University of Bergen
Department of Informatics

Learning a Parallelepiped attack against Hawk digital signature scheme

Author: Eirik Djupvik Skjerve

Supervisors: Igor Aleksandrovich Semaev & Martin Feussner

UNIVERSITY OF BERGEN
Faculty of Science and Technology

February, 2025

Abstract

In this work, we do cryptanalysis on the Hawk digital signature scheme using the *Learning a Parallelepiped* method which broke the GGH and basic NTRU digital signature schemes.

Acknowledgements

Acknowledgements here

`Eirik D. Skjerve`

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Context and motivation

Digital signatures are an integral part of secure communication today. They enable a receiver of a digital message to mathematically verify the sender is who they say they are. The widely used Digital Signature Algorithm (DSA) and the Rivest, Shamir, Adleman (RSA) signature scheme are in peril due to the potential emergence of quantum computers which can 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 the looming threat has already begun. In 2016, the National Institute of Standards and Technology (NIST) announced a process for selecting new standard schemes for Key Encapsulation Methods (KEMs) and digital signatures that are resilient against quantum attacks (https://www.nist.gov/pqcrypto). 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 entirely new phenomenon, however. NTRU-Sign [2], the signature counterpart of the NTRU crypto-system, is a digital signature scheme based on the hardness of the Closest Vector Problem (CVP), a well known lattice problem (source?). The original scheme was broken by Phong. Q. Nguyen & Oded Regev in 2006 [6]; not by solving the CVP, but by retrieving a secret key by observing enough signatures. In other words, each signature leaks some information about the secret key. The title of their paper and the name of the attack is *Learning a Parallelepiped*, and the problem to solve in this attack will henceforth be denoted as the Hidden

Parallelepiped Problem (HPP) as one tries to *learn* a parallelepiped. Countermeasures for this attack was proposed, but ultimately broken again in 2012 due to a more advanced extension of the original attack [1].

Hawk [3] is a digital signature scheme submitted to NIST's standardization process and is a viable candidate for standardization due to its speed and small signature- and keysizes. It is also a lattice-based signature scheme akin to NTRU-sign, but with some significant changes, and a different underlying hard problem on which its security is based upon. This thesis will investigate if a method based on HPP can be aimed at Hawk to retrieve information about the secret key, and ultimately break the scheme.

1.2 Objectives

The objective of 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 [3] in the Rust programming language. [1] Implementing a scheme and its algorithms is a good way to more deeply learn how it works. I chose to do the implementation in Rust for the sake of becoming more adept at this programming language as a personal bonus objective of the thesis. Moreover, having ones own implementation of a scheme makes it easier to experiment on, run simulations with, adjust, and modify it to ones need. It would in any case be challenging to understand and work with dense, long, and complicated source code someone else has written. For the Hawk teams source code in C and a reference implementation in Python see https://github.com/hawk-sign.
- **Cryptanalysis and experimentation**. The second part of this thesis is cryptanalysis of Hawk. The goal is to use the *Learning a parallelepiped* attack and do suitable modifications to attack Hawk. This requires both theoretical and practical work, and experiments will, like the Hawk implementation itself, be implemented in Rust.

[1] Disclaimer: this implementation is not meant to be comparable with the Hawk teams implementation for real life usage, as it is not highly optimized and not all formal requirements are met.

1.3 Thesis outline

Chapter 2 will introduce important notions and mathematical background used in this thesis. Chapter 3 will introduce Hawk and its implementation, and the *Learning a Parallelepiped* attack. In Chapter 4 the cryptanalysis of Hawk is presented. The final chapter will discuss results and future work.

1.4 Detailed tentative roadmap

1. Introduce idea of a digital signature
2. Introduce lattice facts and lattice problems used in digital signatures
3. Introduce other linear algebra and statistics / probability theory stuff
4. Introduce notion of gradient search and variations
5. Describe Hawk in detail
6. Describe Hawk implementation in detail
7. Describe basic HPP attack using notation from original paper
8. Proof and discussion of HPP against normally distributed samples, still using notation from original paper
9. Describe general application of HPP against Hawk using Hawk notation and conventions (e.g. column vectors instead of row vectors, matrix B instead of V, etc.)
10. Describe measuring of DGD properties, and implementation of this
11. Detailed description of attack in practice, discuss implementation challenges w.r.t. memory, runtime, etc.
12. Results and discussion of these, limitations, considerations, etc.

Chapter 2

Background

In this chapter, the field of cryptology will be introduced, with an emphasis on digital signatures and cryptanalysis. We also introduce some necessary facts and notions related to linear algebra and lattices, as well as probability theory and distributions. Lastly, we introduce the notion of *Gradient Search* and ADAM-optimizers, which will be a central tool in this thesis.

2.1 Cryptology

For this section, [4] will be used.

2.1.1 Cryptography

2.1.2 Cryptanalysis

2.2 Digital Signatures

2.2.1 Hash-and-Sign

2.2.2 GGH

2.2.3 NTRU

2.3 Linear Algebra and Lattices

Denote by $\mathbf{v}$ an $n \times 1$ column vector on the form

$$\mathbf{v} = \begin{bmatrix} v_0 \\ v_1 \\ ... \\ v_{n-1} \end{bmatrix}$$

and by $\mathbf{B}$ an $n \times m$ matrix on the form

$$\mathbf{B} = \begin{bmatrix} b_{0,0} & b_{0,1} & \cdots & b_{0,n-1} \\ b_{1,0} & b_{1,1} & \cdots & b_{1,n-1} \\ \cdots & \cdots & \cdots & \cdots \\ b_{m-1,0} & b_{m-1,1} & \cdots & b_{m-1,n-1} \end{bmatrix}$$

Generally, entries v_i and $b_{i,j}$ are integers unless stated otherwise. Some places the thesis will use row notation instead of column notation for the vectors, so that $\mathbf{v}$ is a $1 \times n$ row vector on the form

$$\mathbf{c} = [v_0, v_1, ..., v_{n-1}]$$

In these cases this will be pointed out.

