$ aralığında keyfi bir değerdir. Fakat δ değeri arttıkça tabanın indirgemesi zorlaşır.

Tanım 2.7 LLL algoritması, $\delta \in \left(\frac{1}{4}, 1\right)$ ve $B = \{b_1, b_2, \dots, b_n\}$ kafes tabanından bir LLL indirgenmiş tabanını şu şekilde elde eder:

$k = 2$
while $k \leq n$
    b_k vektörünün boyutu indirgenir ve $j = 1, \dots, k-1$ için $\mu_{k,j}$ tekrar hesaplanır.
    if $\left\|\tilde{b}_k + \mu_{k,k-1}\tilde{b}_{k-1}\right\|^2 < \delta\left\|\tilde{b}_{k-1}\right\|^2$
        Then swap (b_k, b_{k-1})
            $k = \max(k-1, 2)$
    Else $k = k + 1$
Return $b_1, b_2, \dots, b_n$

Eğer $j = 1, \dots, k-1$ için $\mu_{k,j}$ (2.1) eşitsizliğini sağlamazsa b_k'nın boyutu şu şekilde indirgenir:

$$\tilde{b}_k = b_k - \sum_{j<k} \lfloor \mu_{k,j} \rceil \tilde{b}_j$$

Burada $\lfloor \mu_{k,j} \rceil$, $\mu_{k,j}$'ye en yakın tamsayıyı gösterir. Algoritmanın ikinci kısmında (2.2) eşitsizliğinin sağlanıp sağlanmadığı test edilir. Eğer sağlanmazsa b_{k-1} ve b_k taban vektörleri yer değiştirir. Bu durumda $\mu_{k-1,j}$ ve $\mu_{k,j}$ değerleri değişeceğinden tekrar hesaplanır. $k = n + 1$ e kadar bu iterasyonlar devam eder.

Yüksek boyutlu kafeslerde, kafes indirgeme algoritmaları en kısa vektörü tam olarak bulamaz. Bunun yerine en kısa vektörün γ katına eşit veya daha küçük uzunluklu bir kafes vektörü bulur. Bu nedenle böyle algoritmalar, $\|b_1\| \leq \gamma \cdot \sigma(\Lambda)$ olacak şekilde yaklaşık en kısa b_1 vektörü ile $B = \{b_1, b_2, \dots, b_n\}$ indirgenmiş tabanı bulur.

Bu bölümde LLL hakkında kısaca bilgi verilmiştir. LLL için yapılan geliştirmeler ve diğer kafes indirgeme algoritmaları hakkında daha detaylı bilgi [14], [15], [16] kaynaklarında mevcuttur.

2.5 Bazı Özel Kafesler

2.5.1 Modüler Kafes

Tanım 2.8 Bir modüler kafes

$$\boldsymbol{L} = \begin{bmatrix} b\boldsymbol{I} & \boldsymbol{H} \\ \boldsymbol{0} & q\boldsymbol{I} \end{bmatrix}$$

matrisinin satırları tarafından gerilen kafestir. $L_{ML} = rowspan(\boldsymbol{L})$ şeklinde gösterilir. Burada $b, q \in \mathbb{Z}$ olup $2n$ boyutlu $\boldsymbol{L}$ matrisinin tüm elemanları $n \times n$ matrislerdir. $\boldsymbol{I}$ birim matrisi ve $\boldsymbol{0}$ ise 0 matrisini temsil etmektedir. $\boldsymbol{H}$ matrisinin elemanları $mod\ q$'ya göre indirgenmiş haldedir. $\boldsymbol{H}$ matrisi

$$\boldsymbol{H} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_{n-1} & h_0 & \cdots & h_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \cdots & h_0 \end{bmatrix}$$

şeklinde alınırsa, L_{ML} evrişim modüler kafesi ya da dairesel modüler kafes (convolution or circulant modular lattice) olarak adlandırılır ve L_{CML} olarak da gösterilir. Burada

$$\boldsymbol{h} = [h_0 \quad h_1 \quad \cdots \quad h_{n-1}]$$

olmak üzere $\boldsymbol{H}$ matrisi, $\boldsymbol{h}$ vektörünün dairesel matrisi (circulant matrix) olur ve $\boldsymbol{M_h}$ ile gösterilir. Açıkça görülebileceği gibi $\boldsymbol{M_h}$ matrisinin satırları, $\boldsymbol{h}$ vektörünün elemanlarının dairesel olarak kaydırılmasından elde edilmektedir. Ayrıca

$$[a_0 \quad a_1 \quad \cdots \quad a_{n-1} \quad b_0 \quad b_1 \quad \cdots \quad b_{n-1}]$$

vektörü evrişim modüler kafesin elemanı ise

$$k = 1,2,\dots,n-1\ için \quad [a_k \quad a_{k+1} \quad \cdots \quad a_{k-1} \quad b_k \quad b_{k+1} \quad \cdots \quad b_{k-1}]$$

şeklindeki tüm vektörler de bu kafesin elemanıdır.

2.5.2 NTRU Kafesi

Tanım 2.9 Aşağıdaki evrişim kafesini ele alalım.

$$L_{CML} = rowspan\left(\boldsymbol{L} = \begin{bmatrix} b\boldsymbol{I} & \boldsymbol{M_h} \\ \boldsymbol{0} & q\boldsymbol{I} \end{bmatrix} \right)$$

Eğer bu kafes aynı zamanda

$$[\boldsymbol{f} \quad \boldsymbol{g}] = [f_0 \quad f_1 \quad \cdots \quad f_{n-1} \quad g_0 \quad g_1 \quad \cdots \quad g_{n-1}]$$

şeklinde bir kısa vektör içeriyorsa, NTRU kafesi (NTRU Lattice) olarak adlandırılır ve L_{NTRU} olarak gösterilir. Buradaki b değerine dengeleme sabiti (balancing constant) denilmektedir.

2.6 Bölüm Halkaları

$M(x) \in \mathbb{Z}_q[x]$ derecesi n olan bir monik polinom olsun. ${\mathbb{Z}_q[x]}/{\langle M(x) \rangle}$ bölüm halkasında herhangi bir $h(x)$ polinomu alındığında

$$L_h = \{ [\boldsymbol{f} \quad \boldsymbol{g}] \mid g(x) \equiv f(x) * h(x) \quad (mod\ q, M(x)) \}$$

* işlemi evrişim çarpımı ve $\boldsymbol{f}$, $\boldsymbol{g}$ sırasıyla bölüm halkasındaki $f(x)$, $g(x)$ polinomlarının katsayı vektörleri olmak üzere, L_h kafesi bir modüler kafes olmaktadır. Eğer $M(x) = x^n - 1$ olarak seçilirse bu kafes evrişim modüler kafesi olur.

Ayrıca $R_q = {\mathbb{Z}_q[x]}/{\langle x^n - 1 \rangle}$ halkasında bir NTRU kafesi üretmek için basit bir yol bulunmaktadır. $f(x), g(x) \in R_q$ polinomları bir evrişim modüler kafesi tanımlamada kullanılabilir. Bunun için yapılması gereken $h(x)$ polinomunu hesaplamaktır. $f(x)$ polinomunun R_q halkasında tersi $f^{-1}(x)$ olduğu kabul edilirse

$$h(x) = f^{-1}(x) * g(x) \quad (mod\ q, \langle x^n - 1 \rangle)$$

şeklinde hesaplanabilir. Ayrıca hesaplanan $h(x)$ polinomuna karşılık gelen kafesin

$$L_h = rowspan\left(\begin{bmatrix} \boldsymbol{I} & \boldsymbol{M_h} \\ \boldsymbol{0} & q\boldsymbol{I} \end{bmatrix} \right)$$

olduğu göz önünde bulundurulursa

$$[\boldsymbol{f} \quad \boldsymbol{g}] = [f_0 \quad f_1 \quad \cdots \quad f_{n-1} \quad g_0 \quad g_1 \quad \cdots \quad g_{n-1}]$$

vektörü bu kafesin elemanı olan kısa vektördür.

BÖLÜM 3

NTRU

NTRU halka tabanlı bir asimetrik kriptosistemdir. İlk olarak J. Hoffstein, J. Piper ve J. H. Silverman [7] tarafından sunulmuştur. İlk versiyonundan sonra birkaç kez değiştirilmiş ve optimize edilmiştir, fakat temel ilkeler aynı kalmıştır [17], [18], [19].

NTRU'nun icadı yaygın olarak kullanılan asimetrik kriptosistem için umut verici bir alternatif olmuştur. Hız, anahtar üretimi ve sistem gereksinimleri konularında avantajları vardır.

3.1 Parametreler

NTRU kriptosistemi tarafından kullanılan temel elemanlar, n tamsayı olmak üzere $\mathcal{R} = {\mathbb{Z}[x]}/{\langle x^n - 1\rangle}$ halkasındaki polinomlardır. Bu polinomlar, şifreleme ve deşifreleme işlemleri sırasında aralarında asal iki tamsayıya indirgenir. Bu tamsayı parametreleri p ve q ile gösterilir ve asal olmalarına gerek yoktur. Ancak q, p'den her zaman oldukça büyük olmalıdır.

NTRU kriptosistemindeki hesaplamalar $\mathcal{R}$ halkasında yapılır. Bir $a \in \mathcal{R}$, bir polinom veya bir vektör olarak gösterilebilir.

$$a = \sum_{i=0}^{n-1} a_i x^i = [a_0 , a_1 , \dots , a_{n-1}]$$

$\mathcal{R}$ halkasındaki çarpma işlemi

$$a = a_0 + a_1 x + a_2 x^2 + \ \dots + a_{n-1} x^{n-1} \ \in \mathcal{R}$$

$b = b_0 + b_1x + b_2x^2 + \ \dots + b_{n-1}x^{n-1} \ \in \mathcal{R}$

ve $\forall\, k = 0,1, \ \dots, n-1$ için

$$c_k = \sum_{i=0}^{k} a_i b_{k-i} + \sum_{j=k+1}^{n-1} a_j b_{n-j}$$

olmak üzere

$a * b = c = c_0 + c_1x \ + c_2x^2 + \ \dots \ + \ c_{n-1}x^{n-1}$

şeklinde tanımlanır ve devirli evrişim çarpımı (cyclic convolution product) olarak adlandırılır.

