General Exam

Lattices & the Knapsack Problem

Tashfeen, Ahmad

Computer Science, University of Oklahoma

Oral Portion, Fall 2023

Overview

1. Introduction: Cryptography, $\mathcal{P}$ and $\mathcal{NP}$ complexity classes.
2. Maths & Notation: Define notation and recall maths.
3. Cryptosystem: The Merkle–Hellman Scheme.
4. Cryptanalysis
5. Lattice Reduction: Example
6. Lattice Problems
7. Conclusion & References

Introduction

The scale of theoretical complexity from the easiest to the hardest is arranged as shown in figure 1. Out of curiosity, do you want $\mathcal{P}$ to be equal to $\mathcal{NP}$ or not?

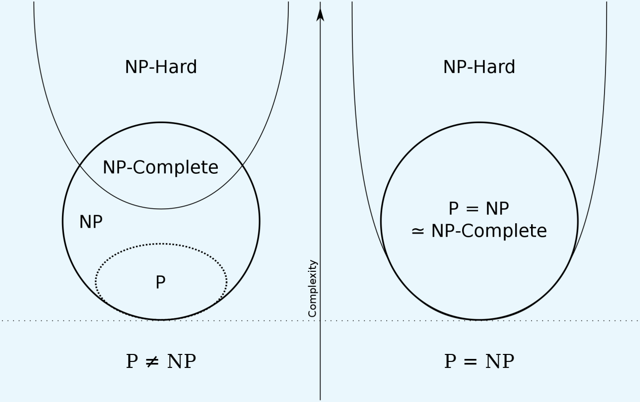

Figure 1: ☺ $\subseteq \mathcal{P} \overset{?}{\subseteq} \mathcal{NP} \subset \mathcal{NP}$-complete $\subset \mathcal{NP}$-hard $\subset$ ☠

Introduction

1. Cryptography operates around $\mathcal{P}$ problems disguised with special information as $\mathcal{NP}$ problems.
2. Someone missing the special information is forced to treat cryptographic problems as $\mathcal{NP}$.
3. We will refer to such $\mathcal{P}$ problems *disguised with special information* as $\mathcal{NP}$-imposter problems. E. g., for Rivest-Shamir-Adleman (RSA),

$$
\begin{aligned}
n &= pq \\
m &\equiv c^{ed \mod \varphi(n)} \mod n \\
d &\equiv e^{-1} \mod \varphi(n) \\
\varphi(n) &= (p-1)(q-1)
\end{aligned}
$$

Special Information

Introduction

1. Prime factorisation is not proven as $\mathcal{NP}$-complete [3].
2. Merkle and Hellman tried to *base* a cryptosystem on a proven $\mathcal{NP}$-complete problem in the 1970s [4] [8].
3. The particular $\mathcal{NP}$-complete problem is known as the Knapsack Problem or the Subset Sum problem.
4. We solve an instance of this problem, each time we make change.

$M' = \{1, 5, 10, 25\}$	Coin Denominations
$S' = 31$	Change
$M' \supset \{1, 5, 25\}$	

Let a and b be two linearly independent vectors spanning $\mathbb{R}^2$. Note that a and b are not orthogonal.

$$a_1 = \text{proj}_b(a) = \left(\frac{a \cdot b}{b \cdot b}\right) b$$

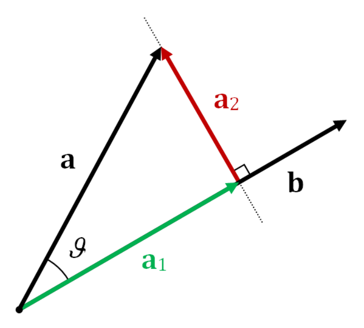

Figure 2: Projection of a onto b, i. e., $a_1 = \text{proj}_b(a)$.

Now, note that a_2 and b are orthogonal.

$$
\begin{aligned}
a_1 + a_2 &= a \\
a_2 &= a - a_1 \\
&= a - \mathrm{proj}_b(a) \\
&= a - \left(\frac{a \cdot b}{b \cdot b}\right) b \\
&= a - \mu b \\
\mu &= \left(\frac{a \cdot b}{b \cdot b}\right)
\end{aligned}
$$

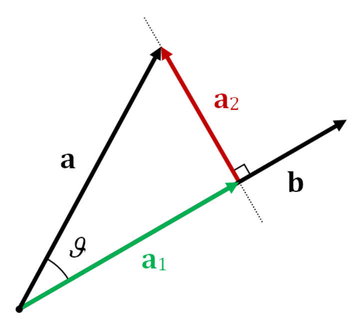

Figure 2: Projection of a onto b, i. e., $a_1 = \mathrm{proj}_b(a)$.

Maths & Notation

The technique of projections to achieve an orthogonal basis is generalised as the *Gram-Schmidt* process [4]. Let $v_1, v_2, v_2, \ldots, v_n$ be a set of linearly independent vectors forming a basis of $\mathbb{R}^n$, then we define the orthogonal basis as $u_1, u_2, u_2, \ldots, u_n$.