We denote by $\langle \cdot, \cdot \rangle$ the dot-product of two vectors of equal dimensions as $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^t \mathbf{y} = \displaystyle\sum_{i=0}^{n-1} x_i y_i$

2.4 Probability Theory

2.5 Gradient Search

Chapter 3

Hawk and HPP

3.1 Hawk

3.2 Implementation of Hawk

3.3 HPP

In this section we present the *Learning a Parallelepiped* attack as described in [6]. Note that in this section $\mathbf{v}$ denotes a *row*-vector to follow the same notation as in the original paper.

3.3.1 Setup and idealized version

Let $\mathbf{V} \in \mathcal{GL}(\mathbb{R})$ be a secret $n \times n$ unimodular matrix and $\mathcal{P}(\mathbf{V})$ be a fundamental parallelepiped, defined as $\{\mathbf{xV} : \mathbf{x} \in [-1, 1]^n\}$. Let $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t\}$ be t row vectors of length n with entries uniformly distributed over $[-1, 1] \in \mathbb{Q}$ and $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_t\} = \{\mathbf{V}\mathbf{x}_1, \mathbf{V}\mathbf{x}_2, ..., \mathbf{V}\mathbf{x}_t\}$ such that $\mathbf{v} \subset \mathcal{P}(\mathbf{V})$. By observing $\mathbf{v}$ for large enough t, one is able to retrieve the rows of $\pm\mathbf{V}$ by the following steps:

1. Estimate covariance matrix $\mathbf{V}^t\mathbf{V}$
2. Transform samples $\mathbf{v} \in \mathcal{P}(\mathbf{V})$ to $\mathbf{c} \in \mathcal{P}(\mathbf{C})$ where $\mathcal{P}(\mathbf{C})$ is a hypercube, i.e. $\mathbf{CC}^t = \mathbf{I}$
3. Do gradient descent to minimize the fourth moment of one-dimensional projections and reveal a row of $\pm\mathbf{C}$ which finally can be transformed into a row of $\pm\mathbf{V}$

In the following, each of these steps will be covered in detail.

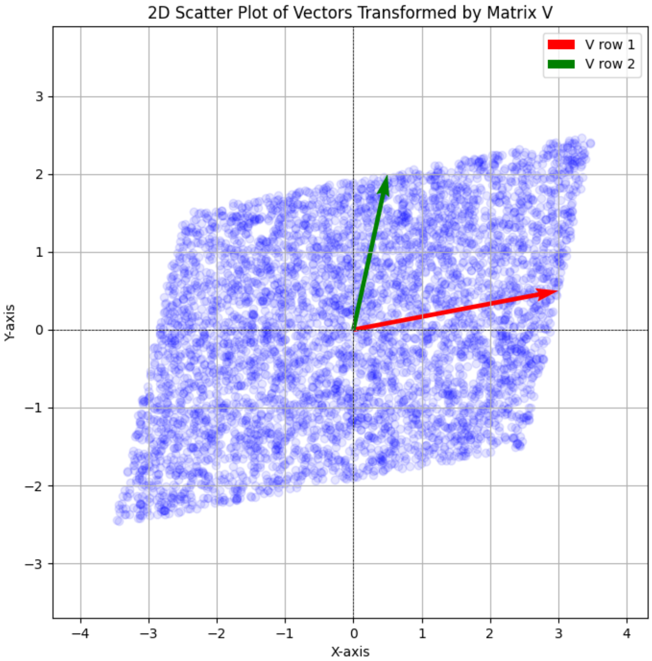

Figure 3.1: Hidden parallelepiped problem in dimension 2

3.3.2 Covariance matrix estimation

Given enough samples on the form $\mathbf{v} = \mathbf{V}\mathbf{x}$, we want to estimate the covariance matrix $\mathbf{G} \approx \mathbf{V}^t\mathbf{V}$. This is achieved as $\mathbf{v}^t\mathbf{v} = (\mathbf{V}\mathbf{x})^t(\mathbf{V}\mathbf{x}) = \mathbf{V}^t\mathbf{x}^t\mathbf{x}\mathbf{V}$. Now, by taking the expectation of the inner term we get $\mathbb{E}[\mathbf{x}^t\mathbf{x}] = \mathbf{I}/3$ where $\mathbf{I}$ is the identity matrix because of the following: Since all $x_i \in \mathbf{x}$ are distributed according to the uniform distribution over $[-1, 1]$ and each x_i are independent, we have that $\mathbb{E}[x_i] = 0$ and $\mathbb{E}[x_i x_j] = \mathbb{E}[x_i]\mathbb{E}[x_j] = 0$ when $i \neq j$. For the case when $i = j$, we have

$$\mathbb{E}[x_i x_j] = \mathbb{E}[x^2] = \int_a^b x^2 \frac{1}{b-a} dx = \int_{-1}^1 x^2 \frac{1}{2} dx = \frac{1}{3}$$

Therefore, $\mathbb{E}[\mathbf{x}^t\mathbf{x}]$ is $\frac{1}{3}$ down the diagonal and is 0 otherwise, i.e. $\mathbf{I}/3$. Thus, as number of samples grow, $\mathbf{v}^t\mathbf{v} \to \mathbf{V}^t(\mathbf{I}/3)\mathbf{V}$, and therefore $\mathbf{v}^t\mathbf{v} \cdot 3 \to \mathbf{V}^t\mathbf{V}$. In conclusion: by taking the average of $\mathbf{v}^t\mathbf{v}$ for all collected samples, and multiplying the resulting $n \times n$ matrix with 3, one has a good approximation of the covariance matrix $\mathbf{V}^t\mathbf{V}$.

3.3.3 Hidden parallelepiped to hidden hypercube transformation

Given a good approximation $\mathbf{G} \approx \mathbf{V}^t\mathbf{V}$, the next step is to calculate a linear transformation $\mathbf{L}$ such that the following is true:

1. $\mathbf{C} = \mathbf{VL}$ is orthonormal, i.e. the rows are pairwise orthogonal and the norm of each row is 1. In other words, $\mathbf{CC}^t = \mathbf{I}$. Consequently, $\mathcal{P}(\mathbf{C})$ becomes a hypercube.