Şifreleme ve deşifreleme sırasında bu çarpma işlemi yapılırken katsayılar $modulo\ q$'ya indirgenir. $Modulo\ q$'da işlem yapılan halka $\mathcal{R}_q = {\mathbb{Z}_q[x]}/{\langle x^n - 1\rangle}$ ile gösterilir. n. dereceden iki polinomun çarpımının hesaplanması n^2 tane çarpım gerektirir. Bu yüzden NTRU, iki polinomun çarpımını hızlıca hesaplamak için küçük katsayılı polinomlar kullanır.

Şifrelemede kullanılan bir diğer işlem polinomların tersini almaktır. $\mathcal{R}_q$ halkasındaki bir a polinomunun tersi $a * a^{-1} = a^{-1} * a = 1$ olacak şekilde $a^{-1} \in \mathcal{R}_q$ polinomudur. Ancak $\mathcal{R}_q$ halkasında her polinomun tersi yoktur. $f(1) = 1$ eşitliğini sağlayan rastgele seçilmiş bir $f \in \mathcal{R}_q$ polinomunun tersinin yüksek olasılıkla var olduğu kanıtlanmıştır [20]. Bir polinomun tersini hesaplamak için [21], [22] de hızlı bir algoritma tanımlanmıştır.

3.2 Bir Polinomun Ortalanması ve Küçük Polinomlar

NTRU'nun şifreleme ve deşifreleme işlemleri sırasında önemli bir nokta, mod alma işleminin tanımıdır. Bu işlemler sırasında oluşabilecek başarısızlıklardan kaçınmak için, polinomların ortalanması gerekmektedir. Yani $f\ (mod\ m)$ işleminde, f polinomunun katsayılarının $[0, m)$ aralığında indirgenmesi yerine, $(\ \lfloor -m/2 \rfloor, \lfloor m/2 \rfloor\ ]$ aralığında indirgenmesi gerekmektedir.

Anahtar üretimi, şifreleme ve deşifreleme işlemleri sırasında $\mathcal{R}_q$ halkasından rastgele f, g ve r polinomları seçilir. Ayrıca gönderilmek istenen mesaj $\mathcal{R}_q$ halkasında bir polinom türünde yazılır. İşlemleri hızlandırmak amacıyla, NTRU tasarımcıları ilgili polinomlar için L_f, L_g, L_m ve L_r ile gösterilen bazı uzaylar tanımlamıştır [7]. İlk olarak, katsayıları $modulo\ p$'den olan polinomların oluşturduğu mesajlar uzayı

$$L_m = \{m \in \mathcal{R} \mid m\ polinomunun\ katsayıları\ (\ \lfloor -p/2 \rfloor, \lfloor p/2 \rfloor\ ]\ aralığındadir.\}$$

olarak tanımlanır. $p = 3$ olduğu durumda, diğer uzayları tanımlamak için aşağıdaki gösterim kullanılır.

$$L(d_1, d_2) = \left\{ f \in \mathcal{R} \;\middle|\; \begin{array}{r} f\ polinomunun\ d_1\ tane\ katsayısı\ 1 \\ d_2\ tane\ katsayısı\ -1 \\ geri\ kalan\ katsayıları\ 0\ dır. \end{array} \right\}$$

Bu notasyon ile f, g ve r polinomlarının uzayları sırasıyla d_f, d_g ve d_r tamsayıları seçilerek

$$L_f = L(d_f,\ d_f - 1)$$

$$L_g = L(d_g\ , d_g)$$

$$L_r = L(d_r\ , d_r)$$

şeklinde tanımlanır. Bu uzaylardan seçilen polinomlara "küçük polinomlar" denir.

f polinomunun 1 ve -1 olan katsayılarının sayısı aynı değildir. Çünkü $f(1) = 0$ eşitliğini sağlayan bir polinom tersinir olamaz [7].

3.3 Anahtar Üretimi

$\mathcal{R}$ halkasının bir alt kümesi olan L_f kümesinden rastgele küçük bir f polinomu seçilir. f polinomunun $mod\ q$ ve $mod\ p$'de tersi olmalıdır ve sırasıyla f_q^{-1} ve f_p^{-1} ile gösterilir. Daha önce belirtildiği gibi f yüksek olasılıkla tersinirdir ve terslerinin hesaplanması kolaydır. Eğer f polinomunun tersi yoksa yeni bir polinom seçilir ve tersinin olup olmadığı tekrar araştırılır.

Daha sonra $\mathcal{R}$ halkasının bir alt kümesi olan L_g kümesinden rastgele küçük bir g polinomu seçilir ve $h \equiv p f_q^{-1} * g\ \ (mod\ q)$ çarpımı hesaplanır. Böylece f_q^{-1}, f_p^{-1} ve g

polinomları gizli tutularak, h açık anahtarı ve f gizli anahtarı elde edilir. 3. ve 4. bölümlerde verilen örnekler [23] de belirtilen kaynaktan alınmıştır.

Örnek 3.1 Parametreleri $n = 11,\ q = 32,\ p = 3,\ d_f = 4,\ d_g = 3$ olan bir NTRU kriptosistemi için bir anahtar çifti üretiniz.

Burada f polinomu 10. dereceden olmalı ve dört tane katsayısı 1, üç tane katsayısı -1 ve diğer katsayıları 0 olmalıdır. Aynı şekilde g, 10.dereceden ve üç tane katsayısı 1, üç tane katsayısı -1 ve diğer katsayıları 0 olan bir polinom olmalıdır.

$f = -1 + x + x^2 - x^4 + x^6 + x^9 - x^{10}$

$g = -1 + x^2 + x^3 + x^5 - x^8 - x^{10}$

olsun. Daha sonra f polinomunun $mod\ p$ ve $mod\ q$'da tersi hesaplanır.

$f_p^{-1} = 1 + 2x + 2x^3 + 2x^4 + x^5 + 2x^7 + x^8 + x^9$

$f_q^{-1} = 5 + 9x + 6x^2 + 16x^3 + 4x^4 + 15x^5 + 16x^6 + 22x^7 + 20x^8 + 18x^9 + 30x^{10}$

Anahtar üretimindeki son adım aşağıdaki çarpma işlemini hesaplamaktır.

$h \equiv p f_q^{-1} * g \ \ (mod\ q)$

$h \equiv 3(5 + 9x + 6x^2 + 16x^3 + 4x^4 + 15x^5 + 16x^6 + 22x^7 + 20x^8 + 18x^9 + 30x^{10})$
$* (-1 + x^2 + x^3 + x^5 - x^8 - x^{10})\ (mod\ 32)$

$h \equiv 8 + 25x + 22x^2 + 20x^3 + 12x^4 + 24x^5 + 15x^6 + 19x^7 + 12x^8 + 19x^9 + 16x^{10}$

Sonuç olarak gizli anahtar $f = -1 + x + x^2 - x^4 + x^6 + x^9 - x^{10}$ ve açık anahtar $h = 8 + 25x + 22x^2 + 20x^3 + 12x^4 + 24x^5 + 15x^6 + 19x^7 + 12x^8 + 19x^9 + 16x^{10}$ dir.

Yukarıdaki gibi gerçekleştirilen anahtar üretme işlemi bize iki adet kafes tabanı elde etme fırsatı verir. Birinci taban, yani açık taban, $2n$ boyutlu

$$\boldsymbol{\mathcal{L}}_{açık} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{M_h} \\ \boldsymbol{0} & q\boldsymbol{I} \end{bmatrix}$$

matrisinin satır vektörlerinin oluşturduğu kümedir. Burada $\boldsymbol{M_h}$, h polinomunun katsayı vektörüne karşılık gelen dairesel matristir. $[\boldsymbol{f} \quad \boldsymbol{g}]$ vektörü ise $2n$ boyutlu kafese ait bir kısa vektördür. Bu taban kötü bir tabandır, yani bu tabanı kullanarak verilen bir hedef

vektöre yakın bir kafes vektörü bulmak ya da kafese ait kısa bir vektör elde etmek zor problemdir. Diğer taraftan gizli anahtarların bilgisi bize $3n$ boyutlu bir kafese ait tabanı verir. Bu tabana ait vektörlerin satırlarının oluşturduğu matris

$$\boldsymbol{\mathcal{L}}_{gizli} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{M_f} & \boldsymbol{I} \\ \boldsymbol{0} & q\boldsymbol{I} & q\boldsymbol{M_{f_p^{-1}}} \\ \boldsymbol{0} & \boldsymbol{0} & p\boldsymbol{I} \end{bmatrix}$$

matrisidir. Burada n boyutlu $\boldsymbol{M_f}$ ve $\boldsymbol{M_{f_p^{-1}}}$ matrisleri sırasıyla f ve f_p^{-1} polinomlarının katsayı vektörlerine karşılık gelen dairesel matrislerdir. Tabanın en önemli özelliği verilen bazı tip hedef vektörlere en yakın kafes vektörlerinin bulunmasının kolay olması, dolayısıyla da iyi bir taban olmasıdır.

3.4 Şifreleme

$\mathcal{R}$ halkasının bir alt kümesi olan L_m düz metin kümesinden bir m mesajı seçilir. Sonra $\mathcal{R}$'nin bir altkümesi olan L_r kümesinden rastgele küçük bir r polinomu seçilir. Bu polinoma mesajı gizlemek için kullanılan "körleştirme değeri (blinding-value)" adı verilir.

$e \equiv r * h + m \ (mod\ q)$ hesaplanarak e şifreli metin elde edilir.

Şifreleme işleminde rastgele bir polinom kullanıldığından m mesajı çok farklı şekillerde şifrelenebilir.