Maths & Notation

$u_1 = v_1$

Maths & Notation

$$u_1 = v_1$$
$$u_2 = v_2 - \text{proj}_{u_1}(v_2)$$

Maths & Notation

$$
\begin{aligned}
u_1 &= v_1 \\
u_2 &= v_2 - \mathrm{proj}_{u_1}(v_2) \\
u_3 &= v_3 - \mathrm{proj}_{u_1}(v_3) - \mathrm{proj}_{u_2}(v_3)
\end{aligned}
$$

Maths & Notation

$$
\begin{aligned}
u_1 &= v_1 \\
u_2 &= v_2 - \text{proj}_{u_1}(v_2) \\
u_3 &= v_3 - \text{proj}_{u_1}(v_3) - \text{proj}_{u_2}(v_3) \\
u_4 &= v_4 - \text{proj}_{u_1}(v_4) - \text{proj}_{u_2}(v_4) - \text{proj}_{u_3}(v_4)
\end{aligned}
$$

Maths & Notation

$$
\begin{aligned}
u_1 &= v_1 \\
u_2 &= v_2 - \mathrm{proj}_{u_1}(v_2) \\
u_3 &= v_3 - \mathrm{proj}_{u_1}(v_3) - \mathrm{proj}_{u_2}(v_3) \\
u_4 &= v_4 - \mathrm{proj}_{u_1}(v_4) - \mathrm{proj}_{u_2}(v_4) - \mathrm{proj}_{u_3}(v_4) \\
&\ddots
\end{aligned}
$$

Maths & Notation

$$
\begin{aligned}
u_1 &= v_1 \\
u_2 &= v_2 - \text{proj}_{u_1}(v_2) \\
u_3 &= v_3 - \text{proj}_{u_1}(v_3) - \text{proj}_{u_2}(v_3) \\
u_4 &= v_4 - \text{proj}_{u_1}(v_4) - \text{proj}_{u_2}(v_4) - \text{proj}_{u_3}(v_4) \\
& \ddots \\
u_n &= v_n - \text{proj}_{u_1}(v_n) - \text{proj}_{u_2}(v_n) - \text{proj}_{u_3}(v_n) - \cdots - \text{proj}_{u_{n-1}}(v_n)
\end{aligned}
$$

Maths & Notation

$$
\begin{aligned}
u_1 &= v_1 \\
u_2 &= v_2 - \left(\frac{v_2 \cdot u_1}{u_1 \cdot u_1}\right) u_1 \\
u_3 &= v_3 - \left(\frac{v_3 \cdot u_1}{u_1 \cdot u_1}\right) u_1 - \left(\frac{v_3 \cdot u_2}{u_2 \cdot u_2}\right) u_2 \\
u_4 &= v_4 - \left(\frac{v_4 \cdot u_1}{u_1 \cdot u_1}\right) u_1 - \left(\frac{v_4 \cdot u_2}{u_2 \cdot u_2}\right) u_2 - \left(\frac{v_4 \cdot u_3}{u_3 \cdot u_3}\right) u_3 \\
&\ddots \\
u_n &= v_n - \left(\frac{v_n \cdot u_1}{u_1 \cdot u_1}\right) u_1 - \left(\frac{v_n \cdot u_2}{u_2 \cdot u_2}\right) u_2 - \left(\frac{v_n \cdot u_3}{u_3 \cdot u_3}\right) u_3 - \cdots - \left(\frac{v_n \cdot u_{n-1}}{u_{n-1} \cdot u_{n-1}}\right) u_{n-1}
\end{aligned}
$$

Let $\mu_{i,j} = \left(\frac{v_i \cdot u_j}{u_j \cdot u_j}\right)$ for $i > j$ then,

Maths & Notation

$$
\begin{aligned}
u_1 &= v_1 \\
u_2 &= v_2 - \mu_{2,1} u_1 \\
u_3 &= v_3 - \mu_{3,1} u_1 - \mu_{3,2} u_2 \\
u_4 &= v_4 - \mu_{4,1} u_1 - \mu_{4,2} u_2 - \mu_{4,3} u_3 \\
& \ddots \\
u_n &= v_n - \mu_{n,1} u_1 - \mu_{n,2} u_2 - \mu_{n,3} u_3 - \cdots - \mu_{n,n-1} u_{n-1}
\end{aligned}
$$

Maths & Notation

Let $\boldsymbol{\mu}$ be an n by n lower-triangular *Gram-Schmidt* coefficients matrix defined as,

$$\boldsymbol{\mu} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ \mu_{2,1} & 0 & 0 & \cdots & 0 & 0 \\ \mu_{3,1} & \mu_{3,2} & 0 & \cdots & 0 & 0 \\ \mu_{4,1} & \mu_{4,2} & \mu_{4,3} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mu_{n,1} & \mu_{n,2} & \mu_{n,3} & \cdots & \mu_{n,n-1} & 0 \end{bmatrix}$$

```python
import numpy as np

def proj(u, v):  # projecting v onto u
    mu = (v @ u.T) / (u @ u.T)
    return mu * u, mu

def gram_schmidt(B):
    U, Mu = np.array(B, dtype=B.dtype), np.zeros(shape=B.shape, dtype=B
    .dtype)
    for i in range(1, B.shape[1]):
        for j in range(i):
            projection, Mu[i][j] = proj(U[:, j], B[:, i])
            U[:, i] -= projection
    return U, Mu
```

Listing 1: Vector projection $\text{proj}_u(v)$ and *Gram-Schmidt* orthogonalisation.