2. If $\mathbf{v}$ is uniformly distributed over $\mathcal{P}(\mathbf{V})$ then $\mathbf{c} = \mathbf{vL}$ is uniformly distributed over $\mathcal{P}(\mathbf{C})$.

This is achieved by taking the Cholesky decomposition of $\mathbf{G}^{-1} = \mathbf{LL}^t$ where $\mathbf{L}$ is a lower-triangular matrix. To compute Cholesky decomposition of $\mathbf{G}^{-1}$, we must first show that $\mathbf{G}$ is symmetric positive definite.

1. $\mathbf{G}$ is symmetric $\iff$ $\mathbf{G}^t = \mathbf{G}$ which is clear as $\mathbf{G}^t = (\mathbf{V}^t\mathbf{V})^t = \mathbf{V}^t(\mathbf{V}^t)^t = \mathbf{V}^t\mathbf{V} = \mathbf{G}$.

2. $\mathbf{G}$ is positive definite if for any non-zero column vector $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^t\mathbf{G}\mathbf{x} > 0$. We have that $\mathbf{x}^t\mathbf{G}\mathbf{x} = \mathbf{x}^t\mathbf{V}^t\mathbf{V}\mathbf{x} = (\mathbf{V}\mathbf{x})^t(\mathbf{V}\mathbf{x})$. Denote by $\mathbf{y} = \mathbf{V}\mathbf{x}$. Since $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{V}$ is invertible (and therefore non-zero) it is clear that $\mathbf{y} \neq \mathbf{0}$ and $\mathbf{y}^t\mathbf{y} = ||\mathbf{y}||^2 > 0$.

From this, we show the following:

1. If $\mathbf{C} = \mathbf{VL}$, then $\mathbf{CC}^t = \mathbf{VLL}^t\mathbf{V}^t = \mathbf{VG}^{-1}\mathbf{V}^t = \mathbf{V}(\mathbf{V}^t\mathbf{V})^{-1}\mathbf{V}^t = \mathbf{VV}^{-1}\mathbf{V}^{-t}\mathbf{V}^t = \mathbf{I}$.

2. Since entries in $\mathbf{x}$ is uniformly distributed, $\mathbf{c} = \mathbf{Cx}$ is uniformly distributed over $\mathcal{P}(\mathbf{C})$.

By multiplying our samples $\mathbf{v}$ by $\mathbf{L}$ on the right, we transform them from the hidden parallelepiped to the hidden hypercube. If one finds the rows of $\pm\mathbf{C}$, one can simply multiply the result on the right by $\mathbf{L}^{-1}$ to obtain the solution for $\mathbf{V}$.

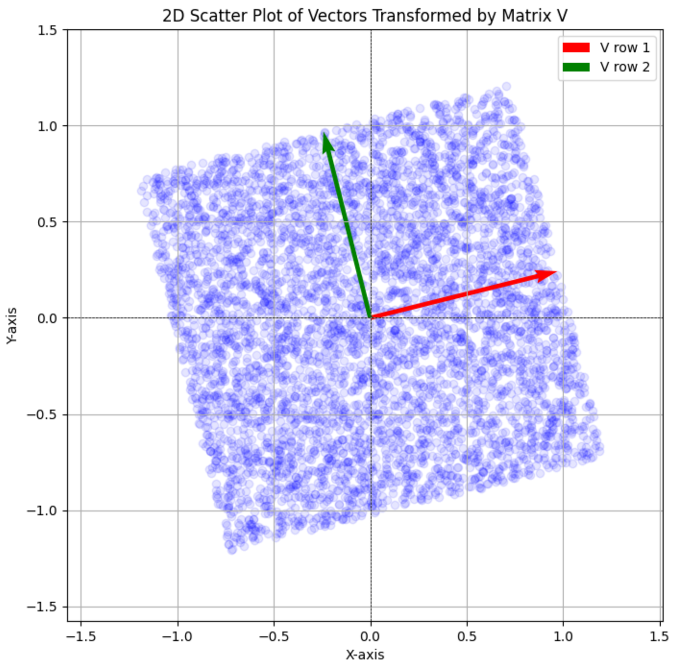

Figure 3.2: Hidden hypercube problem in dimension 2

3.3.4 Moments and Gradient Descent

The last major step in the attack is to measure and minimize the fourth moment of one-dimensional projections to disclose rows of $\pm\mathbf{C}$. Let $mom_{k,\mathbf{C}}(\mathbf{w})$ be defined as the k-th moment of $\mathcal{P}(\mathbf{C})$ projected onto $\mathbf{w}$, i.e. $\mathbb{E}[\langle \mathbf{c}, \mathbf{w}\rangle^k]$ where $\mathbf{c} = \mathbf{xC}$ for $\mathbf{x}$ uniformly distributed and $\mathbf{w} \in \mathbb{R}^n$. Looking at the term $\langle \mathbf{c}, \mathbf{w}\rangle$, we have $\langle \mathbf{xC}, \mathbf{w}\rangle = \langle \sum_{i=1}^n x_i c_i, \mathbf{w}\rangle$ where c_i is the i-th row of $\mathbf{C}$. Since x_i is a scalar, we can move it out of the dot-product brackets. $\sum_{i=1}^n x_i \langle c_i, \mathbf{w}\rangle$. Moving this inside the $\mathbb{E}[\ ]$ we have $\mathbb{E}[\sum_{i=1}^n x_i \langle c_i, \mathbf{w}\rangle^k]$. We look at the two cases when $k = 2$ and $k = 4$.