Örnek 3.2 Parametreleri $n = 11$, $q = 32$, $p = 3$, $d_f = 4$, $d_g = 3$, $d_r = 3$ olan bir NTRU kriptosistemi üzerinde, açık anahtarı

$h = 8 + 25x + 22x^2 + 20x^3 + 12x^4 + 24x^5 + 15x^6 + 19x^7 + 12x^8 + 19x^9 + 16x^{10}$

olan bir kişiye $m = -1 + x^3 - x^4 - x^8 + x^9 + x^{10}$ mesajını göndermek için şifreleyiniz.

$r = -1 + x^2 + x^3 + x^4 - x^5 - x^7$ olsun.

$e \equiv r * h + m \ (mod\ q)$

$$e \equiv (-1 + x^2 + x^3 + x^4 - x^5 - x^7) * (8 + 25x + 22x^2 + 20x^3 + 12x^4 + 24x^5 + 15x^6 + 19x^7 + 12x^8 + 19x^9 + 16x^{10}) + (-1 + x^3 - x^4 - x^8 + x^9 + x^{10}) \ (mod 32)$$

$e = 14 + 11x + 26x^2 + 24x^3 + 14x^4 + 16x^5 + 30x^6 + 7x^7 + 25x^8 + 6x^9 + 19x^{10}$

şifreli metin elde edilir.

Şifreleme işlemi

$$[\mathbf{0} \quad \boldsymbol{e}] = [\boldsymbol{r} \quad \boldsymbol{\psi}] \begin{bmatrix} \boldsymbol{I} & \boldsymbol{M_h} \\ \mathbf{0} & q\boldsymbol{I} \end{bmatrix} + [-\boldsymbol{r} \quad \boldsymbol{m}]$$

şeklinde de gösterilebilmektedir. Burada $\boldsymbol{\psi}$ vektörü, e polinomunun katsayılarını $mod\ q$'da istenilen aralıkta tutacak şekilde seçilir.

3.5 Deşifreleme

f gizli anahtarı ve f_p^{-1} kullanılarak şifreli metin şu şekilde deşifre edilir:

$$a \equiv f * e\ (mod\ q) \tag{3.1}$$

$$a' \equiv a\ (mod\ p) \tag{3.2}$$

ve son olarak

$$c \equiv f_p^{-1} * a'\ (mod\ p) \tag{3.3}$$

hesaplanır. Uygun parametre değerleri için, c polinomu yüksek olasılıkla m orijinal mesaja eşit olacaktır. Fakat bazı parametre değerleri başarısızlığa sebep olabilmektedir (c polinomu m mesajına eşit olmayabilir). Bu yüzden uygulamada her mesaj bloğuna birkaç kontrol biti eklenmelidir. Deşifrelemede başarısızlığın sebebi polinomların doğru şekilde ortalanmamış olmasıdır.

Eğer (3.1) işlemi sırasında a polinomu doğru şekilde ortalanırsa, yani uygun seçilmiş pozitif veya negatif küçük bir x değeri için katsayılar $(\ x + \lfloor -q/2 \rfloor, x + \lfloor q/2 \rfloor\ )$ aralığına indirgenirse, şifreli metin çok büyük olasılıkla deşifrelenebilir. Eğer bu işlem hiçbir x değeri için başarıyla gerçekleştirilemezse "aralık hatası (gap failure)" oluştu denir. Uygulamada göz ardı edilebilecek kadar az karşılaşılan bu durumlarda metin kolay bir şekilde deşifrelenemez [7], [24].

Örnek 3.3 Bir önceki örnekte verilen parametreler ve oluşturulan anahtar çifti için $e = 14 + 11x + 26x^2 + 24x^3 + 14x^4 + 16x^5 + 30x^6 + 7x^7 + 25x^8 + 6x^9 + 19x^{10}$ şifreli metnini deşifre ediniz.

İlk olarak a polinomu hesaplanır ve katsayıları $(\ \lfloor -q/2 \rfloor, \lfloor q/2 \rfloor\ )$ aralığında indirgenir.

$$a \equiv f * e \ (mod\ q)$$

$$a \equiv (-1 + x + x^2 - x^4 + x^6 + x^9 - x^{10}) * (14 + 11x + 26x^2 + 24x^3 + 14x^4 + 16x^5 + 30x^6 + 7x^7 + 25x^8 + 6x^9 + 19x^{10}) \ (mod\ 32)$$

$$a \equiv 3 - 7x - 10x^2 - 11x^3 + 10x^4 + 7x^5 + 6x^6 + 7x^7 + 5x^8 - 3x^9 - 7x^{10} (mod\ 32)$$

$$a' \equiv -x - x^2 + x^3 + x^4 + x^5 + x^7 - x^8 - x^{10} \ \ (mod\ 3)$$

bulunur ve son olarak $c \equiv f_p^{-1} * a' \ (mod\ p)$ denkliği hesaplanarak

$$c \equiv (1 + 2x + 2x^3 + 2x^4 + x^5 + 2x^7 + x^8 + x^9) * (-x - x^2 + x^3 + x^4 + x^5 + x^7 - x^8 - x^{10}) \ (mod\ 3)$$

$$c \equiv -1 + x^3 - x^4 - x^8 + x^9 + x^{10} \ \ (mod\ 3)$$

m mesajına eşit c polinomu elde edilir.

Deşifrelemeyi gerçekleştirme işlemi için aslında yapılması gereken $[\boldsymbol{e} \quad \mathbf{0} \quad \mathbf{0}]$ vektörüne en yakın kafes vektörünü bulmaktır. Bu da

$$[\boldsymbol{e} \quad \boldsymbol{a} \quad \boldsymbol{b}] = [\boldsymbol{e} \quad \boldsymbol{\lambda} \quad \boldsymbol{\mu}] \begin{bmatrix} \boldsymbol{I} & \boldsymbol{M_f} & \boldsymbol{I} \\ \mathbf{0} & q\boldsymbol{I} & q\boldsymbol{M_{f_p^{-1}}} \\ \mathbf{0} & \mathbf{0} & p\boldsymbol{I} \end{bmatrix}$$

işlemi sonunda bulunabilir. Burada $\boldsymbol{\lambda}$ vektörü,

$$a = f * e + q\lambda$$

değeri mümkün olduğunca küçük olacak şekilde seçilen, λ polinomunun katsayı vektörüdür. Bu şekilde seçilmiş $\boldsymbol{\lambda}$ vektörü için

$$a = f * e + q\lambda = pr * g + f * m$$

eşitliği sağlanacaktır. Sonrasında eğer $\boldsymbol{\mu}$ vektörü,

$$b = e + qf_p^{-1} * \lambda + p\mu$$

eşitliğini minimize edecek şekilde seçilirse

$$b = m$$

olduğu görülecektir. Böylece şifreli metin deşifrelenmiş olur.

Eğer yukarıdaki hesaplamalara ayrıntılı bakılırsa

$$a \equiv f * e \pmod{q}$$

$$\equiv f * [r * h + m] \pmod{q}$$

$$\equiv f * \left[r * \left(p f_q^{-1} * g\right) + m\right] \pmod{q}$$

$$\equiv pr * g + f * m \pmod{q}$$

olduğu görülür. Şifrelemede uygun polinomlar seçilmişse, $pr * g + f * m$ polinomunun katsayıları q'dan daha küçük olur. Böylece katsayılar $mod\ q$'ya indirgendiği zaman değişmeyecektir. Bu yüzden a polinomu $pr * g + f * m$ polinomuna tam olarak eşit olacaktır. a'nın katsayıları $mod\ p$'ye indirgendiği zaman $a \equiv pr * g + f * m \equiv f * m \pmod{p}$ elde edilir. Son olarak $mod\ p$'de f_p^{-1} ile $f * m$ çarpılarak $m\ (mod\ p)$ mesajı elde edilir.

Daha önce tanımlanan uzaylardan polinomlar seçildiği zaman, a polinomunun katsayıları $(\lfloor -q/2 \rfloor, \lfloor q/2 \rfloor]$ aralığında olacak şekilde yeterine küçük olacaktır. Yani a polinomu $pr * g + f * m$ polinomuna tam olarak eşit olacaktır. Bu, deşifrelemedeki başarısızlıkları önleyecektir.

3.6 Farklı Güvenlik Seviyeleri İçin Önerilen Parametreler

NTRU kriptosistemi üç tamsayı parametresi n, p, q ve d_f, d_g, d_r, p tamsayılarına bağlı olarak L_f, L_g, L_r, L_m kümelerine sahiptir. Eğer uygun parametreler seçilirse deşifrelemede başarısızlık olasılığı 10^{-5} den daha azdır [25]. Farklı güvenlik seviyeleri için önerilen parametreler aşağıdaki çizelgededir [17].

Çizelge 3.1 Farklı güvenlik seviyeleri için önerilen parametreler

	n	q	p	d_f	d_g	d_r
Orta	167	128	3	61	20	18
Standart	251	128	3	50	24	16
Yüksek	503	256	3	216	72	55

BÖLÜM 4

GELİŞTİRİLMİŞ NTRU

Bu bölümde NTRU işlemlerini hızlandırmak için kullanılabilecek bazı özel teknikler açıklanmıştır.

4.1 f Polinomunun Seçimi

Bölüm 3'teki gibi, f aşağıdaki özelliklere sahip olmak zorundadır.

- $f \mod p$'de tersinir
- $f \mod q$'da tersinir
- f küçüktür

Önceki örneklerde, d_f parametresi kullanılarak f polinomunun küçük olması sağlanmıştır. Ayrıca f, $\mod p$ ve $\mod q$'da tersinir olarak seçilmiştir.

Ticari uygulamalarda, f'nin seçiminde alternatif bir yöntem kullanılır. F küçük bir polinom olmak üzere $f = 1 + pF$ şeklinde alınır. Bu seçim, f'nin $\mod p$'de 1'e eşit olduğu anlamına gelmektedir. Bu ise aşağıdaki avantajları sağlar.

- f her zaman $\mod p$'de tersinirdir (Gerçekten $f^{-1} \equiv 1 \pmod p$'dir). Bu anahtar üretimini hızlandırır. Çünkü f'nin $\mod p$'de tersi hesaplanmak zorunda değildir.
- $f^{-1} \equiv 1 \pmod p$ olduğundan deşifreleme yapılırken artık ikinci bir polinom çarpma işleminin yapılmasına gerek kalmaz. Sadece bir çarpma işlemi gerektirdiği için deşifreleme önemli ölçüde hızlanır. Ayrıca $f_p \equiv f^{-1} \pmod p$ gizli anahtarın bir parçası olarak saklanmak zorunda değildir.