Maths & Notation

We define the *knapsack* problem as, for any $M \in \mathbb{N}^n$, $S \in N$ find $x \in \{0, 1\}^n$ such that,

$$M \cdot x = S$$

From the changing making example,

$$M' = [1, 5, 10, 25], x = [1, 1, 0, 1]$$

Then,

$$M' \cdot x = 31 \text{ cents.}$$

Maths & Notation

Euclidean Space Let $\mathbf{B}$ be a basis matrix with d linearly independent column vectors $\{1 \leq i \leq d : v_i \in \mathbb{R}^n\}$. For $d = n$,

$$\mathbb{R}^n = \{x \in \mathbb{R}^d : \mathbf{B}x\}$$

This is trivial, but as sanity check, we inspect the dimensions, $\dim(\mathbf{B}_{(n,d)} x_{(d,1)}) = (n, 1)$.

Maths & Notation

Euclidean Space Let $\mathbf{B}$ be a basis matrix with d linearly independent column vectors $\{1 \leq i \leq d : v_i \in \mathbb{R}^n\}$. For $d = n$,

$$\mathbb{R}^n = \{x \in \mathbb{R}^d : \mathbf{B}x\}$$

This is trivial, but as sanity check, we inspect the dimensions, $\dim(\mathbf{B}_{(n,d)} x_{(d,1)}) = (n, 1)$.

Lattices Restrict $x \in \mathbb{Z}^d$, i. e., $\mathbf{B}x$ to only the integral linear combinations and allow $d \neq n$, we obtain a lattice,

$$\mathcal{L} = \{x \in \mathbb{Z}^d : \mathbf{B}x\}$$

The dimension of the lattice is $\dim(\mathcal{L}) = d$, i. e., the number of vectors in the basis matrix $\mathbf{B}$.

Maths & Notation

Euclidean Space Let $\mathbf{B}$ be a basis matrix with d linearly independent column vectors $\{1 \leq i \leq d : v_i \in \mathbb{R}^n\}$. For $d = n$,

$$\mathbb{R}^n = \{x \in \mathbb{R}^d : \mathbf{B}x\}$$

This is trivial, but as sanity check, we inspect the dimensions, $\dim(\mathbf{B}_{(n,d)} x_{(d,1)}) = (n, 1)$.

Lattices Restrict $x \in \mathbb{Z}^d$, i. e., $\mathbf{B}x$ to only the integral linear combinations and allow $d \neq n$, we obtain a lattice,

$$\mathcal{L} = \{x \in \mathbb{Z}^d : \mathbf{B}x\}$$

The dimension of the lattice is $\dim(\mathcal{L}) = d$, i. e., the number of vectors in the basis matrix $\mathbf{B}$.

Knapsack Lattices Lattices where $\mathbf{B} \in \mathbb{Z}^{n \times d}$ are called the Lagarias-Odlyzko lattices [1].

Maths & Notation

Both **B** and **B'** span the same space, i. e., $\mathbb{R}^2$, but not the same lattice.

$$\mathbf{B} = \begin{bmatrix} 47 & 95 \\ 215 & 460 \end{bmatrix}$$

$$\mathbf{B'} = \begin{bmatrix} 0 & 50 \\ 50 & 0 \end{bmatrix}$$

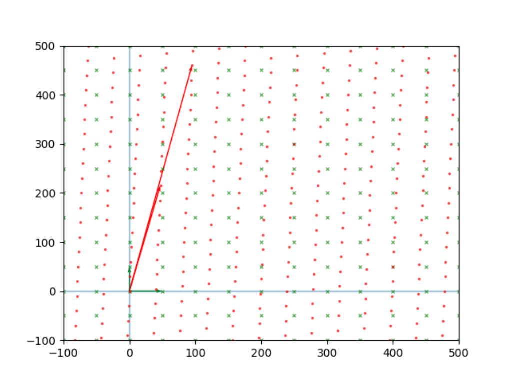

Figure 3: Lattice spanned by **B** and **B'**.

How may we obtain an orthogonal lattice basis? Does one exist?

Cryptosystem

Super Increasing Sequences These are sets $M' = \{r'_1, r'_2, r'_3, \cdots, r'_n\}$ with

$$r'_{i+1} \geq 2r'_i$$

What's an example? You've seen one.

Cryptosystem

Super Increasing Sequences These are sets $M' = \{r'_1, r'_2, r'_3, \cdots, r'_n\}$ with

$$r'_{i+1} \geq 2r'_i$$

What's an example? You've seen one.