- $\mathbf{k = 2}$: $\mathbb{E}[(\sum_{i=1}^n x_i \langle c_i, \mathbf{w}\rangle)^2] = \mathbb{E}[\sum_i^n \sum_j^n x_i x_j \langle c_i, \mathbf{w}\rangle \langle c_j, \mathbf{w}\rangle]$. As seen before in section 3.3.2, $\mathbb{E}[x_i x_j] = \frac{1}{3}$ when $i = j$ and 0 otherwise. Thus, we have the final expression $mom_{2,C}(\mathbf{w}) = \frac{1}{3}\sum_i^n \langle c_i, \mathbf{w}\rangle^2$ which can also be written as $= \frac{1}{3}\mathbf{w}\mathbf{C}^t\mathbf{C}\mathbf{w}^t$
- $\mathbf{k = 4}$:

$$\mathbb{E}[(\sum_{i=1}^n x_i \langle c_i, \mathbf{w}\rangle)^4] = \mathbb{E}[\sum_i^n \sum_j^n \sum_k^n \sum_l^n x_i x_j x_k x_l \langle c_i, \mathbf{w}\rangle \langle c_j, \mathbf{w}\rangle \langle c_k, \mathbf{w}\rangle \langle c_l, \mathbf{w}\rangle]$$

There are three cases for the indices i, j, k and l:

1. **All equal:** If $i = j = k = l$, we simply have $\sum_i^n \mathbb{E}[x^4]\langle c_i, \mathbf{w}\rangle^4 = \frac{1}{5}\sum_i \langle c_i, \mathbf{w}\rangle^4$ due to the fact that $\mathbb{E}[x^4] = \int_{-1}^{1} x^4 \frac{1}{2} dx = \frac{1}{5}$
2. **None equal:** If $i \neq j \neq k \neq l$ the expression is zero due to $\mathbb{E}[x_i] = 0$ and all x_i, x_j, x_k, x_l are independent.
3. **Pairwise equal:** If either
   - $i = j \neq k = l$
   - $i = k \neq j = l$
   - $i = l \neq j = k$

   then we have $\sum_{i\neq j} \mathbb{E}[x_i^2 x_j^2]\langle c_i, \mathbf{w}\rangle^2\langle c_j, \mathbf{w}\rangle^2 = \frac{1}{9}\sum_{i\neq j}\langle c_i, \mathbf{w}\rangle^2\langle c_j, \mathbf{w}\rangle^2$. By putting the above together we get

   $$\frac{1}{5}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4 + 3(\frac{1}{9}\sum_{i\neq j}\langle c_i, \mathbf{w}\rangle^2\langle c_j, \mathbf{w}\rangle^2)$$

   and the final expression becomes

   $$mom_{4,\mathbf{C}}(\mathbf{w}) = \frac{1}{5}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4 + \frac{1}{3}\sum_{i\neq j}\langle c_i, \mathbf{w}\rangle^2\langle c_j, \mathbf{w}\rangle^2$$

Now, since $\mathbf{C}$ is orthonormal, and by restricting $\mathbf{w}$ to the unit sphere in $\mathbb{R}^n$, we can simplify the expressions further. The second moment becomes $mom_{2,C}(\mathbf{w}) = \frac{1}{3}\mathbf{w}\mathbf{C}^t\mathbf{C}\mathbf{w}^t = \frac{1}{3}\mathbf{w}\mathbf{I}\mathbf{w}^t = \frac{1}{3}\mathbf{w}\mathbf{w}^t = \frac{1}{3}\|\mathbf{w}\|^2 = \frac{1}{3}$.
By rewriting and expanding the fourth moment:

$$mom_{4,\mathbf{C}}(\mathbf{w}) = \frac{1}{5}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4 + \frac{1}{3}\sum_{i\neq j}\langle c_i, \mathbf{w}\rangle^2\langle c_j, \mathbf{w}\rangle^2$$

$$mom_{4,\mathbf{C}}(\mathbf{w}) = \frac{1}{5}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4 + \frac{1}{3}\sum_{i=1}^{n}\sum_{j=1}^{n}\langle c_i, \mathbf{w}\rangle^2\langle c_j, \mathbf{w}\rangle^2 - \frac{1}{3}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^2\langle c_i, \mathbf{w}\rangle^2$$

$$mom_{4,\mathbf{C}}(\mathbf{w}) = \frac{1}{5}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4 + \frac{1}{3}\sum_{i=1}^{n}\sum_{j=1}^{n}\langle c_i, \mathbf{w}\rangle^2\langle c_j, \mathbf{w}\rangle^2 - \frac{1}{3}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4$$

$$mom_{4,\mathbf{C}}(\mathbf{w}) = \frac{1}{3}\|\mathbf{w}\|^4 - \frac{2}{15}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4 = \frac{1}{3} - \frac{2}{15}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4$$

We also need to compute the gradient of the fourth moment, $\nabla mom_{4,\mathbf{C}}(\mathbf{w})$.

- The gradient of the first term, $\frac{1}{3}\|\mathbf{w}\|^4$, is computed as follows:
  We rewrite $\|\mathbf{w}\|^4$ as $(\mathbf{w}\mathbf{w}^t)^2$. Using the chain rule we have that
  $\nabla((\mathbf{w}\mathbf{w}^t)^2) = 2(\mathbf{w}\mathbf{w}^t) \cdot \frac{\partial}{\partial \mathbf{w}_j}(\mathbf{w}\mathbf{w}^t) = 2(\mathbf{w}\mathbf{w}^t) \cdot 2\mathbf{w}$.
  The gradient of the first term is then $\frac{1}{3} \cdot 2(\mathbf{w}\mathbf{w}^t) \cdot 2\mathbf{w} = \frac{4}{3}\|\mathbf{w}\|^2\mathbf{w}$.
- For the second term, $\frac{2}{15}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4$ we have the following:

$$\nabla \sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4 = \sum_{i=1}^{n}\nabla(\langle c_i, \mathbf{w}\rangle^4)$$