4.2 p'nin Seçimi

Şimdiye kadar küçük polinomların hepsi (p her zaman küçük seçildiğinden) küçük katsayılara sahip polinomlar olarak tanımlanmıştır. Deşifrelemenin başarısı a'nın katsayılarının $mod\ q$'ya indirgendiği zaman değişmemiş olmasına bağlıdır. f, g, m ve r'nin katsayıları daha küçük olursa, a'nın katsayıları da genellikle daha küçük olacaktır. Bu durumda eğer p'nin boyutu azaltılabilirse, (deşifreleme işlemi başarılı olacak şekilde) daha kolay parametre kümesi seçilmiş olur.

p'nin bir tamsayı olmasını gerektirecek bir şey yoktur. Tek şart $\mathcal{R}$ halkasında p ve q'nun aralarında asal olmasıdır. Bu durumda p, $2 + x$ şeklinde küçük bir polinom olarak alınabilir.

p bu formda seçildiği zaman, üçlü (trinary, katsayıları -1, 0, +1 olan) polinomlar yerine ikili (binary, katsayıları 0 ve 1 olan) polinomlar kullanmak mesajın daha kolay şifrelenmesini sağlar. Öte yandan, deşifrelemede a polinomunu $mod\ p$ ye indirgemek biraz daha karmaşık hale gelir.

Bir a polinomunu $mod\ 2 + x$'e indirgeme işlemi,

$$d(-2) \equiv a(-2)\ mod\ 2^n + 1$$

denkliğini sağlayan ve katsayıları $mod\ q$'dan olan n. dereceden bir d polinomu bulma işlemidir [18].

4.3 a Polinomunun Ortalanması

Bölüm 3'teki deşifreleme işlemlerini hatırlayalım.

$$a \equiv f * e\ (mod\ q)$$

$$\equiv f * (r * h + m)\ (mod\ q)$$

$$\equiv f * \left(r * p * f_q * g + m\right)\ (mod\ q)$$

$$\equiv r * p * g + f * m\ (mod\ q)$$

r, g ve m polinomları 0'da ortalanmıştır (yani 1'ler ve -1'ler eşit sayıda) ve f polinomu neredeyse 0'da ortalanmıştır (yani f, -1 den daha fazla +1 katsayısına sahip). Bu durumda

$r(1) = g(1) = m(1) = 0$

$f(1) = 1$

dir.

$r * p * g + f * m$ polinomun katsayıları $(-q/2\,, q/2]$ aralığında olduğundan deşifreleme çalışır. Fakat $f = 1 + p * F$ formunda alındığı zaman, yukarıdaki gibi r, g, m ve f değerlerini sıfırda ortalamak hatalı deşifrelemeye neden olabilir. Bu yüzden deşifreleme yapılırken $mod\, p$'ye indirgeme işlemini yapılmadan önce, $r * p * g + f * m$ polinomunun katsayılarının doğru aralıkta olup olmadığına bakılmalıdır. Bunun için aşağıdaki metot kullanılır.

I $= f_q(1) \cdot \big(a(1) - p(1) \cdot r(1) \cdot g(1)\big)\, mod q$

$$Avg = \frac{p(1) \cdot r(1) \cdot g(1) + \mathrm{I}\, f(1)}{n}$$

Beklenen katsayı aralığı $Avg - q/2$ ve $Avg + q/2$ arasında olacaktır. Avg genellikle bir tamsayı olmayacaktır, bu yüzden gerçek beklenen değer $Avg - q/2$ ve $Avg + q/2$ arasındaki q tamsayıları olacaktır [18].

4.4 Örnekler

Örnek 4.1 Çizelge 4.1'de verilen parametreler ile belirli bir NTRU kriptosistemi için anahtar üretimi, şifreleme ve deşifreleme işlemleri şu şekilde gerçekleştirilir:

Çizelge 4.1 Bir NTRU kriptosistemi için parametreler kümesi

n	q	p	d_F	d_g	d_r	f
11	32	$2 + x$	4	5	4	$1 + p * F$

Anahtar üretimi:

$F = 1 + x^4 + x^7 + x^9$ olsun. $f = 1 + (2 + x) * F$ den f gizli polinomu hesaplanır.

$f = 1 + (2 + x) * 1 + x^4 + x^7 + x^9 = 3 + x + 2x^4 + x^5 + 2x^7 + x^8 + 2x^9 + x^{10}$

bulunur. Açık anahtarı hesaplamak için ilk başta f'nin $mod\, q$'da tersi hesaplanır.

$f_q = -7 - 5x + 12x^2 - 3x^3 - 2x^4 + 6x^5 + 13x^6 - 10x^7 - 8x^8 - 8x^9 - 15x^{10}$

bulunur. Sonra, rastgele küçük bir g polinom üretilir. $g = 1 + x + x^4 + x^6 + x^{10}$ olsun. Son olarak, h açık anahtarı şu şekilde hesaplanır:

$h = p * f_q * g$

$h = 15 + 11x + 9x^2 - 14x^3 - 12x^4 + 12x^5 - 7x^6 - 12x^7 - 13x^8 - 8x^9 - 2x^{10}$

Şifreleme:

m mesajı, rastgele seçilmiş küçük bir r polinomu ve yukarıda hesaplanan h açık anahtarı kullanılarak $e \equiv r * h + m\ (mod q)$ şeklinde şifrelenir. Şifrelenmek istenen ikili mesaj 01010100111 olsun. Bu ikili mesaj, polinoma çevrilir.

$m = x + x^3 + x^5 + x^8 + x^9 + x^{10}$

Sonra küçük bir r polinomu üretilir. $d_r = 4$ için $r = 1 + x^3 + x^4 + x^8$ olsun.

$e \equiv r * h + m\ (mod q)$

$e \equiv 8 + 11x + 11x^2 - 7x^3 + 2x^4 - 12x^5 + 12x^6 - 8x^7 + 3x^8 + 9x^9 - 11x^{10}$

şifrelenmiş mesaj elde edilir.

Deşifreleme:

e mesajını deşifrelemek için, f gizli anahtarı kullanılarak $a \equiv f * e\ mod q$ hesaplanır.

$a \equiv f * e\ (mod q)$

$a \equiv 7 + 14x + 10x^2 + 15x^3 + 14x^4 + 13x^5 + 10x^6 + 11x^7 + 15x^8 + 14x^9 + 15x^{10}$

Normalde, a polinomun ortalanma değeri hesaplanmalıdır, fakat burada bu adım atlanır. Katsayılar $[7, 15]$ aralığında sıkı şekilde kümelenmiş olduğundan ortalamak gerekmez. Sonraki örnekte deşifrelenmiş mesajı doğru şekilde elde etmek için ortalanma değerinin hesaplanması gereken bir durum gösterilmiştir.

Sonra $p = 2 + x$ olmak üzere a polinomu $mod\ p$'ye indirgenir. Yani, aşağıdaki özelliklere sahip bir d polinomu bulunur.

$d(-2) \equiv a(-2)\ mod\ 2^n + 1$

Böylece $d = x + x^3 + x^5 + x^8 + x^9 + x^{10}$ polinomu elde edilir. Bu polinom aşağıdaki gibi kolayca kontrol edilebilir:

$a(-2) = 10971$

$2^n + 1 = 2049$

$a(-2)\ mod(2^n + 1) \equiv 726$

$d(-2) = 726$

Son olarak d polinomu ikili mesaj 01010100111 e çevrilir. Bu, m mesajına eşittir. Böylece deşifreleme işlemi başarıyla gerçekleştirildi.

Örnek 4.2 Bu örnekte, ortalanma değerinin seçiminin, deşifrelemenin başarısı için bir farklılık oluşturduğu gösterilmiştir. Çizelge 4.1'de belirtilen parametre değerleri kullanılmıştır.

Anahtar üretimi:

$d_F = 4$ ile belirli F polinomu $F = 1 + x^4 + x^6 + x^7$ olsun. $f = 1 + (2 + x) * F$ den f gizli polinomu hesaplanır:

$f = 3 + x + 2x^4 + x^5 + 2x^6 + 3x^7 + x^8$

Açık anahtarı hesaplamak için ilk başta f'nin $mod\ q$'da tersi hesaplanır.

$f_q = 14 + 4x - x^2 - 5x^3 + 10x^4 + 9x^5 + 6x^6 + 13x^7 + 4x^8 + 3x^9 + 12x^{10}$

$d_g = 5$ ile belirli rastgele küçük bir g polinom üretilir. $g = x + x^2 + x^4 + x^6 + x^7$ olsun. h açık anahtarı şu şekilde hesaplanır:

$h = p * f_q * g$

$h = 16 + 9x - 3x^2 + 8x^3 - 2x^4 - x^5 + 16x^6 + 4x^7 - 4x^8 + 8x^9 - 8x^{10}$

f gizli anahtar olarak saklanır ve h açık anahtar olarak kamuya açık hale getirilir.

Şifreleme:

m mesajını şifrelemek için $d_r = 4$ değeri kullanılarak rastgele küçük bir r polinomu üretilir ve $r * h + m\ (modq)$ hesaplanır. $m = x + x^2 + x^6 + x^7 + x^8 + x^{10}$ ve $r = x + x^2 + x^4 + x^5$ olsun.

$e \equiv r * h + m\ (mod q)$

$e \equiv -12 + 9x - 2x^2 + 6x^3 + 13x^4 - x^5 + 4x^6 - 11x^7 - 5x^8 - 3x^9 - 12x^{10}$

e şifrelenmiş mesaj elde edilir.