Coin Denominations $M' = \{1, 5, 10, 25\}$

```
x = [0 for i in M_]                        # Hoffstein Prop. 7.5 (pg. 379)
for i in range(len(M_)-1, -1, -1): # Loop i from n down to 1
    if S_ >= M_[i]:                        #     If S >= M'[i],
        x[i] = 1                           #         set x[i] = 1 and
        S_ = S_ - M_[i]                    #         subtract M'[i] from S
    else:                                  #     Else
        x[i] = 0                           #         set x[i] = 0
```

Listing 3: Linear time algorithm for *super increasing* sets M_ $= M'$.

Cryptosystem

The Merkle–Hellman scheme [4] is as follows,

Alice	Eve	Bob
Pick $M' = [r'_1, \ldots, r'_n]$, such that $r'_1 > 2^n, r'_{i+1} \geq 2r'_i$. Pick A, B with $B > 2r'_n$ and $\gcd(A, B) = 1$.		

Cryptosystem

The Merkle–Hellman scheme [4] is as follows,

Alice	Eve	Bob
Pick $M' = [r'_1, \ldots, r'_n]$, such that $r'_1 > 2^n, r'_{i+1} \geq 2r'_i$. Pick A, B with $B > 2r'_n$ and $\gcd(A, B) = 1$.		
	M	
Let $r_i \equiv Ar'_i \mod B$ & $M = \{r'_i \in M' : r_i\}$ $\longleftarrow$	$\longrightarrow$	M

The Merkle–Hellman scheme [4] is as follows,

Alice–	Eve	Bob
Pick $M' = [r'_1, \ldots, r'_n]$, such that $r'_1 > 2^n, r'_{i+1} \geq 2r'_i$. Pick A, B with $B > 2r'_n$ and $\gcd(A, B) = 1$.		
	M	
Let $r_i \equiv Ar'_i \mod B$ & $M = \{r'_i \in M' : r_i\}$ ⟵	⟶	M
	S	
S ⟵	⟶	$S = Mx$
(M', S') is $\mathcal{O}(n)$.	(M, S) is $\mathcal{NP}$-imposter.	

Cryptosystem

The Merkle–Hellman scheme [4] is as follows,

Alice–	Eve	Bob
Pick $M' = [r'_1, \ldots, r'_n]$, such that $r'_1 > 2^n, r'_{i+1} \geq 2r'_i$. Pick A, B with $B > 2r'_n$ and $\gcd(A, B) = 1$.		
	M	
Let $r_i \equiv Ar'_i \mod B$ & $M = \{r'_i \in M' : r_i\}$ ⟵	⟶	M
	S	
S ⟵	⟶	$S = Mx$
(M', S') is $\mathcal{O}(n)$. Let $S' \equiv A^{-1}S \mod B$. Solve $(M', S') \longrightarrow x$. We have $M'x = S'$.	(M, S) is $\mathcal{NP}$-imposter.	

Cryptosystem

We know that $M'x = S'$ if and only if $S = Mx$.

$$
\begin{aligned}
S' &\equiv A^{-1}S \mod B \\
&\equiv A^{-1}Mx \mod B && \text{Bob's Encryption } S = Mx \\
&\equiv \sum_{i=1}^{n} A^{-1} r_i x_i \mod B && \text{Since } M = \{r'_i \in M' : r_i\} \\
&\equiv \sum_{i=1}^{n} A^{-1}(A r'_i) x_i \mod B && \text{Since } r_i \equiv A r'_i \\
&\equiv \sum_{i=1}^{n} r'_i x_i \mod B \\
&\equiv M'x \mod B \\
&= M'x && \text{Since } M'x \leq r'_1 + r'_2 + r'_3 +, \ldots, r'_n < 2r'_n < B
\end{aligned}
$$

Cryptanalysis

Proof.

Any general knapsack problem (M, S) can be solved in $\mathcal{O}(2^{\frac{n}{2}})$.

$$M_L = \left\{1 \leq i < \left\lfloor \frac{n}{2} \right\rfloor + 1 : M_i \right\}, \quad M_R = \left\{ \left\lfloor \frac{n}{2} \right\rfloor + 1 \leq i \leq n : M_i \right\}$$

Let $b_j(i) = \left\lfloor \frac{i}{2^j} \right\rfloor \mod 2.$

$$L = \left\{ 0 \leq i < 2^{\left\lfloor \frac{n}{2} \right\rfloor} : x_i = \{0 \leq j \leq \lfloor \lg i \rfloor : b_j(i)\}, \sum_{j=0}^{\lfloor \lg i \rfloor} x_{i,j} M_{L_j} \right\}$$

$$R = \left\{ 0 \leq i < 2^{\left\lceil \frac{n}{2} \right\rceil} : x_i = \{0 \leq j \leq \lfloor \lg i \rfloor : b_j(i)\}, \sum_{j=0}^{\lfloor \lg i \rfloor} x_{i,j} M_{R_j} \right\}$$

Read the written potion.