  Looking at just the inner term, $\nabla(\langle c_i, \mathbf{w}\rangle^4) = 4\langle c_i, \mathbf{w}\rangle^3 \cdot \frac{\partial}{\partial w_j}(\langle c_i, \mathbf{w}\rangle) = 4\langle c_i, \mathbf{w}\rangle^3 \cdot c_i$
  The gradient of the second term is therefore $\displaystyle\frac{8}{15}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^3 c_i$

Putting together the terms we get

$$\nabla mom_{4,\mathbf{C}}(\mathbf{w}) = \frac{4}{3}\|\mathbf{w}\|^2\mathbf{w} - \frac{8}{15}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^3 c_i$$

The key observation is that by minimizing $mom_{4,\mathbf{C}(\mathbf{w})}$ one has to maximize the term $\displaystyle -\frac{2}{15}\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4$. Since c_i and $\mathbf{w}$ are both on the unit circle, and all c_i are orthogonal to each other, the term is maximized whenever $\mathbf{w} = c_i$ since $\langle c_i, \pm c_i\rangle = \langle \mathbf{w}, \pm\mathbf{w}\rangle = 1$. Using this observation, we can run a gradient descent to minimize $mom_{4,\mathbf{C}}(\mathbf{w})$ and find rows of $\pm\mathbf{C}$. A simple gradient descent is described in the following algorithm:

Algorithm 1 Gradient descent

Require: Parameter δ, samples $\mathbf{c}$
1: Choose random vector $\mathbf{w}$ on the unit sphere
2: Compute an approximation $\mathbf{g} \leftarrow \nabla mom_{4,\mathbf{C}}(\mathbf{w})$
3: Set $\mathbf{w}_{new} \leftarrow \mathbf{w} - \delta\mathbf{g}$
4: Normalize $\mathbf{w}_{new}$ as $\frac{\mathbf{w}_{new}}{\|\mathbf{w}_{new}\|}$
5: Approximate $mom_{4,\mathbf{C}}(\mathbf{w})$ and $mom_{4,\mathbf{C}}(\mathbf{w}_{new})$
6: **if** $mom_{4,\mathbf{C}}(\mathbf{w}) \geq mom_{4,\mathbf{C}}(\mathbf{w}_{new})$ **then**
7:     Return $\mathbf{w}$
8: **else**
9:     Set $\mathbf{w} \leftarrow \mathbf{w}_{new}$ and go to step 2

This procedure is repeated until all rows of $\pm\mathbf{C}$ is found. Each row is transformed by $\mathbf{L}^{-1}$ to get the row in $\mathbf{V}$.

3.4 HPP against the Normal Distribution

Assume now that the signatures on the form $\mathbf{v} = \mathbf{x}\mathbf{V}$ where $x_i \sim \mathcal{N}(0, \sigma^2)$. After converting $\mathcal{P}(\mathbf{V})$ to $\mathcal{P}(\mathbf{C})$, the fourth moment of $\mathcal{P}(\mathbf{C})$ is constant over $\mathbf{w}$ on the unit circle. To show this, we simply do the same calculations as in the previous section, with the difference that x follows a normal distribution. Considering

$$mom_{4,\mathbf{C}}(\mathbf{w}) = \mathbb{E}[(\sum_{i=1}^{n} x_i \langle c_i, \mathbf{w} \rangle)^4] = \mathbb{E}[\sum_{i}^{n}\sum_{j}^{n}\sum_{k}^{n}\sum_{l}^{n} x_i x_j x_k x_l \langle c_i, \mathbf{w}\rangle \langle c_j, \mathbf{w}\rangle \langle c_k, \mathbf{w}\rangle \langle c_l, \mathbf{w}\rangle]$$

for the three different cases:

- **All equal:** If $i = j = k = l$, we simply have $\sum_i^n \mathbb{E}[x_i^4]\langle c_i, \mathbf{w}\rangle^4 = 3\sigma^4 \sum_i \langle c_i, \mathbf{w}\rangle^4$ due to the well known fact that $\mathbb{E}[x^4] = 3\sigma^4$ for x distributed according to $\mathcal{N}(0, \sigma^2)$.
- **None equal:** Since $\mathbb{E}[x] = 0$ as in the case of the uniform distribution, this results in zero.
- **Pairwise equal:** If either
  - $i = j \neq k = l$
  - $i = k \neq j = l$
  - $i = l \neq j = k$

  we have $\sum_{i \neq j} \mathbb{E}[x_i^2 x_j^2]\langle c_i, \mathbf{w}\rangle^2 \langle c_j, \mathbf{w}\rangle^2 = \sigma^4 \sum_{i \neq j} \langle c_i, \mathbf{w}\rangle^2 \langle c_j, \mathbf{w}\rangle^2$ since $\mathbb{E}[x^2] = \sigma^2$.

Putting together expressions we get

$$mom_{4,\mathbf{C}}(\mathbf{w}) = 3\sigma^4 \sum_i \langle c_i, \mathbf{w}\rangle^4 + 3(\sigma^4 \sum_{i \neq j} \langle c_i, \mathbf{w}\rangle^2 \langle c_j, \mathbf{w}\rangle^2)$$

$$mom_{4,\mathbf{C}}(\mathbf{w}) = 3\sigma^4 \sum_i \langle c_i, \mathbf{w}\rangle^4 + 3\sigma^4 \sum_i \sum_j \langle c_i, \mathbf{w}\rangle^2 \langle c_j, \mathbf{w}\rangle^2 - 3\sigma^4 \sum_i \langle c_i, \mathbf{w}\rangle^4$$

$$mom_{4,\mathbf{C}}(\mathbf{w}) = 3\sigma^4 \sum_i \sum_j \langle c_i, \mathbf{w}\rangle^2 \langle c_j, \mathbf{w}\rangle^2$$

$$mom_{4,\mathbf{C}}(\mathbf{w}) = 3\sigma^4 \sum_i \sum_j \langle c_i, \mathbf{w} \rangle^2 \langle c_j, \mathbf{w} \rangle^2$$

$$mom_{4,\mathbf{C}}(\mathbf{w}) = 3\sigma^4 (\|\mathbf{w}\|^2)^2$$

Since $mom_{4,\mathbf{C}}(\mathbf{w})$ is constant for $\mathbf{w}$ on the unit sphere regardless of σ^2, the term no longer depends on the secret matrix $\mathbf{C}$ and the attack will not work.