Deşifreleme:

e mesajı alınarak $a \equiv f * e\ (mod\ q)$ hesaplanır ve sonra d'yi elde etmek için sonuç $mod\ p$'ye indirgenir.

$$a = f * e\ = -23 + 12x - 19x^2 - 50x^3 - 23x^4 - 22x^5 - 47x^6 - 49x^7 + 17x^8 + 12x^9 + 10x^{10}$$

$$a \equiv 9 + 12x + 13x^2 + 14x^3 + 9x^4 + 10x^5 - 15x^6 + 15x^7 - 15x^8 + 12x^9 + 10x^{10}\ (mod 32)$$

Fakat bu polinom $mod\ p$'ye indirgendiği zaman gerçek mesaj yerine $m' = x + x^2 + x^3 + x^6 + x^8 + x^9$

elde edilir. Bunun nedeni a kısmi deşifrelenmiş mesajın yanlış ortalanmasıdır. I'yı hesaplamak için aşağıda verilen metot kullanılır.

$$I = f_q(1)\big(a(1) - p(1)r(1)g(1)\big) = 69(74 - 3 \cdot 4 \cdot 5) = 69 \cdot 14 \equiv 6\ (mod 32)$$

$$Avg = \frac{p(1) \cdot r(1) \cdot g(1) + \mathrm{I} \cdot \mathrm{f}(1)}{n} = \frac{(3 \cdot 4 \cdot 5 + 6 \cdot 13)}{11} = \frac{60 + 78}{11} = 12{,}5454$$

a'nın katsayıları $(Avg - 16,\ Avg + 16) = (-3,\ 28)$ aralığında değiştirilir. Bu durumda

$$a\ (mod\ q) \equiv 9 + 12x + 13x^2 + 14x^3 + 9x^4 + 10x^5 + 17x^6 + 15x^7 + 17x^8 + 12x^9 + 10x^{10}$$

olur. Bulunan a polinomu $mod\ p$'ye indirgenerek doğru mesaj

$m = x + x^2 + x^6 + x^7 + x^8 + x^{10}$

elde edilir.

4.5 Küçük Hamming Ağırlıklı Polinomlar

Hesaplamalarda kullanılan polinomların çoğu küçük polinomlardır. NTRU kriptosistemi, bu küçük polinomların oluşturulmasıyla ilgili birçok esneklik sağlar. Bu bölümün başında sistemin merkezi işlemlerini etkilemeden f gizli anahtarının değiştirilmesi ile ilgili bir yöntem verilmiştir. Burada ise küçük polinomları seçmek için önemli performans avantajları sağlayan başka bir yöntem tanımlanmıştır.

İlk olarak, 3. dereceden polinomlar için evrişim çarpımına küçük bir örnek verelim.

$$[4,5,7] * [5,3,2] = 4 * [5,3,2] + 5 * [2,5,3] + 7 * [3,2,5]$$
$$= [20,12,8] + [10,25,15] + [21,14,35]$$
$$= [20 + 10 + 21, 12 + 25 + 14, 8 + 15 + 35]$$
$$= [51,51,58]$$

Bu örnekte iki kübik polinomun çarpımı 9 çarpma ve 6 toplama gerektirir. Şimdi polinomlardan birinin sadece 1 ve 0'larda oluştuğu duruma bakalım.

$$[1,0,0,1,0] * [5,3,2,9,10]$$
$$= 1 * [5,3,2,9,10] + 0 * [10,5,3,2,9] + 0 * [9,10,5,3,2]$$
$$+1 * [2,9,10,5,3] + 0 * [3,2,9,10,5]$$
$$= [5 + 2\ , 3 + 9\ , 2 + 10\ , 9 + 5\ , 10 + 3]$$
$$= [7,12,12,14,13]$$

Sonuç olarak, bir polinomun sadece 0 ve 1 katsayılarından oluşması işlemleri önemli ölçüde hızlandırır.

4.5.1 Küçük Hamming Ağırlıklı Polinomların Üretilmesi

Başarılı deşifreleme mesajın, gizli anahtarın ve körleştirme değerinin küçük olmasına bağlıdır. Ancak çok küçük polinomlarda güvenlik riski mevcuttur. Açık anahtar $h = f_q * g$'yi bilen ve gizli anahtar f'yi bulmaya çalışan bir saldırgan düşünelim.

Eğer f çok az sayıda sıfırdan farklı katsayıya sahipse, saldırganın sırasıyla tüm olası f polinomlarını denemesi mümkündür. Eğer saldırgan $f * h\ (mod q)$ küçük olacak şekilde

f'yi bulursa, mesajı deşifreleyebilir. Bu yüzden bu "kaba kuvvet (brute force)" saldırılarını önlemek için, küçük polinomların arama alanı (search space) yeterince büyük olmalıdır.

h iki küçük elemandan oluşturulduğundan, f ve g'yi elde etmek için h üzerinde "kısa kafes vektör" saldırısı uygulamak mümkündür. Bu kısa kafes vektörü saldırısı, NTRU'ya saldırı için bilinen en iyi yoldur. Bu saldırının temeli olan kafes indirgeme algoritmalarının çalışma süresi şunlara bağlıdır:

- Kafesin boyutu: Daha büyük boyutlu kafes, saldırının daha uzun sürmesine neden olur.
- En kısa vektör: En kısa vektör daha kısa olursa, onu bulmak daha kolay olur.

Eğer f çok az sayıda 1 içeriyorsa, kafes saldırıları ile bulunması çok kolaydır. Bu yüzden f'nin (ve sistem içindeki diğer tüm küçük polinomların) bu saldırıyı önlemek için yeterince uzun olması gerekmektedir.

Sonuç olarak, NTRU'nun gizli anahtarı küçük polinomlar olmalıdır fakat çok küçük olmamalıdır.

Hiçbir güvenlik açığıyla karşılaşmadan az sayıda 1'e sahip olmasından fayda sağlanılan bir yol vardır. f tek bir küçük polinom yerine, birkaç tane küçük polinomun birleşimi olarak alınabilir [26].

Örneğin, $n = 13$ ve küçük polinomların 9 tane sıfırdan farklı katsayısı olsun (tam-boyutlu kriptosistem versiyonlarında, $n = 251$, genellikle küçük polinomlar, katsayıların yaklaşık üçte biri sıfırdan farklı olacak şekilde alınır).

$$i = i_1 * i_2 = [1,0,1,0,0,0,0,1,0,0,0,0,0] * [1,0,0,1,1,0,0,0,0,0,0,0,0]$$
$$= [1,0,1,1,1,1,1,1,0,0,1,1,0]$$

i polinomunun 9 tane katsayısı 1'dir. Bu yüzden $i * a$'yı hesaplarken her katsayı için 9 toplama yapılır. Ama eğer bunun yerine i_1 ve i_2 gibi iki bileşen kullanılırsa, direkt olarak $i * a$'nın hesaplanmasına gerek kalmaz. Bunun yerine ilk başta her katsayı için 3 toplama yapılarak $i_2 * a$ hesaplanır ve sonra diğer 3 toplama yapılarak $i_1 * (i_2 * a)$

hesaplanır. Böylece 9 toplama yerine, her katsayı için toplamda 6 toplama yapılarak $i * a$ hesaplanır.

NTRU'nun ticari uygulamaları için $n = 251$ dir. Saldırılardan kaçınmak için, küçük polinomlar yaklaşık 72 sıfırdan farklı katsayıya polinomlar olarak seçilir. Bu durumda işlemler şu şekilde hızlandırılabilir:

$d_F = 72$ olmak üzere $f = 1 + p * F$ yerine, $d_{f_1} = 8,\ d_{f_2} = 8,\ d_{f_3} = 8$ olacak şekilde $f = 1 + p * ((f_1 * f_2) * f_3)$ alınır. Bu durumda F ile çarpmada her katsayı için (72 değil) 24 toplama yapılır.

$d_r = 72$ olacak şekilde r küçük polinomu yerine, $d_{r_1} = 8, d_{r_2} = 8, d_{r_3} = 8$ olacak şekilde $r = (r_1 * r_2) + r_3$ alınır. Böylece r ile çarpmada her katsayı için (72 değil) 24 toplama yapılır.

g sadece anahtar üretimi boyunca kullanıldığından üzerine bir değişiklik yapılmaz.

Küçük polinomları oluşturan bu metot, kafes saldırıları için extra güvenlik açığı ortaya çıkarmaz [26]. Fakat kaba kuvvet saldırısının hala pratik olmadığından emin olunmalıdır. Eğer $n = 251$ ve her f_i'nin 8 tane katsayısı 1 ise olası f polinomlarının sayısı 251^3'tür. Bu sayı, kaba kuvvet saldırısının zorluğunda bir üst sınırdır. Aslında saldırgan daha fazla bellek kullanmak pahasına saldırı hızını arttırmak için "ortada karşılaşma (meet-in-the-middle)" tekniği olarak bilinen saldırıyı kullanabilir. Bu saldırı [27] de detaylı olarak tanımlanmıştır.

BÖLÜM 5

NTRU ÜZERİNE SALDIRILAR

Bu bölümde NTRU kriptosistemi üzerine düzenlenen saldırılardan bahsedilmiştir. Bu saldırılardan sistemi korumak için, güvenlik parametreleri NTRU şirketi tarafından yapılan deneyler sonucu belirlenir. Bugün, NTRU'nun bu saldırılara karşı güvenli olduğu kanıtlanmıştır ve Efficient Embedded Security Standards (EESS#1) [28] olarak standart hale getirilmiştir. Bu standart, şifreleme ve imzanın birlikte uygulanmasına ve geliştirilmesine yardımcı olmak için gerekli bilgileri içerir. NTRU'nun şimdiki versiyonlarında desteklenen parametre seçenekleri için [28] de mevcuttur.

Aşağıda bir kriptosistem için en genel saldırı olan kaba kuvvet saldırısı verilmiştir.

