University of Bergen
Department of Informatics

Hidden parallelepiped in Hawk

Author: Eirik Djupvik Skjerve

Supervisors: Igor Aleksandrovich Semaev & Martin Feussner

UNIVERSITETET I BERGEN

Det matematisk-naturvitenskapelige fakultet

January, 2025

Abstract

Some abstract here

Acknowledgements

Thank you to some people

Eirik D. Skjerve

Wednesday 8^{th} January, 2025

Contents

1 Introduction **1**

1.1 Context and motivation 1

1.2 Objectives 2

2 Background **3**

2.1 Asymmetric cryptography 3

2.1.1 Digital signatures 3

2.1.2 Hash-Then-Sign 3

2.1.3 GGH 3

2.2 Linear algebra & lattices 3

2.2.1 Lattices 3

2.3 Probability theory 3

2.3.1 Distributions 3

3 Hawk and *Learning a parallelepiped* **4**

3.1 Hawk 4

3.2 Learning a parallelepiped 4

3.2.1 Solving the Hidden Parallelepiped Problem 4

3.2.2 HPP against normally distributed samples 5

4 Implementation **8**

4.1 Implementation of Hawk 8

4.2 Implementation of HPP 8

5 Adapting HPP to Hawk **9**

5.1 Covariance matrix secret key in Hawk 9

5.2 Secret Key Recovery 9

Bibliography **10**

A Generated code **11**

Chapter 1

Introduction

1.1 Context and motivation

Digital signatures are an integral part of secure communication today. The widely used DSA (Digital Signature Algorithm) and RSA (Rivest, Shamir, and Adleman) signature schemes are in peril due to the potential emergence of quantum computers which, theoretically, are able to break the hard problems DSA and RSA-sign are based upon. Whether practical quantum computers with these powers will emerge any time soon is debatable. However, measures against this potential looming threat has already begun. In 2016, NIST (National Institute of Standards and Technology) announced a process for selecting new standard schemes for Key Encapsulation Methods (abbr. KEMs) and digital signatures that are resilient against attacks based on quantum computers (source to NIST pages here). Many of the submissions to this process (including KRYSTALS-Dilithium which is to be standardized) are based on lattice problems that are believed to be hard to solve for both classical and quantum computers.

Cryptographic schemes based on lattice problems are not an enirely new phenomenon, however. The NTRU scheme and its signature counterpart NTRU-Sign, published in 2003 (source for this), is a digital signature scheme based on the hardness of the Closest Vector Problem. The original scheme was broken by Phong. Q. Nguyen & Oded Regev in 2006

1.2 Objectives

The objective for this thesis consists of two main parts:

- **Implementation of Hawk in Rust**. As the first part of this thesis I implement the Hawk digital signature scheme according to
- **Cryptanalysis and experimentation**. The second part of this thesis is cryptanalysis of Hawk. The goal is to use the *Learning a parallelepiped* attack

Chapter 2

Background

2.1 Asymmetric cryptography

2.1.1 Digital signatures

2.1.2 Hash-Then-Sign

2.1.3 GGH

2.2 Linear algebra & lattices

2.2.1 Lattices

2.3 Probability theory

2.3.1 Distributions

Chapter 3

Hawk and *Learning a parallelepiped*

3.1 Hawk

In the following Hawk, the digital signature scheme, will be presented, as in

3.2 Learning a parallelepiped

The paper *Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures* by Phong Q. Nguyen and Oded Regev from 2006 introduced a method for breaking digital signature schemes based on the GHH scheme

3.2.1 Solving the Hidden Parallelepiped Problem

First we define an idealized version of both the problem to solve and the solution as proposed in

Learning a Hypercube: The second step is to learn the hypercube. Given samples over $\mathcal{P}(C)$, we deduce the rows of the secret matrix C with the method described in Algorithm 1. After the rows are approximated, one can multiply the rows $\{\mathbf{c}_1, ..., \mathbf{c}_2\}$ by L^{-1} such that we have $\{\mathbf{v}_1, ..., \mathbf{v}_2\}$, and we are done.

Algorithm 1 Learning a Hypercube

Require: Descent parameter δ, samples $\mathcal{X}$ uniformly distributed over $\mathcal{P}(C)$
Ensure: A row vector $\pm\mathbf{v}_i$ of C
  Choose uniformly at random $\mathbf{w}$ on the unit sphere of $\mathbb{R}^n$
  loop
    Compute $\mathbf{g}$, an approximation of $\nabla mom_4(\mathbf{w})$
    Let $\mathbf{w}_{new} = \mathbf{w} - \delta\mathbf{g}$
    Place $\mathbf{w}_{new}$ back on the unit sphere by dividing it by $\|\mathbf{w}_{new}\|$
    if $mom_4(\mathbf{w}_{new}) \geq mom_4(\mathbf{w})$ **then** ▷ mom_4 are approximated by samples
      return w
    else
      Replace $\mathbf{w}$ with $\mathbf{w_{new}}$ and continue loop
    end if
  end loop