QUOD ERAT DEM■

Cryptanalysis

We show an example where $M = \{2, 3, 5, 7, 11, 13\}$ and $S = 26$,

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} & 0 \\ & 2 \\ & 3 \\ & 5 \\ & 5 \\ & 7 \\ & 8 \\ \ell \longrightarrow & 10 \end{bmatrix} \star \begin{bmatrix} 0 & \longleftarrow \mathscr{r} \\ 7 & \\ 11 & \\ 13 & \\ 18 & \\ 20 & \\ 24 & \\ 31 & \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 11 \\ 13 \end{bmatrix}$$

$$L_{\ell,1} + R_{\mathscr{r},1} \in \{10, \qquad \}$$

Cryptanalysis

We show an example where $M = \{2, 3, 5, 7, 11, 13\}$ and $S = 26$,

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} & 0 \\ & 2 \\ & 3 \\ & 5 \\ & 5 \\ & 7 \\ & 8 \\ \ell \rightarrow & 10 \end{bmatrix} \star \begin{bmatrix} 0 & \leftarrow r \\ 7 & \leftarrow r \\ 11 & \\ 13 & \\ 18 & \\ 20 & \\ 24 & \\ 31 & \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 11 \\ 13 \end{bmatrix}$$

$$L_{\ell,1} + R_{r,1} \in \{10, 17, \qquad\qquad \}$$

We show an example where $M = \{2, 3, 5, 7, 11, 13\}$ and $S = 26$,

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} & 0 \\ & 2 \\ & 3 \\ & 5 \\ & 5 \\ & 7 \\ & 8 \\ \ell \longrightarrow & 10 \end{bmatrix} \star \begin{bmatrix} 0 & \leftarrow r \\ 7 & \leftarrow r \\ 11 & \leftarrow r \\ 13 & \\ 18 & \\ 20 & \\ 24 & \\ 31 & \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 11 \\ 13 \end{bmatrix}$$

$$L_{\ell,1} + R_{r,1} \in \{10, 17, 21, \qquad \}$$

We show an example where $M = \{2, 3, 5, 7, 11, 13\}$ and $S = 26$,

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} & 0 \\ & 2 \\ & 3 \\ & 5 \\ & 5 \\ & 7 \\ & 8 \\ \ell \longrightarrow & 10 \end{bmatrix} \star \begin{bmatrix} 0 & \leftarrow r \\ 7 & \leftarrow r \\ 11 & \leftarrow r \\ 13 & \leftarrow r \\ 18 & \\ 20 & \\ 24 & \\ 31 & \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 11 \\ 13 \end{bmatrix}$$

$$L_{\ell,1} + R_{r,1} \in \{10, 17, 21, 23, \qquad \}$$

We show an example where $M = \{2, 3, 5, 7, 11, 13\}$ and $S = 26$,

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} & 0 \\ & 2 \\ & 3 \\ & 5 \\ & 5 \\ & 7 \\ & 8 \\ \ell \rightarrow & 10 \end{bmatrix} \star \begin{bmatrix} 0 & \leftarrow r \\ 7 & \leftarrow r \\ 11 & \leftarrow r \\ 13 & \leftarrow r \\ 18 & \leftarrow r \\ 20 & \\ 24 & \\ 31 & \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 11 \\ 13 \end{bmatrix}$$

$$L_{\ell,1} + R_{r,1} \in \{10, 17, 21, 23, 28, \quad \}$$

Cryptanalysis

We show an example where $M = \{2, 3, 5, 7, 11, 13\}$ and $S = 26$,

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} & 0 \\ & 2 \\ & 3 \\ & 5 \\ & 5 \\ & 7 \\ \ell \rightarrow & 8 \\ \ell \rightarrow & 10 \end{bmatrix} \star \begin{bmatrix} 0 & \leftarrow r \\ 7 & \leftarrow r \\ 11 & \leftarrow r \\ 13 & \leftarrow r \\ 18 & \leftarrow r \\ 20 & \\ 24 & \\ 31 & \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 11 \\ 13 \end{bmatrix}$$

$$L_{\ell,1} + R_{r,1} \in \{10, 17, 21, 23, 28, 26\}$$

We recovered $x = [0, 1, 1, 1, 1, 0]$ in less than $2^{6/2} = 8$ steps.

Both **B** and Gram-Schmidt(**B**) span the same space, i. e., $\mathbb{R}^2$, but not the same lattice.

$$\mathbf{B} = \begin{bmatrix} 47 & 95 \\ 215 & 460 \end{bmatrix}$$

Gram-Schmidt(**B**)

$$\| $$

$$\begin{bmatrix} 47 & -155875/48434 \\ 215 & 34075/48434 \end{bmatrix}$$

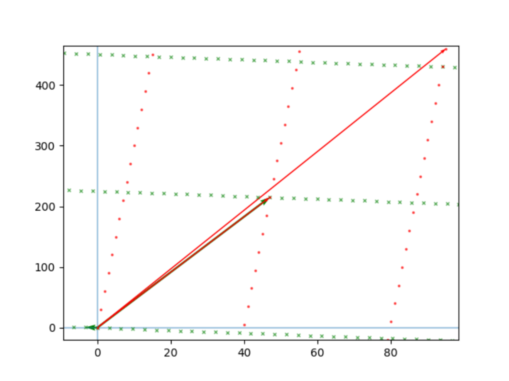

Figure 4: Lattice by **B** and Gram-Schmidt(**B**).