5.1 Kaba Kuvvet Saldırısı (Brute Force Attack)

Bölüm 3'te bahsedildiği gibi, NTRU'nun anahtar üretimi, açık olarak bilinen d_f ve d_g parametreleri ile belirli $\mathcal{L}_f$ ve $\mathcal{L}_g$ kümelerinden seçilen f ve g polinomları ile oluşturulur. Saldırgan NTRU'nun gizli anahtarı üzerine kaba kuvvet saldırısı uygulayabilir. İlk olarak, d_f ile belirli olası tüm $f \in \mathcal{L}_f$ leri sıralar ve bu f polinomları ile $g \equiv f * h \,(mod\ q)$ polinomlarını hesaplar. Ayrıca d_g parametresi ile bu g polinomlarının küçük katsayılara sahip olup olmadığını test eder. Daha sonra $g * h^{-1}\ (mod\ q)$'nin küçük katsayılara sahip olup olmadığını test ederek gizli anahtarı elde edebilir.

NTRU'nun parametre kümesine (Çizelge 3.1) göre d_g, d_f'den daha küçüktür. Yani $\mathcal{L}_g$'nin arama uzayı $\mathcal{L}_f$'den daha küçük olacaktır. Bu yüzden pratikte tüm $g \in \mathcal{L}_g$

polinomlarını denemek daha mantıklıdır. Eğer tüm g polinomları sıralanarak $g * h^{-1}$ polinomları hesaplanırsa, arama süresi $|\mathcal{L}_g| = \binom{n}{d_g}$ olacaktır.

Benzer şekilde, mesaj üzerine de kaba kuvvet saldırısı uygulanabilir. Önceki bölümlerden mesajın $e - r * h \ (mod\ q)$'ya eşit olduğunu hatırlayalım. Olası tüm $r \in \mathcal{L}_r$'er denenerek ve hesaplanan polinomların küçük katsayılara sahip olup olmadığı test edilerek mesaja ulaşılabilir. Bu durumda arama süresi $|\mathcal{L}_r| = \binom{n}{d_r}$ olacaktır.

n büyük bir sayı olarak seçildiği zaman, arama uzayları da büyük olacağından bu saldırı etkili olmayacaktır.

5.2 Ortada Karşılaşma Saldırısı (Meet in the Middle Attack)

NTRU'nun r rastgele polinomu kullanılarak bir ortada buluşma saldırısının uygulanabileceğini ilk olarak Andrew Odylzko dikkat çekti. Sonra NTRU'nun tasarımcıları bu saldırının f gizli anahtarına karşı da kullanılabileceğini gözlemledi [27].

$\mathcal{R}$'de a polinomunun i_inci rotasyonu $a * x^i$ veya $a^i = [a_{n-i}, a_{n-i+1}, \dots, a_{n-1}, a_0, a_1, \dots, a_{n-i-1}]$ olarak ifade edilir. NTRU'nun anahtar üretiminde, eğer $f * h \equiv g \ (mod\ q)$ (yani $h \equiv f_q^{-1} * g \ (mod\ q)$) ise f ve g nin i_inci rotasyonları da denkliği sağlar yani $f * x^i * h \equiv g * x^i \ (mod\ q)$ sağlar. Bu yüzden, eğer orijinal anahtar ile aynı sayıda d_f ve d_g içeren ve $f * h \equiv g \ (mod\ q)$ yu sağlayan bir (f, g) çifti bulunabilirse, bu orijinal anahtar çiftinin bir rotasyonu olabilir. Başka bir deyişle, f'nin bir rotasyonu bulunduğu zaman, f gizli anahtarı için geriye sadece n olasılık kalır.

f ve g'nin ikili polinomlar olduğu ve n, d_f'nin çift olduğu bir durumda bu saldırıyı tanımlayalım. Ayrıca q'nun, 2'nin bir kuvveti olduğunu varsayalım. (Diğer değerler için değişiklik kolaydır.) Bu saldırıda f , $f_1 \| f_2$ şeklinde iki parçaya bölünür, burada $\|$ sembolü birleştirmeyi (concatenation) gösterir. f_1 ve f_2'nin her biri $d_f/2$ tane 1'e sahip olmak üzere $n/2$ uzunluğundadır. (Aslında burada f nin 1'lerinin sayısı iki parçaya eşit bir şekilde ayrılmayabilir fakat f'nin rotasyonlarından en az birinin bu özelliği sağladığı gösterilmiştir [27].) Sonra saldırı şöyle devam eder:

$f * h \equiv g \ (mod\ q)$

$(f_1 \parallel f_2) * h \equiv g \ (mod\ q)$

$f_1 * h + f_2 * h \equiv g \ (mod\ q)$

g'nin i_inci koordinatı 0 veya 1 olduğu zaman, $f_1 * h$ ve $f_2 * h$'ın i_inci koordinatları aşağıdaki gibi olur.

$(f_1 * h)_i \equiv \{0, 1\} - (f_2 * h)_i \ \ (mod\ q)$

İlk olarak saldırgan f_1'leri sıralar. f_1 polinomu için $\binom{n/2}{d_f/2}$ farklı seçenek vardır. Daha sonra saldırgan $f_1 * h \ (mod\ q)$ polinomunu hesaplar ve her f_1'i bir diziye saklamak için bu polinomun ilk k koordinatını kullanır. k tane koordinatın her biri için şunlar yapılır: Eğer katsayının değeri $[0, q/2)$ aralığında ise dizinin ilgili girdisi 0 olacaktır. Eğer katsayının değeri $[q/2, q)$ aralığında ise ilgili girdi 1 olacaktır. Tüm k tane bit hesaplandıktan sonra, f_1 polinomu bu k bit uzunluklu dizi ile etiketlenerek saklanır. En fazla 2^k dizi olabilir ve saldırgan $\binom{n/2}{d_f/2}$ olası f_1'e sahiptir. Bu nedenle saldırgan başlangıçta, maksimum sayıdaki olası polinomları saklayacak yeterli dizi sayısı için $\binom{n/2}{d_f/2} < 2^k$ olacak şekilde k'yı seçmelidir.

İkinci adımda, saldırgan f_2'leri sıralar ve $f_2 * h$ polinomlarını hesaplar. Yine $\binom{n/2}{d_f/2}$ farklı f_2 vardır. Saldırgan her $0 \leq i < k$ için $(f_1 * h)_i \equiv \{0, 1\} - (f_2 * h)_i \ \ (mod\ q)$ denkliğini sağlayan doğru (f_1, f_2) çifti arayacaktır. f_1 polinomları dizide saklandığından saldırgan uygun dizileri arar. Her f_2 polinomu için saldırgan şu kuralları kullanarak $-f_2 * h$ polinomunun dizi değerini hesaplar: ilk adımda dizinin girdilerini belirler ve sonra f_1'i içeren diziye gider. Fakat $-f_2 * h$'ın her katsayısı için 1 ile eklenmiş olma olasılığı vardır. Bu yüzden eğer $-(f_2 * h)\ mod\ q$ polinomunun ilgili katsayısına 1 eklenerek değiştirilen dizi değeri kritik bir sayı ise saldırgan olası diziyi de inceler. Son olarak, saldırgan bu dizilerden f_1'leri alır ve $(f_1 \parallel f_2) * h \ (mod\ q)$ polinomunun ikili olup olmadığını kontrol eder. Eğer ikili polinom ise f gizli anahtarı bulunur. Bu, diziler içinde yer alan her f_1 için tekrarlanır. Saldırgan gizli anahtarı bulana kadar tüm f_2'leri araştırır.

Örneğin, $n = 2$ ve $q = 8$ olsun.

$f_1 * h\ (mod\ q) = [7,2,3,5]$ ise f_1, $[1,0,0,1]$ dizisi ile etiketlenerek saklanır.

$-f_2 * h\ (mod\ q) = [6,2,1,5]$ ise sadece $[1,0,0,1]$ dizisi elde edilir ve bu dizi ile etiketlenmiş f_1'ler alınarak $(f_1 \parallel f_2) * h\ (mod\ q)$ polinomunun ikili olup olmadığı kontrol edilir.

$-f_2 * h\ (mod\ q) = [6,2,3,5]$ ise $[1,0,0,1]$ ve $[1,0,1,1]$ dizileri ile etiketlenmiş f_1'ler için gerekli kontroller yapılır.

f iki parçaya bölündüğünden bu saldırının arama uzayı kaba kuvvet saldırısının arama uzayından daha küçüktür. Saldırgan f_1'leri diziler içinde sakladığından tüm f_1'leri kontrol etmesine gerek kalmaz, sadece ilgili dizilerin içindekileri kontrol eder. Daha önce belirtildiği gibi f'nin rotasyonlarından en az biri, ilk $n/2$ girdisinde $d/2$ tane 1, ikinci yarısında $d/2$ tane 1'e sahiptir. Bu özelliğe sahip f'nin $\sqrt{n}$'den daha fazla rotasyonu vardır [27]. Bu gözlemler ve bazı geliştirmeler ile ortada karşılaşma saldırısının tahmini çalışma süresi ve bellek gereksinimi [27] de

$$\frac{\binom{n/2}{d/2}}{\sqrt{n}}$$

olarak verilmiştir.

5.3 Kafes Saldırıları

NTRU kriptosisteminin güvenliği SVP kafes problemine dayanır. NTRU'nun tasarımcıları gizli anahtarın, LLL algoritması ile kafesin en kısa vektörünü hesaplayarak bulunamadığını gösterdi. Ancak Copppersmith ve Shamir orijinal gizli f anahtarını bulan veya şifreli metni deşifrelemek için f nin yerine kullanılabilecek bir f' alternatif anahtarı bulan başka kafes tabanlı bir saldırı sundu [29]. Saldırının amacı, h açık anahtarını kullanan bir kafes inşa ederek ve bu kafese kafes indirgeme algoritmaları uygulanarak, gizli anahtarı bulmaktır. Bu saldırıdan sonra NTRU güvenlik parametrelerini değiştirdi ve kafes saldırılarına karşı sistemi geliştirdi.

5.3.1 Standart NTRU Kafesi

n, p, q, d_f, d_g ve h bilgileri herkese açıktır. h açık anahtarı kullanılarak, bir L vektörler kümesi aşağıdaki gibi tanımlanır:

$$L = \{(a, b) \in \mathbb{Z}^{2n} |\ a * h \equiv b\ mod\ q\}$$

L kümesinin bir kafes olduğu ve (f, g) vektörünü içerdiği açıktır. Önerilen parametrelere bağlı olarak bu vektörün uzunluğu $\sqrt{2d_f - 1 + 2d_g}$ olacaktır (n ye göre oldukça küçüktür). Bu yüzden bu vektör L kafesinin kısa bir vektörü olacaktır.