3.2.2 HPP against normally distributed samples

In the following, we see what happens to the computations the *Learning a parallelepiped* attack is based on if we replace the uniform distribution by a normal distribution. The key component and assumption of the *Learning a parallelepiped* attack is that the provided samples are distributed uniformly over $\mathcal{P}(V)$. Recall that $\mathcal{P}(V)$ is defined as $\{\sum_{i=1}^{n} x_i\mathbf{v}_i : x_i \in [-1,1]\}$ where $\mathbf{v}_i$ are rows of V and x_i is uniformly distributed over $[-1,1]$ (generally one can take another interval than $[-1,1]$ and do appropriate scaling). One runs into trouble if the sampled vectors are on the form $\mathbf{v} = \mathbf{x}V$ where $\mathbf{x}$ follows a normal distribution, i.e. $x_i \sim \mathcal{N}(\mu, \sigma^2)$. Although one might be able to approximate the covariance matrix V^tV, one can not do a gradient descent based on the fourth moment given such samples; the fourth moment is constant since the samples form a hypersphere.

Adapting the definition of $\mathcal{P}(V)$ Firstly, the distribution $\mathcal{N}(\mu, \sigma^2)$ is defined over the interval $(-\infty, \infty)$, so it does not make sense to talk about samples "normally distributed over $\mathcal{P}(V)$" without tweaking any definitions. Therefore, let $[-\eta, \eta]$ be a finite interval on which to consider a truncated normal distribution $\mathcal{N}_\eta(\mu, \sigma^2)$ such that $\int_{-\eta}^{\eta} f_X(x)dx = 1 - \delta$ for some negligibly small δ where $f_X(x)$ is the probability density function of $\mathcal{N}(\mu, \sigma^2)$. Now we consider $\mathcal{P}_\eta(V) = \{\sum_{i=1}^{n} x_i\mathbf{v}_i : x_i \in [-\eta, \eta]\}$ and proceed as in the original HPP with $\mathcal{P}_\eta(V)$ instead of $\mathcal{P}(V)$.

Approximating V^tV Let $V \in \mathcal{GL}_n(\mathbb{R})$. Let $\mathbf{v}$ be chosen from a truncated normal distribution $\mathcal{N}_\eta(0, \sigma^2)$ over $\mathcal{P}_\eta(V)$. Then $\lim_{\eta\to\infty} \mathbb{E}[\mathbf{v}^t\mathbf{v}] = V^tV \cdot \sigma^2$.

Proof. Let samples be on the form $\mathbf{v} = \mathbf{x}V$, where $\mathbf{x}$ is a row vector where each element $x_i \sim \mathcal{N}_\eta(0, \sigma^2)$. Then $\mathbf{v}^t\mathbf{v} = (\mathbf{x}V)^t(\mathbf{x}V) = (V^t\mathbf{x}^t)(\mathbf{x}V) = V^t\mathbf{x}^t\mathbf{x}V$. Considering $\mathbb{E}[\mathbf{x}^t\mathbf{x}]$, we see that for $i \neq j$, $\mathbb{E}[x_i x_j] = \mathbb{E}[x_i]\mathbb{E}[x_j] = 0 \cdot 0 = 0$ due to independent random variables. For $i = j$, $\lim_{\eta\to\infty} \mathbb{E}[x_i^2] = \mathbb{V}[x_i] = \sigma^2$ since $\mathbb{V}[x_i] = \mathbb{E}[x_i^2] - \mathbb{E}[x_i]^2 = \mathbb{E}[x_i^2] - 0 = \sigma^2$. Therefore, $\lim_{\eta\to\infty} \mathbb{E}[\mathbf{x}^t\mathbf{x}] = I_n \cdot \sigma^2$, i.e. the matrix with σ^2 on the diagonal. Consequently, $\lim_{\eta\to\infty} \mathbf{v}^t\mathbf{v} = V^t\mathbb{E}[\mathbf{x}^t\mathbf{x}]V = V^t(I_n \cdot \sigma^2)V = (V^tV) \cdot \sigma^2$ and conversely $\lim_{\eta\to\infty} V^tV = (\mathbf{v}^t\mathbf{v})/\sigma^2$. □

This means that we can in theory approximate the covariance matrix V^tV by averaging over $\mathbf{v}^t\mathbf{v}$ and dividing by σ^2. However, it is not immediately clear if one needs more samples for this approximation than in the original attack due to the difference in distributions. In addition, it is not clear exactly how accurate our approximation needs to be for the remaining parts of the attack to work.