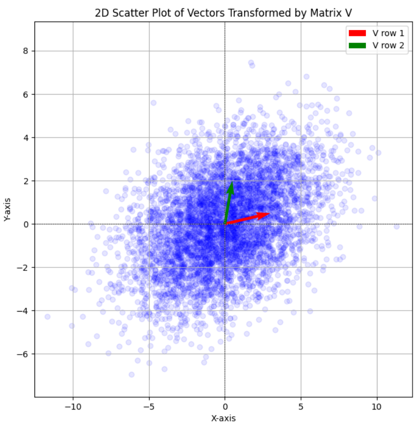

Figure 3.3: Hidden parallelepiped problem in dimension 2 for normal distribution

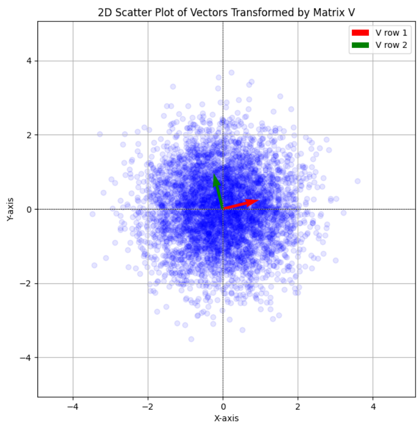

Figure 3.4: Hidden hypercube problem in dimension 2 for normal distribution

Chapter 4

Cryptanalysis of Hawk

In this chapter we perform the cryptanalysis of Hawk. Note that we now switch notation so that $\mathbf{v}$ denotes a column vector to align with the notation and conventions of [3].

4.1 Overview

The original HPP attack can not work if the vector $\mathbf{x}$ multiplied with secret $\mathbf{V}$ has normally distributed entries as shown in section 3.4. In Hawk, the distribution of entries of $\mathbf{x}$ is the Discrete Gaussian Distribution (DGD). As the name implies, this distribution is discrete, not continuous. Instead of showing theoretical and asymptotic results for the DGD, we use our implementation of Hawk to measure and estimate the properties of the distribution. The belief is that the discretization of the normal distribution in this manner makes the result in section 3.4 not hold in practice. Consequently, by applying the HPP attack on Hawk signatures one might be able to disclose the secret key.

The attack can be summarized as follows:

Algorithm 2 HPP Hawk

1: Collect signatures $\mathbf{w} = \mathbf{B}^{-1}\mathbf{x}$
2: Using public key $\mathbf{Q}$, find $\mathbf{L}$ s.t. $\mathbf{Q} = \mathbf{L}\mathbf{L}^t$
3: Transform samples s.t. $\mathbf{c} = \mathbf{L}^t\mathbf{w}$
4: Find columns of $\pm\mathbf{C}$ by doing gradient search over $\mathcal{P}(\mathbf{C})$
5: Multiply columns of $\pm\mathbf{C}$ by $\mathbf{L}^{-t}$ on the left to get columns in $\pm\mathbf{B}^{-1}$

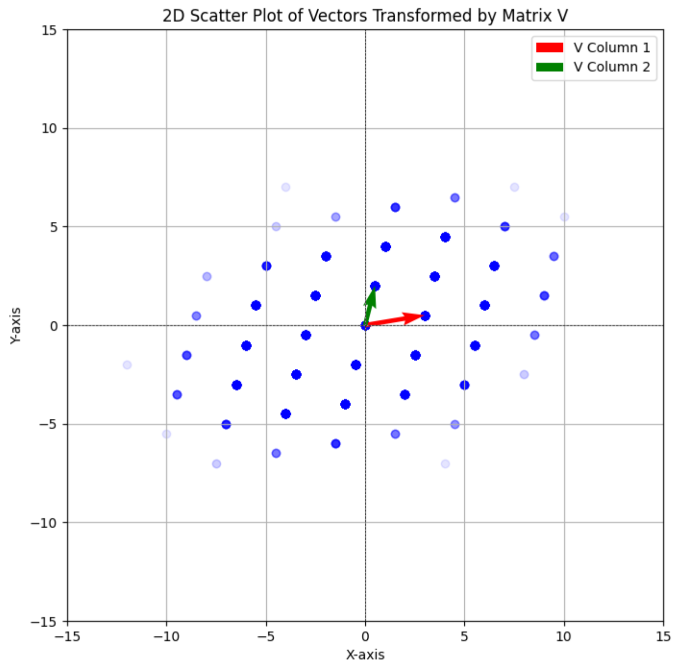

Figure 4.1: Hidden parallelepiped problem in dimension 2 for rounded normal distribution

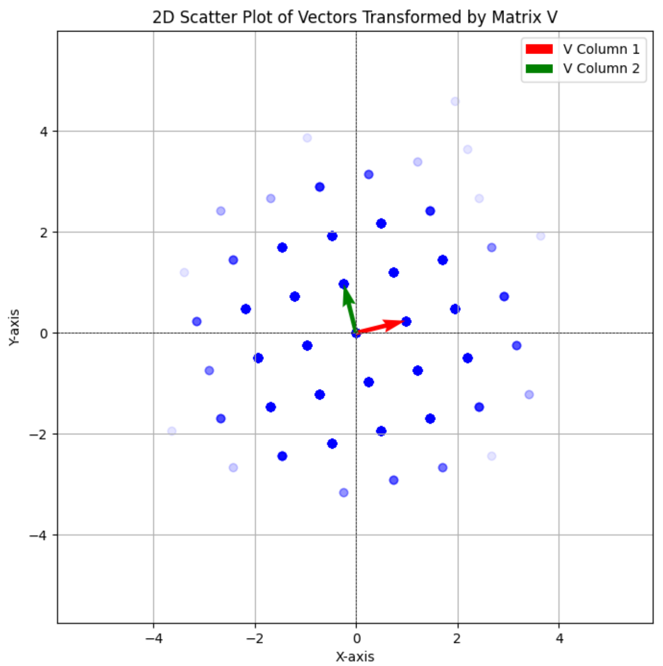

Figure 4.2: Hidden hypercube problem in dimension 2 for rounded normal distribution