Coppersmith ve Shamir 4 bloktan oluşan bir $2nx2n$ matrisinin satırları ile gösterilen bir taban buldu ve bu taban ile üretilen bir kafes inşa etti [29]. Bu kafes, NTRU kafesi olarak adlandırılır ve kafesi üreten taban matrisi L^{NT} ile gösterilir.

$$L^{NT} = \left[\begin{array}{cccc|cccc} \lambda & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{n-1} \\ 0 & \lambda & \cdots & 0 & h_{n-1} & h_0 & \cdots & h_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda & h_1 & h_2 & \cdots & h_0 \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right]$$

Bu matriste, h_i'ler h polinomunun i_inci katsayılarıdır. λ, kafes indirgemeyi daha etkili yapmak için seçilen küçük bir sabittir. NTRU kafesinin

$$L = \{(\lambda a, b) \in \mathbb{Z}^{2n} |\ a * h \equiv b\ mod\ q\}$$

kümesinin tüm elemanlarını içerdiği aşikardır ve bu durumda $(\lambda f, g)$ vektörü NTRU kafesinde olur. Bu vektörün uzunluğu $\sqrt{(2d_f - 1)\lambda^2 + 2d_g}$ dir. d_f, d_g ve λ küçük sayılar olduğundan, $(\lambda f, g)$ vektörü NTRU kafesinin kısa bir vektörü olur.

NTRU kafesinde gizli anahtardan başka kısa vektörler varsa ne olur? Yeterince kısa bir (a, b) vektörü mesajı deşifrelemek için kullanılabilir mi? Coppersmith ve Shamir bu sorulara cevap aramış ve bu vektörün gizli anahtarı bulmak için yeterli olmadığını göstermiştir. Ayrıca NTRU'nun tasarımcıları deneyler yaparak ve kafes indirgeme algoritmaları uygulayarak gizli anahtardan başka mesajı deşifreleyebilecek bir vektörün

mümkün olmadığını gösterdiler [7]. Bu yüzden kafes indirgeme algoritmalarının gizli anahtarı bulmak dışında bir seçeneği yoktur.

Bir kafesin en kısa vektörünün uzunluğu yaklaşık olarak $\sqrt{\frac{n}{2\pi e}}(\det L)^{1/n}$ dir [30]. Tanımlanan NTRU kafesinde, boyut $2n$ ve determinant $q^n\lambda^n$ dir. Bu değerler denklemde yerine yazılırsa NTRU kafesinin en kısa vektörünün uzunluğu yaklaşık olarak

$$\sigma(L) = \sqrt{\frac{n\lambda q}{\pi e}}$$

bulunur. Burada λ, NTRU kafesini oluştururken seçilen sabittir. Bu yüzden saldırgan λ değerine karar vermelidir.

Hedef vektörü olarak adlandırılan $(\lambda f, g)$ vektörünü τ ile gösterelim. Saldırgan kafes indirgeme metotlarını kullanarak bu vektörü bulmaya çalışır. En kısa vektörün beklenen uzunluğunun hedef vektörünün uzunluğuna oranı c_h kafes sabiti ile tanımlanır. Yani $c_h = \frac{\|\sigma(L)\|}{\|\tau\|}$ dir. NTRU kafesinde vektörlerin uzunluklarını yerine yazılarak, c_h sabiti

$$c_h = \frac{\sqrt{\frac{n\lambda q}{\pi e}}}{\sqrt{\lambda^2\|f\|^2 + \|g\|^2}}$$

olarak bulunur.

Deneylerde bu sabitin daha küçük alındığı zaman kafes indirgeme metotlarının daha iyi şekilde çalıştığı gözlenmiştir [30]. Bu yüzden saldırgan bu sabiti olabildiğince küçük yapan λ'yı seçmeye çalışır. τ'nin arama etkinliğini maksimize etmek için λ sabitinin $\lambda = {\|g\|}/{\|f\|}$ şeklinde seçilmesi gerekir. Saldırgan d_f ve d_g değerlerini bildiğinden λ'yı hesaplayabilir ve hedef vektörünü bulmak için kafes indirgemesini daha kolay uygular [30].

Kafes sabitinin azaltılması LLL çalışmasını daha kolay yapmasına rağmen, kısa bir vektör bulmak için gerekli zaman hala n üsteldir. Bu yüzden Coppersmith ve Shamir'in makalelerinden sonra, başka saldırılar öne sürülmüştür.

5.3.2 Sıfır Tekrarlı Kafes (Zero-Run Lattice)

d_f ve d_g bilindiğinden $(\lambda f, g)$ hedef vektörünün bileşenleri hakkında bir tahmin yapılabilir. Alexander May kafes indirgeme ile NTRU'nun kriptoanalizinde yeni sonuçlar ortaya çıkarmıştır [31]. f ve g polinomlarında çok sayıda 0 katsayısının var olduğunu ve bunun kafes saldırılarında bir avantaj olarak kullanılabileceğini gözlemlemiştir. May, g polinomunun r tane ardışık katsayısının 0 olduğunu varsayar. Gizli anahtarın tüm rotasyonları kafeste olduğundan, bu sıfırların yerleri önemli değildir. L^{NT} nin r ardışık sütununun büyük bir θ sayısı ile çarpılarak elde edilen kafes "sıfır tekrarlı kafes (zero-run lattice)" olarak adlandırılır ve taban matrisi L_θ^r ile gösterilir.

$$L_\theta^r = \left[\begin{array}{cccc|cccccc} \lambda & 0 & \cdots & 0 & h_0\theta & \cdots & h_{r-1}\theta & h_r & \cdots & h_{n-1} \\ 0 & \lambda & \cdots & 0 & h_{n-1}\theta & \cdots & h_{r-2}\theta & h_{r-1} & \cdots & h_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda & h_1\theta & \cdots & h_r\theta & h_{r+1} & \cdots & h_0 \\ \hline 0 & 0 & \cdots & 0 & q\theta & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & q \end{array}\right]$$

θ ile sütunları çarpmanın sonucunda, bu r tane koordinatı 0 olan ve buna karşılık gelen en kısa vektörü bulmak için kafes indirgeme algoritmaları uygulanır. Bir kafes vektöründe eğer r tane koordinatın herhangi biri sıfırdan farklı ise vektörün uzunluğunda bir θ çarpanı olacaktır. Bu yüzden bu vektör, kafesin bir kısa vektörü olamaz. Bu fikir kısa vektörlerin arama alanını (uzayını) azaltır.

Daha sonra Silverman, θ ile r tane ardışık sütunu çarpmak yerine θ ile r tane rastgele sütunu çarpmanın daha avantajlı olacağını önerdi [32].

5.3.3 Sıfır Kuvvetli Kafes (Zero-Forced Lattice)

Silverman [32], May'ın fikrini genelledi ve sıfır tekrarlı kafeslerden daha iyi performans sergileyen "sıfır kuvvetli kafes (zero-forced lattice)"i sundu. Silverman temel olarak, May gibi seçilen koordinatlardaki sıfırlar ile kısa vektörleri bulmak için kafes indirgeme algoritmaları uygular. Ancak koordinatları büyük bir sayı ile çarpmak yerine daha küçük boyutlu bir kafes oluşturmaya çalışır.

Bir sıfır kuvvetli kafesi oluşturmak için ilk adım $0 \leq j_1 < j_2 < \ldots < j_r < n$ 'i sağlayan $J = \{j_1, j_2, \ldots, j_r\}$ indisler kümesi seçmektir. Burada amaç $j_1, j_2, \ldots, j_r$ koordinatları 0 olan hedef vektörlerini araştırmaktır. Bunun için, L^{NT} kafesinin oluşturulmasında kullanılan orijinal kongrüanslar yazılır:

$$\forall\, 0 \leq i, j, k < n \; için \quad g_j \equiv \sum_{i+k \equiv j \;(mod\; n)} h_k f_i \;\; (mod\; q)$$

$g_{j_1}, g_{j_2}, \ldots, g_{j_r} = 0$ varsayılarak, $modulo\, q$'da r tane lineer bağıntı elde edilir. $\forall\, i \in \{i_1, i_2, \ldots, i_r\}$ için r tane kongrüans kolayca çözülebilir. Geriye kalan $n - r$ kongrüans ile yeni bir sistem oluşturulur.

$$\forall\, 0 \leq j < n, j \notin J\,, i \notin I \; için \quad g_j = \sum_{i+k \equiv j \;(mod\; n)} a_{k,j} f_i \;\; (mod\; q)$$

Bu sistemde $a_{i,j}$'ler bilinenler, f_i ve g_j'ler bilinmeyenlerdir. Dolayısıyla $2(n-r)$ bilinmeyenli $n - r$ tane kongrüanstan oluşan bir sistem elde edilir. Böylece aşağıdaki matrisin satırları tarafından üretilen $2(n-r)$ boyutlu bir kafes oluşturulabilir. Bu kafese sıfır kuvvetli kafes adı verilir.

$$L_j^{ZF} = \left[\begin{array}{cccc|cccc} \lambda & 0 & \cdots & 0 & & & & \\ 0 & \lambda & \cdots & 0 & & & & \\ \vdots & \vdots & \ddots & \vdots & & a_{i,j} & & \\ 0 & 0 & \cdots & \lambda & & & & \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right]$$

Sıfır kuvvetli kafes kullanılarak yapılan saldırıda, bu kafese kafes indirgeme algoritmaları uygulanır ve g'nin her $j \in J$ katsayısı 0'a eşit iken (f, g) hedef vektörü bulunmaya çalışılır. Bu kafesin boyutu, standart NTRU kafesinden daha küçük olduğundan kafes indirgeme algoritmalarının çalışma süresi azalacaktır.