Hypercube transformation Assume now that we know V^tV. Consider instead of $\mathcal{P}(V)$, $\mathcal{P}_\eta(V)$. Then by following part 1 of **Lemma 2** and its proof from

Learning a hypercube It is clear that samples uniformly over $\mathcal{P}_\eta(C)$ centered at the origin form a hypersphere for which any orthogonal rotation leaves the sphere similar in shape. As a consequence, the fourth moment of one-dimensional projections is constant over the unit circle, which renders the gradient search method in

Analogous to

For $k = 4$:

$$\mathbb{E}[(\sum_{i=1}^{n} x_i\langle \mathbf{v}_i, \mathbf{w}\rangle)^4] = \mathbb{E}[\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} x_i x_j x_k x_l \langle \mathbf{v}_i, \mathbf{w}\rangle\langle \mathbf{v}_j, \mathbf{w}\rangle\langle \mathbf{v}_k, \mathbf{w}\rangle\langle \mathbf{v}_l, \mathbf{w}\rangle]$$

We consider three cases for the indices i, j, k, and l:

1. **None equal**: if i, j, k, and l are different, the expression equals 0 due to independent random variables.
2. **All equal**: if $i = j = k = l$, then we have $\sum_{1=1}^{n} \mathbb{E}[x_i^4]\langle \mathbf{v}_i, \mathbf{w}\rangle^4$. A well known result for the normal distribution $\mathcal{N}(0, \sigma^2)$ is that $\mathbb{E}[x^4] = 3\sigma^4$.
3. **Pairwise equal**: if either
   - $i = j \neq k = l$
   - $i = l \neq j = k$
   - $i = k \neq j = l$

we have the following:

$$\sum_{i \neq j} \mathbb{E}[x_i^2 x_j^2] \langle \mathbf{v}_i, \mathbf{w} \rangle^2 \langle \mathbf{v}_j, \mathbf{w} \rangle^2$$

Since $\mathbb{E}[x_i^2 x_j^2] = \mathbb{E}[x_i^2]\mathbb{E}[x_j^2] = \sigma^4$ due to independent random variables, we have

$$\sigma^4 \sum_{i \neq j} \langle \mathbf{v}_i, \mathbf{w} \rangle^2 \langle \mathbf{v}_j, \mathbf{w} \rangle^2$$

Putting together the expressions above we have

$$mom_{V,4}(\mathbf{w}) = 3\sigma^4 \sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{w} \rangle^4 + 3(\sigma^4 \sum_{i \neq j} \langle \mathbf{v}_i, \mathbf{w} \rangle^2 \langle \mathbf{v}_j, \mathbf{w} \rangle^2)$$

since there are three cases where indices pair up two and two. The final result becomes:

$$mom_{V,4}(\mathbf{w}) = 3\sigma^4 (\sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{w} \rangle^4 + \sum_{i \neq j} \langle \mathbf{v}_i, \mathbf{w} \rangle^2 \langle \mathbf{v}_j, \mathbf{w} \rangle^2) \tag{3.1}$$

Claim: If $V \in \mathcal{O}(\mathbb{R})$, and $\mathbf{w}$ is on the unit sphere, $mom_{V,4}(\mathbf{w})$ is constant.

Proof. This can be shown by rewriting (3.2) as

$$mom_{V,4}(\mathbf{w}) = 3\sigma^4 (\sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{w} \rangle^4 + \sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{w} \rangle^2 \sum_{j=1}^{n} \langle \mathbf{v}_j, \mathbf{w} \rangle^2 - \sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{w} \rangle^4)$$

$$mom_{V,4}(\mathbf{w}) = 3\sigma^4 (\sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{w} \rangle^2 \sum_{j=1}^{n} \langle \mathbf{v}_j, \mathbf{w} \rangle^2) = 3\sigma^4 (\sigma^2 \|\mathbf{w}\|^2)^2 = 3\sigma^8$$

because $mom_{V,2}(\mathbf{w}) = \sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{w} \rangle^2 = \sigma^2 \|\mathbf{w}\|^2$ when $V \in \mathcal{O}(\mathbb{R})$ and $\|\mathbf{w}\|^2 = 1$ when $\mathbf{w}$ lies on the unit sphere. $\square$

In conclusion, if samples over the secret parallelepiped $\mathcal{P}_\eta(V)$ follow a continuous normal distribution, a gradient descent based on the fourth moment described in

The discrete Gaussian distribution We now do the same computations considering the discrete Gaussian distribution as described in

Chapter 4

Implementation

Introduction to the implementation part of the thesis

4.1 Implementation of Hawk

Something something about the implementation of Hawk in Rust. Mentions of sampling, integer/float types, speed and comparison to Hawk team's C code?

4.2 Implementation of HPP

Something something about the implementation of HPP in Rust. Something about speedup/parallelization of gradient descent?

Chapter 5

Adapting HPP to Hawk

In this chapter we investigate the steps needed to possibly apply the Hidden Parallelepiped Problem to the Hawk digital signature scheme.

5.1 Covariance matrix secret key in Hawk

Nothing yet I'm afraid

5.2 Secret Key Recovery

Since $\mathbf{x}$ follows some distribution close to some normal distribution, we hope that enough vectors $\mathbf{w}$ will disclose some information about $\mathbf{B}^{-1}$. If we know $\mathbf{B}^{-1}$ we know $\mathbf{B}$. This is the goal.

Bibliography

Appendix A

Generated code

Listing A.1: Source code of something

```
println!("Goodbye World");
```