4.2 HPP against practical Discrete Gaussian Distribution

4.2.1 Overview of method

Consider the Discrete Gaussian Distribution as described in [3] and in section 2.4... We use our implementation of Hawk to sample many points from the practical distribution. Let $\mathcal{D}$ denote the theoretical discrete Gaussian distribution, and let $\widehat{\mathcal{D}}$ denote the practical discrete Gaussian distribution from sampled points. Let 0, σ^2 be the expectation and variance of $\mathcal{D}$, and $\hat{\mu}$, $\hat{\sigma}^2$ be the expectation and variance of $\widehat{\mathcal{D}}$. Assume we sample t points from $\widehat{\mathcal{D}}$ as $X = \{x_1, x_2, ..., x_t\}$. We estimate $\hat{\mu}$ and $\hat{\sigma}^2$ simply as $\hat{\mu} = \dfrac{1}{t}\sum_{i=1}^{t} x_i$ and $\hat{\sigma}^2 = \dfrac{1}{t}\sum_{i=1}^{t}(x_i - \hat{\mu})^2$. For simplicity, we can also assume $\hat{\mu} = \mu = 0$ as claimed in [3]. To simplify later computations we also normalize our samples by computing $Z = \{z_1, z_2, ..., z_t\} = \{\frac{x_1}{\hat{\sigma}}, \frac{x_2}{\hat{\sigma}}, ..., \frac{x_t}{\hat{\sigma}}\}$ such that $\mathbb{V}[z_i] = 1$.

Now, denote by $\mu_4 = \mathbb{E}[z_i^4] = \dfrac{1}{t}\sum_{i=1}^{n} z_i^4$. Assume observed signatures on the form $\mathbf{c} = \mathbf{C}\mathbf{z}$. By rewriting the terms from section 3.4 for this new, normalized, distribution $\widehat{\mathcal{D}}$, we have that

$$mom_{4,\mathbf{C}}(\mathbf{w}) = 3\|\mathbf{w}\|^4 + (\mu_4 - 3)\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4$$

and

$$\nabla mom_{4,\mathbf{C}}(\mathbf{w}) = 12\|\mathbf{w}\|^2\mathbf{w} + 4(\mu_4 - 3)\sum_{i=1}^{n}\langle c_i, \mathbf{w}^3\rangle c_i$$

Maybe show more computations here

This means that if the difference $(\mu_4 - 3)$ is big enough, one might be able to employ the same minimization technique as in the original attack to reveal a column of $\mathbf{V}$. Note that if $(\mu_4 - 3) < 0$ we have the same case as in the original attack, where minimization of the entire term entails maximization of $\sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4$, which gives us a row of $\pm\mathbf{C}$. If $(\mu_4 - 3) > 0$, we need to maximize the entire term $3\|\mathbf{w}\|^4 + \sum_{i=1}^{n}\langle c_i, \mathbf{w}\rangle^4$, which is achieved by doing a gradient *ascent* instead of a gradient *descent*.

4.2.2 Covariance matrix and hypercube transformation

In the original HPP attack one has to estimate the matrix $\mathbf{G} \approx \mathbf{V}^t\mathbf{V}$ as $\mathbf{v}^t\mathbf{v} \cdot 3$. We show that this is possible even if x is normally distributed, as one can estimate $\frac{\mathbf{v}^t\mathbf{v}}{\sigma^2}$. For Hawk, the signatures are on the form $\mathbf{w} = \mathbf{B}^{-1}\mathbf{x}$. Then we would need to compute $\mathbf{G} = \mathbf{B}^{-1}\mathbf{B}^{-t} \approx \frac{\mathbf{w}\mathbf{w}^t}{\sigma^2}$. In Hawk, however, the public key $\mathbf{Q} = \mathbf{B}^*\mathbf{B}$ which for columns $\mathbf{b} \in \mathbb{Q}^n$ is equivalent to $\mathbf{B}^t\mathbf{B}$, enables us to skip this step. Recall that in the original attack one has to take Cholesky decomposition (or an equivalent decomposition) of the inverse of the covariance matrix such that $\mathbf{G}^{-1} = \mathbf{L}\mathbf{L}^t$. For $\mathbf{G} = \mathbf{B}^{-1}\mathbf{B}^{-t}$, the inverse of $\mathbf{G}$, $\mathbf{G}^{-1} = \mathbf{B}^t\mathbf{B} = \mathbf{Q}$. Therefore, we can simply take the Cholesky decomposition of $\mathbf{Q} = \mathbf{L}\mathbf{L}^t$. By multiplying our samples $\mathbf{w}$ by $\mathbf{L}^t$ on the left, we have transformed our samples to the hidden hypercube as in the original attack.
By taking $\mathbf{C} = \mathbf{L}^t\mathbf{B}^{-1}$, we have that

$$\mathbf{C}^t\mathbf{C} = (\mathbf{L}^t\mathbf{B}^{-1})^t(\mathbf{L}^t\mathbf{B}^{-1}) = \mathbf{B}^{-t}\mathbf{L}\mathbf{L}^t\mathbf{B}^{-1} = \mathbf{B}^{-t}\mathbf{Q}\mathbf{B}^{-1} = \mathbf{B}^{-t}\mathbf{B}^t\mathbf{B}\mathbf{B}^{-1} = \mathbf{I}$$

and

$$\mathbf{C}\mathbf{C}^t = (\mathbf{L}^t\mathbf{B}^{-1})(\mathbf{L}^t\mathbf{B}^{-1})^t = \mathbf{L}^t\mathbf{B}^{-1}\mathbf{B}^{-t}\mathbf{L} = \mathbf{L}^t\mathbf{Q}^{-1}\mathbf{L} = \mathbf{L}^t(\mathbf{L}\mathbf{L}^t)^{-1}\mathbf{L} = \mathbf{L}^t\mathbf{L}^{-t}\mathbf{L}^{-1}\mathbf{L} = \mathbf{I}$$

Since $\mathbf{x}$ is distributed according to $\widehat{\mathcal{D}}$ over $\mathcal{P}(\mathbf{B^{-1}})$, by taking $\mathbf{c} = \mathbf{L}^t\mathbf{w}$ we have $\mathbf{c} = \mathbf{L}^t\mathbf{B}^{-1}\mathbf{x} = \mathbf{C}\mathbf{x}$, $\mathbf{c}$ is distributed according to $\widehat{\mathcal{D}}$ over $\mathcal{P}(\mathbf{C})$.