5.3.4 Boyutu İndirgenmiş Kafes

May, sıfır kuvvetli kafeslere alternatif olarak taban matrisinden birkaç sütunun atılmasını önerdi [31]. Böylece kafesin boyutu azalır ve kafes indirgeme hızlanır. Bu

saldırıda koordinatlar tekrar rastgele seçilir ve bu rastgele seçilmiş koordinatlar atılarak L^{NT} kafesinden yeni bir kafes inşa edilir. Bu yeni kafese indirgeme algoritmaları uygulanarak, başlangıçta atılan bu koordinatlar bilinmediğinden, hedef vektörünün tümü elde edilemez. Böylece bu yeni hedef vektörün uzunluğu onun gerçek uzunluğundan daha küçük olacaktır. Silverman [32], bu saldırı ile hedef vektörün uzunluğunun en kısa vektörün beklenen uzunluğuna yaklaşacağını iddia etti. Teorik olarak c_h kafes sabiti 1'e yaklaşır. Yani hedef vektörünü, diğer (uzunluğu, en kısa vektörün beklenen uzunluğuna yaklaşık olarak eşit olan) kısa vektörlerden ayırt etmek zor olacaktır. May tarafından gösterilen deneylerde [31], düşük boyutlarda indirgeme algoritması hızlıdır. Ancak n 100 ü aştığı zaman, bu saldırı ile sistemi kırmak için indirgeme algoritmalarının hızı yeterli değildir.

BÖLÜM 6

SONUÇ VE ÖNERİLER

Günümüzde aktif olarak kullanılan açık anahtar kriptosistemleri sayılar teorisindeki bazı problemlerin zorluğuna dayanır. İşlemci gücünün giderek artması, kuantum ve dağıtık hesaplamaların geliştirilmesiyle bu kriptosistemlerin güvenilirlikleri gelecekte tehlike altına girecektir. Bu nedenle başka matematiksel problemleri kullanan yeni kriptosistemler üretilmelidir ve alternatif olarak sunulmalıdır. Bu amaçla kafes problemlerinin zorlukları üzerine yapılan araştırmalar Kafes Tabanlı Kriptosistemleri ortaya çıkarmıştır.

Bu tezde, NTRU'nun matematiksel temelleri üzerinde çalışılmıştır. Hazırlık olarak, kafesin temel özellikleri, bazı kafes problemleri ve kafes indirgeme algoritmalarına değinilmiştir. Daha sonra NTRU'nun anahtar üretimi, şifreleme ve deşifreleme işlemleri örneklerle açıklanmıştır. Deşifreleme işlemlerinde bazı parametrelerin başarısızlığa neden olabileceği gösterilmiştir. NTRU'nun hesaplamalarında en çok zaman alan işlemler polinomların çarpımını ve bir polinomun tersini hesaplamaktır. Bazı parametrelerin farklı formda alınarak, işlemlerin hızlandırıldığı gösterilmiştir ve bu yeni parametre kümeleriyle oluşturulan NTRU kriptosistemine sayısal örnekler verilmiştir. Son kısımda ise NTRU üzerine yapılan saldırı teknikleri açıklanmıştır.

KAYNAKLAR

[1] Diffie, W. ve Hellman, M.E., (1976). "New Directions in Cryptography", IEEE Transactions on Information Theory, 22: 644-654.

[2] Rivest, R.L., Shamir, A. ve Adleman, L., (1978). "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", Communications of the ACM 21, 120-126.

[3] Miller, V., (1985). "Use of Elliptic Curves in Cryptography", Advances in Cryptology CRYPTO '85, Lecture Notes in Computer Science, 218: 417-426.

[4] Ajtai, M., (1996). "Generating Hard Instances of Lattice Problems",ECCC, TR96-007.

[5] Ajtai, M., (1998). "The Shortest Vector Problem in L_2 is NP-hard for Randomized Reductions", Proceedings of the 30th Annual ACM Symposium on Theory of Computing, New York, USA, 10-19, ACM Press.

[6] Goldreich, O., Goldwasser, S. ve Halevi, S., (1996). "Public Key Cryptosystems from Lattice Reduction Problems", Electronic Colloquim on Computational Complexity (ECCC), TR96-056.

[7] Hoffstein, J., Piper, J. ve Silverman, J.H., (1998). "NTRU: A ring-based public key cryptosystem", Algorithmic Number Theory, Lecture Notes in Computer Science, 1423: 267-288.

[8] Dwork, C., (1998). Lattices and Their Application to Cryptography, Lecture Notes, Stanford University, Spring 1998.

[9] Regev, O., (2009). Lattices in Computer Science, Tel Aviv University, Fall 2009, http://www.cs.tau.ac.il/~odedr/teaching/lattices_fall_2009/index.html, 25 Mayıs 2011.

[10] Goldreich, O., Micciancio, D., Safra, S. ve Seifert, J.P., (1999). "Approximating Shortest Lattice Vectors is not Harder than Approximating Closest Lattice Vectors", Information Proceeding Letters 71(2): 55-61.

[11] Cai, J.Y., (2000). "The Complexity of Some Lattice Problems", Algorithmic Number Theory, Lecture Notes in Computer Science, 1838: 1-32.

[12] Schnorr, C.P. ve Euchner, M., (1994). “Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems”, Mathematical Programming, 66: 181-199.

[13] Lenstra, A.K., Lenstra, H.W. ve Lovasz, L., (1982). “Factoring Polynomials with Rational Coefficients”, Mathematische Annalen, 261: 515-534.

[14] Cohen, H., (1995). A Course in Computational Algebraic Number Theory, Springer-Verlag, New York.

[15] Miccincio, D., (1995). “Lattices in Cryptography and Cryptanalysis”, http://cseweb.ucsd.edu/~daniele/CSE207C/, 25 Mayıs 2011.

[16] Schnorr, C.P., (1987). “A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms”, Theoretical Computer Science, 53: 201-224.

[17] Hoffstein, J., Lieman, D., Pipher, J. ve Silverman, J.H., (1999). “NTRU: A public key cryptosystem”, http://grouper.ieee.org/groups/1363/lattPK/submissions.html, 25 Mayıs 2011.

[18] Hoffstein, J. ve Silverman, J., (2000). “Optimizations for NTRU”, Public Key Cryptography and Computational Number Theory, Warsaw.

[19] Howgrave-Graham, N., Silverman, J.H. ve Whyte, W., (2005). “Choosing Parameter Sets for NTRUEncrypt with NAEP and SVES-3”, Topics in Cryptology CT-RSA, Lecture Notes in Computer Science, 3376: 118-135.

[20] Silverman, J.H., (1998). “Invertibility in Truncated Polynomial Rings”, NTRU Cryptosystems Technical Report 9, http://www.securityinnovation.com/cryptolab/tech_notes.shtml, 25 Mayıs 2011.

[21] Silverman, J.H., (1999). “Almost Inverses and Fast NTRU Key Creation”, NTRU Cryptosystems Technical Report 14, http://www.securityinnovation.com/cryptolab/tech_notes.shtml, 25 Mayıs 2011.

[22] Silverman, J.H., (1999). “High-Speed Multiplication of (Truncated) Polynomials”, NTRU Cryptosystems Technical Report 10, http://www.securityinnovation.com/cryptolab/tech_notes.shtml, 25 Mayıs 2011.

[23] Security Innovation, Inc., “The NTRU Public Key Cryptosystem - A Tutorial”, http://securityinnovation.com/pdf/Ntru_Public_Key_Cryptosystem_Tutorial.pdf, 25 Mayıs 2011.

[24] Silverman, J.H., ve Whyte, W. “Estimating Decryption Failure Probabilities for NTRUEncrypt”, NTRU Cryptosystems Technical Report 18, http://www.securityinnovation.com/cryptolab/tech_notes.shtml, 25 Mayıs 2011.

[25] Silverman, J.H., (2001). “Wraps, Gaps and Lattice Constants”, NTRU Cryptosystems Technical Report 11- Version 2, http://www.securityinnovation.com/cryptolab/tech_notes.shtml, 25 Mayıs 2011.

[26] Hoffstein, J. ve Silverman, J.H., (2003). “Random Small Hamming Weight Products with Applications to Cryptography”, Discrete Applied Mathematics, 130: 37-49.

[27] Silverman, J.H., (2003). “A Meet in The Middle Attack on an NTRU Private Key”, NTRU Cryptosystems Technical Report 4, Version 2, http://www.securityinnovation.com/cryptolab/tech_notes.shtml, 25 Mayıs 2011.

[28] Efficient Embedded Security Standards (EESS), (2003). EESS#1: Implementation Aspects of NTRUEncrypt and NTRUSign, Version 2.0, Consortium for Efficient Embedded Security, http://grouper.ieee.org/groups/1363/lattPK/submissions/EESS1v2.pdf, 25 Mayıs 2011.

[29] Coppersmith, D. ve Shamir, A., (1997). “Lattice Attacks on NTRU”, Advances in Cryptology – Eurocrypt’97, Lecture Notes in Computer Science, 1233: 52-61.

[30] Hoffstein, J., Silverman, J.H. ve Whyte, W., (2003). “Estimated Breaking Times for NTRU Lattices”, NTRU Cryptosystems Technical Report 12, Version 2, http://www.securityinnovation.com/cryptolab/tech_notes.shtml, 25 Mayıs 2011.

[31] May, A., (1999). “Cryptanalysis of NTRU”, Yayınlanmamış Ön Baskı, www.informatik.uni-frankfurt.de/~alex/ntru.ps, 25 Mayıs 2011.

[32] Silverman, J.H., (1999). “Dimension-Reduced Lattices, Zero-Fored Lattices and the NTRU Public Key Cryptosystem”, NTRU Cryptosystems Technical Report 13, http://www.securityinnovation.com/cryptolab/tech_notes.shtml, 25 Mayıs 2011.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Sevcan TEKİN

Doğum Tarihi ve Yeri : 05.12.1987 / Bakırköy

Yabancı Dili : İngilizce

E-posta : st_34@hotmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Matematik	Yıldız Teknik Üniversitesi	2010
Lise	Fen Bilimleri	Cibali Lisesi	2004