Cryptanalysis

1. A. K. Lenstra, H. W. Lenstra, L. Lovász published general lattice reduction algorithm (LLL) in 1982 [6].
2. LLL reduces any general lattice in polynomial time of,

$$O(n^2 \log n + n^2 \log \max(\mathbf{B}))$$

3. $|M| = n$ corresponds to the number of coordinates in any given lattice vector.
4. $\max \mathbf{B}$ is defined as the basis vector with the largest euclidean norm.

Cryptanalysis

```python
def lovasz_condition(G, Mu, k, delta):
    c = delta - Mu[k][k - 1]**2
    return G[:, k] @ G[:, k].T >= c * (G[:, k - 1] @ G[:, k - 1].T)

def lll(bad_basis, delta=0.75):
    B = np.array(bad_basis)
    G, Mu = gram_schmidt(B)  # G are the B*
    k, n = 1, B.shape[1] - 1
    while k <= n:
        for j in range(k - 1, -1, -1):
            if abs(Mu[k][j]) > 0.5:  # size condition not satisfied
                B[:, k] -= round(Mu[k][j]) * B[:, j]
                G, Mu = gram_schmidt(B)
        if lovasz_condition(G, Mu, k, delta):
            k = k + 1
        else:
            B[:, [k, k - 1]] = B[:, [k - 1, k]]  # swap
            G, Mu = gram_schmidt(B)
            k = max(k - 1, 1)
    return B
```

Listing 4: Tashfeen's Python implementation of the general LLL lattice reduction algorithm.

Cryptanalysis

It is here, where we use the specialised construction of the *Gram-Schmidt*, i. e., the *Gram-Schmidt* coefficients matrix,

$$\texttt{Mu} = \boldsymbol{\mu} \iff \texttt{Mu[k][j]} = \mu_{k,j} = \left(\frac{v_k \cdot u_j}{u_j \cdot u_j}\right) \text{ for } k > j.$$

Cryptanalysis

Both $\mathbf{B}$ and $\text{LLL}(\mathbf{B})$ span the same space, i. e., $\mathbb{R}^2$, and the same lattice.

$$\mathbf{B} = \begin{bmatrix} 47 & 95 \\ 215 & 460 \end{bmatrix}$$

$$\text{LLL}(\mathbf{B}) = \begin{bmatrix} 1 & 40 \\ 30 & 5 \end{bmatrix}$$

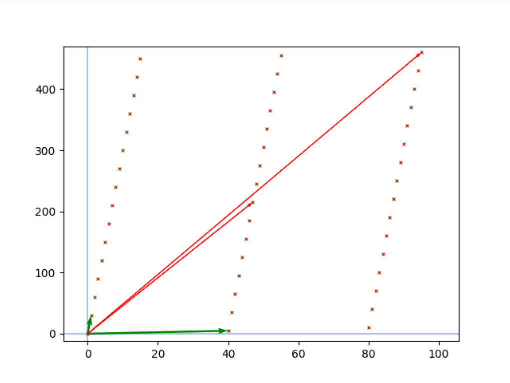

Figure 5: Lattice spanned by $\mathbf{B}$ and $\text{LLL}(\mathbf{B})$.

Note that $\text{LLL}(\mathbf{B})$ is short and almost orthogonal.

Cryptanalysis

We set up a lattice based attack on the Merkle–Hellman scheme.
Recall that $r_i \in \Theta(2^{2n})$,

$$\Theta(2^{2n}) \ni 2^{2n} = \underbrace{(2 \cdot 2^{n-1} \cdot 2^n)}_{B} > \underbrace{(2^{n-1} \cdot 2^n)}_{r'_n} \geq \cdots \geq \underbrace{(2 \cdot 2 \cdot 2^n)}_{r'_3} \geq \underbrace{(2 \cdot 2^n)}_{r'_2} \geq r'_1$$

Consider basis $\boldsymbol{\kappa} \in \mathbb{Z}^{d \times d}$ with $\dim(\boldsymbol{\kappa}) = d = n + 1$,

$$\boldsymbol{\kappa} = \begin{bmatrix} 2 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 2 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 2 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & 1 \\ r_1 & r_2 & r_3 & \cdots & r_n & S \end{bmatrix}$$

The lattice spanned by $\boldsymbol{\kappa}$ must have a vector that is the result of the following linear combination due to x,

$$t = \begin{bmatrix} 2 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 2 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 2 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & 1 \\ r_1 & r_2 & r_3 & \cdots & r_n & S \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \\ -1 \end{bmatrix} = \begin{bmatrix} 2x_1 - 1 \\ 2x_2 - 1 \\ 2x_3 - 1 \\ \vdots \\ 2x_n - 1 \\ M \cdot x - S \end{bmatrix} = \begin{bmatrix} 2x_1 - 1 \\ 2x_2 - 1 \\ 2x_3 - 1 \\ \vdots \\ 2x_n - 1 \\ 0 \end{bmatrix}$$