We summarize this step of the attack against Hawk in the following algorithm

Algorithm 3 Hawk Hypercube Transformation

Require: Samples $\mathbf{w} = \mathbf{B}^{-1}\mathbf{x}$ and public key $\mathbf{Q}$
1: Compute $\mathbf{L}$ s.t. $\mathbf{Q} = \mathbf{L}\mathbf{L}^t$ ▷ by Cholesky decomposition
2: Compute $\mathbf{c} = \mathbf{L}^t\mathbf{w}$
3: **return** $\mathbf{c}$ and $\mathbf{L}^{-t}$

4.2.3 Gradient search overview

After having transformed the samples $\mathbf{w} \in \mathcal{P}(\mathbf{B^{-1}})$ to $\mathbf{c} \in \mathcal{P}(\mathbf{C})$ we want to recover columns of $\pm\mathbf{C}$ and transform them back to columns of $\pm\mathbf{B}^{-1}$ by multiplying by $\mathbf{L}^{-1}$ on the left. Due to the special structure of Hawk (and NTRU) private keys, finding one column of $\mathbf{B}$, one has automatically found n columns. Unfortunately, since revealing a single column of $\mathbf{B}^{-1}$ reveals a rotation of either the two polynomials G and g *or* F and

f, this is not enough to disclose the entire matrix. If samples were on the form $\mathbf{w} = \mathbf{Bx}$, a single column would reveal f and g, and one could simply reconstruct F and G by solving the NTRU-equation as in the key generation step of Hawk. Nevertheless, if one finds two columns of $\mathbf{B}^{-1}$, it is easy to check if they are shifts of each other. If they are not, one has found shifts of all four polynomials in the secret key, and by trying all combinations of shifts, of which there are $4n^2$ (accounting for negative and positive sign), one can easily verify if a candidate $\mathbf{B'}^{-1}$ is valid by computing $\mathbf{B'} = (\mathbf{B'}^{-1})^{-1}$, and checking if $\mathbf{B'}^t\mathbf{B'} = \mathbf{Q}$. If so, one is able to forge signatures, and the attack is done.

Now, given samples $\mathbf{c} \in \mathcal{P}(\mathbf{C})$, we run a gradient search to minimize or maximize the fourth moment of one-dimensional projections, as in the original attack. In addition to multiplying the samples by $\mathbf{L}^t$, we also divide them by the scalar σ, to normalize the samples so each sample in the vector $\mathbf{c}$ has variance 1. This also aligns with our theoretical analysis of $mom_{4,\mathbf{C}}(\mathbf{w})$. Even though the distribution $\widehat{\mathcal{D}}$ is discrete, the shape of the samples $\mathbf{c} \in \mathcal{P}(\mathbf{C})$ will still have a spherical shape as illustrated in 4.2. The hope is that the areas in the directions of the columns of $\pm\mathbf{C}$ will deviate just enough from a perfect spherical shape to give us a global minima/maxima in the gradient landscape.

4.2.4 Gradient search for Hawk

Due to the noisy and unpredictable gradient landscape we need to search, the *vanilla* gradient search method as used in the original HPP attack is not very good. A constant stepsize means either a slow convergence (with a small value for δ), or risking overshooting the correct extremum (with a big value for δ). Therefore, we employ a more advanced method for gradient search, namely the ADAM-optimizer [5].

Acronyms

CVP Closest Vector Problem.

DGD Discrete Gaussian Distribution.

DSA Digital Signature Algorithm.

HPP Hidden Parallelepiped Problem.

KEMs Key Encapsulation Methods.

NIST National Institute of Standards and Technology.

RSA Rivest, Shamir, Adleman.

Bibliography

[1] Léo Ducas and Phong Q. Nguyen. Learning a zonotope and more: cryptanalysis of ntrusign countermeasures. In *Proceedings of the 18th International Conference on The Theory and Application of Cryptology and Information Security*, ASIACRYPT'12, page 433–450, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 9783642349607. doi: 10.1007/978-3-642-34961-4_27.
URL: `https://doi.org/10.1007/978-3-642-34961-4_27`.

[2] Jill Pipher Joseph H. Silverman-William Whyte Jeffrey Hoffstein, Nick Howgrave-Graham. Ntrusign: Digital signatures using the ntru lattice. Technical report, NTRU Cryptosystems, 2003.

[3] Léo Ducas Serge Fehr Yu-Hsuan Huang Thomas Pornin Eamonn W. Postlethwaite Thomas Prest Ludo N. Pulles Joppe W. Bos, Olivier Bronchain and Wessel van Woerden. Hawk. Technical report, NXP Semiconductors, Centrum Wiskunde & Informatica, Mathematical Institute at Leiden University, NCC Group, PQShield, Institut de Mathématiques de Bordeaux, September 2024.
URL: `https://hawk-sign.info/`.

[4] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography.* CRC Press, Boca Raton, FL, 3 edition, 2020.

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
URL: `https://arxiv.org/abs/1412.6980`.

[6] Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of ggh and ntru signatures, 2009.

Appendix A

Generated code from Protocol buffers

Listing A.1: Source code of something

```
System.out.println("Hello Mars");
```