Therefore,

$$x \in \{0, 1\}^n \Rightarrow 2x_i - 1 = \pm 1 \Rightarrow ||t|| = \sqrt{n}$$

$||t||$ is at a stark contrast with the other vectors in the lattice spanned by $\boldsymbol{\kappa}$ due to the relative size of $r_i \in \Theta(2^{2n})$

Eve has,

$$S = 2002491457667039$$

$$M = [r_1, r_2, r_3, \cdots, r_{25}]$$

$$= [67108861, 134217725, 268435453, 536870909, 1073741821, 2147483645, 4294967293, 8589934589, 17179869181, 34359738365, 68719476733, 137438953469, 274877906941, 549755813885, 1099511627773, 2199023255549, 4398046511101, 8796093022205, 17592186044413, 35184372088829, 70368744177661, 140737488355325, 281474976710653, 562949953421309, 1125899906842621]$$

And the encoding table,

␣	A	B	···	T	Y	Z
0	1	2	···	20	25	26
00000	00001	00010	···	10100	11001	11010

Eve generate the bad basis $\boldsymbol{\kappa}$.

$$\boldsymbol{\kappa} = \begin{bmatrix}
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \\
r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 & r_8 & r_9 & r_{10} & r_{11} & r_{12} & r_{13} & r_{14} & r_{15} & r_{16} & r_{17} & r_{18} & r_{19} & r_{20} & r_{21} & r_{22} & r_{23} & r_{24} & r_{25} & S
\end{bmatrix}$$

Eve computes LLL($\boldsymbol{\kappa}$).

$$\begin{bmatrix}
-4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1174258 \\
2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 3 & -1 & 0 & 0 & 0 & 0 & 1174260 \\
0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1174260 \\
0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1174258 \\
0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & -3 & 0 & 0 & 0 & 0 & 1174262 \\
0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1174264 \\
0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1174266 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1174276 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 3 & 0 & 0 & 0 & 0 & 1174296 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1174328 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1174396 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1174534 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1174810 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1175364 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -1 & -4 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1176470 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & -4 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1178676 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 & -4 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1183098 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 2 & -4 & -3 & -3 & 0 & 0 & 0 & 0 & 1191934 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2 & -1 & -1 & 0 & 0 & 0 & 0 & 1209608 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -3 & 0 & 0 & 0 & 0 & 1244958 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & -4 & 0 & 0 & 0 & 1315654 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 2 & -4 & 0 & 0 & 1457052 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 & -4 & 0 & 1739842 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 2 & -4 & 2305430 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 2 & 3436596 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 0 & 3 & 3 & 3 & 3 & 0 & 0 & 3 & 3 & 3 & 3 & 782839
\end{bmatrix}$$

Eve computes LLL($\boldsymbol{\kappa}$).

$$
\begin{bmatrix}
-4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1174258 \\
2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 3 & -1 & 0 & 0 & 0 & 0 & 1174260 \\
0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1174260 \\
0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1174258 \\
0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & -3 & 0 & 0 & 0 & 0 & 1174262 \\
0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1174264 \\
0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1174266 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1174276 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 3 & 0 & 0 & 0 & 0 & 1174296 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1174328 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1174396 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1174534 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1174810 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1175364 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -1 & -4 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1176470 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & -4 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1178676 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2 & -4 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1183098 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 2 & -4 & -3 & -3 & 0 & 0 & 0 & 0 & 1191934 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2 & -1 & -1 & 0 & 0 & 0 & 0 & 1209608 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -3 & 0 & 0 & 0 & 0 & 1244958 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & -4 & 0 & 0 & 0 & 1315654 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 2 & -4 & 0 & 0 & 1457052 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 & -4 & 0 & 1739842 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 2 & -4 & 2305430 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 2 & 3436596 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 0 & 3 & 3 & 3 & 3 & 0 & 0 & 3 & 3 & 3 & 3 & 782839
\end{bmatrix}
$$

Eve verifies,

$$||t|| = \sqrt{n} = \sqrt{25} = 5$$

She then lets,

$$x \equiv t - 1 \equiv [0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1] \mod 3$$

Breaks x per encoding,

$$00011 \quad 01000 \quad 00101 \quad 01110 \quad 00111$$

Eve verifies,

$$||t|| = \sqrt{n} = \sqrt{25} = 5$$

She then lets,

$$x \equiv t - 1 \equiv [0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1] \mod 3$$

Breaks x per encoding,

$$\begin{array}{ccccc} 00011 & 01000 & 00101 & 01110 & 00111 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 3 & 8 & 5 & 14 & 7 \end{array}$$

Eve verifies,

$$||t|| = \sqrt{n} = \sqrt{25} = 5$$

She then lets,

$$x \equiv t - 1 \equiv [0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1] \mod 3$$

Breaks x per encoding,

$$\begin{array}{ccccc} 00011 & 01000 & 00101 & 01110 & 00111 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 3 & 8 & 5 & 14 & 7 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \text{C} & \text{H} & \text{E} & \text{N} & \text{G} \end{array}$$

Lattice Problems

Shortest Vector Problem (SVP) The knapsack problem is at most as hard as the problem of finding the shorted vector in a lattice.

Shortest Vector Length Unlike knapsack lattices, the length of the shortest vector $\lambda_1(\mathcal{L})$ is unknown in the general case.

$$\lambda_1(\mathcal{L}) \leq \sqrt{\gamma_d}(\det \mathcal{L})^{1/d}$$

$$\det \mathcal{L} \leq \prod_{i=1}^{d} ||b_i||$$

Correlated Orth. & Equality

Hermite's constant γ_d is only known for $d < 9$.

Lattice Problems

Three easier problems,

Hermite-SVP For $\alpha > 0$, find a vector $v \in \mathcal{L}$ such that $||v|| < \alpha \cdot (\det \mathcal{L})^{1/d}$.

Approx-SVP For $\alpha > 0$, find a vector $v \in \mathcal{L}$ such that $||v|| < \alpha \cdot \lambda_1(\mathcal{L})$.

Unique-SVP For $g > 1$, such that $\lambda_2(\mathcal{L})/\lambda_1(\mathcal{L}) \geq g$, find the unique shortest $v \in \mathcal{L}$.

Any algorithm that solves the Hermite-SVP with an approximation factor of α also solves the Approx-SVP with α^2 [7] [1].

Lattice Problems

Hermite $\alpha^{1/d}$	LLL	BKZ	DEEP
Empirical	1.0219	1.0128	1.011
Theoretical	1.0754	1.0337	1.075

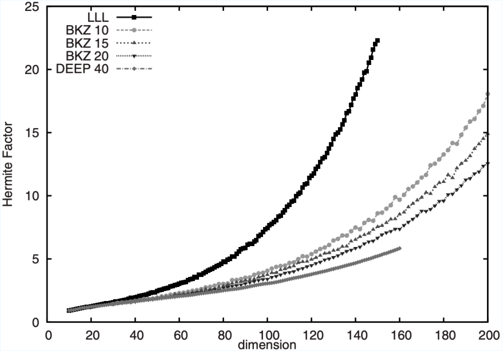

Table 1: Hermite factor gap for LLL, BKZ and DEEP where $1 \leq d \leq 200$ from by Gama et al [1].

For LLL,

$$\alpha^2 = \left(\frac{4}{3}\right)^{\frac{151-1}{4}\times 2} = \left(\frac{4}{3}\right)^{\frac{151-1}{2}}$$

Judging key size of 150 using the theoretical upper bound,

$$\left(\frac{4}{3}\right)^{\frac{151-1}{2}} < 2346417266$$

Judging key size of 150 using the empirical upper bound,

$$(1.0219)^{\frac{151-1}{2}} < 5.1$$

For the example we showed with key $n = 26$,

$$1.0754^{2\times 25} \approx 38 \gg 3 \approx 1.0219^{2\times 25}$$

Conclusion

1. We spoke about $\mathcal{P}$ and $\mathcal{NP}$and an attempt in $\mathcal{NP}$-complete based cryptosystem.
2. The $\mathcal{NP}$-complete problem was the knapsack problem.
3. We saw different lattice reduction algorithms and how they can used to solve the knapsack problem.
4. We observed a gap in theoretical upper bounds on lattice reduction algorithms and empirical estimates.
5. We saw this gap playing out in key sizes.

References I

Nicolas Gama and Phong Q Nguyen.
Predicting lattice reduction.
In *Advances in Cryptology–EUROCRYPT 2008: 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings 27*, pages 31–51. Springer, 2008.

Daniel Goldstein and Andrew Mayer.
On the equidistribution of hecke points.
Forum Mathematicum, 15:165–189, 2003.

T. Gowers, J. Barrow-Green, and I. Leader.
The Princeton Companion to Mathematics, page 583.
Princeton University Press, 2010.

Jeffrey Hoffstein, Jill Pipher, Joseph H Silverman, and Joseph H Silverman.
An introduction to mathematical cryptography, volume 1, pages 377, 378, 387, 381, 437, 443, 395.
Springer, 2008.

References II

Steven George Krantz.
The proof is in the pudding: The changing nature of mathematical proof, page 203.
Springer, 2011.

Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász.
Factoring polynomials with rational coefficients.
Mathematische annalen, 261(ARTICLE):515–534, 1982.

László Lovász.
An algorithmic theory of numbers, graphs and convexity, cbms-nsf reg.
In *Conf. Ser. Appl. Math*, volume 50, page 91, 1986.

Ralph C Merkle and Martin E Hellman.
Hiding information and signatures in trapdoor knapsacks.
In *Secure communications and asymmetric cryptosystems*, pages 197–215. Routledge, 2019.

Thank You!

Questions?