Coppersmith's Method : Solutions to Univariate Polynomials

Presented By: Shreya Dey

Institute for Advancing Intelligence, TCG CREST
&
Ramkrishna Mission Vivekananda Educational and Research Institute

About Me

- Completed My B.Sc and M.Sc from University of Calcutta
- Junior Research Fellow from August 2021
- Finished the courseworks
- Area of Research: Lattice Based Cryptography
- Co-Supervisor : Dr. Avijit Dutta

1st Semester

1. Discrete Mathematics
2. Graph Theory and Matroid
3. Algebra and It's Application
4. Basic Cryptology

2nd Semester

1. Design and Analysis of Algorithm
2. Trends in Combinatorics and Topology
3. Advanced Cryptology
4. Research Methodology

Papers

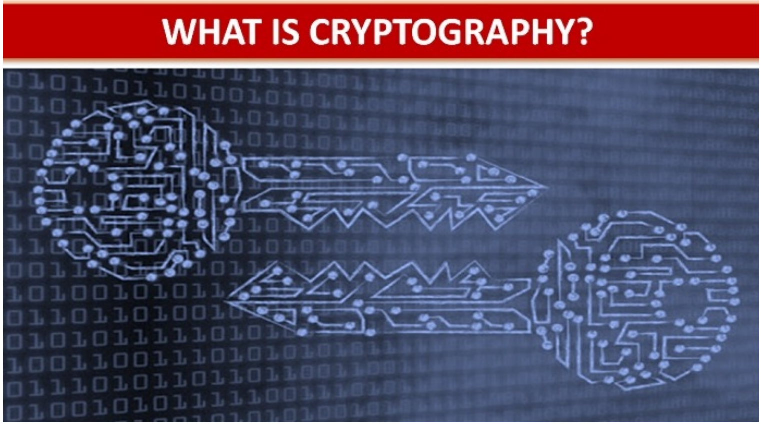
1. Don Coppersmith: Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. Journal of Cryptology(1997)10(4): 233-260 (1997)
2. Jean-Sébastien Coron: Finding Small Roots of Bivariate Integer Polynomial Equations Revisited. EUROCRYPT 2004: 492-505
3. Dan Boneh, Glenn Durfee, Nick Howgrave-Graham: Factoring $N = p^r q$, for large r. CRYPTO 1999: 326-337
4. Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman NTRU: A Ring-Based Public Key Cryptosystem. ANTS 1998: 267-288

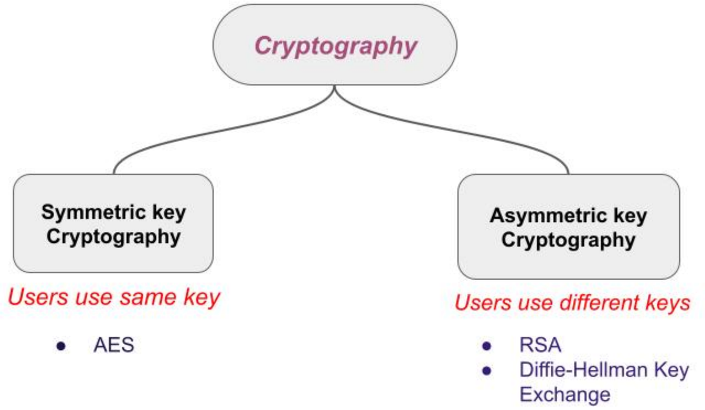
Books I

1. Steven D. Galbraith: Mathematics of Public Key Cryptography. Cambridge University Press 2012, ISBN 9781107013926
    - Chapter 19: Coppersmith's Method and Related Applications
2. Daniele Micciancio, Shafi Goldwasser: Complexity of lattice problems - a cryptograhic perspective. The Kluwer international series in engineering and computer science 671, Springer 2002, ISBN 978-0-7923-7688-0, pp. I-X, 1-220
    - Chapter 2: Approximations Algorithms
3. Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman: An Introduction to Mathematical Cryptography. ISBN: 978-0-387-77993-5
    - Chapter 6: Lattices and Cryptography

Books II

4. Oded Goldreich: The Foundations of Cryptography - Volume 1: Basic Techniques. Cambridge University Press 2001, ISBN 0-521-79172-3
   - Chapter 2: Computational Difficulty
5. Oded Regev: Lattices in Computer Science. Tel-Aviv University, Fall 2004
   - Lecture 1
   - Lecture 2
   - Lecture 3


WHAT IS CRYPTOGRAPHY?


Cryptography
Symmetric key Cryptography
Asymmetric key Cryptography
Users use same key
AES
Users use different keys
RSA
Diffie-Hellman Key Exchange

Security

"Security comes from hard problems"

Hard : no known polynomial time algorithm to solve using classical computer

- For **RSA**, Integer Factorization Problem
- For **Diffie-Hellman Key Exchange**, Discrete Logarithm Problem

New paradigm of computation

Quantum Computation

Quantum Algorithms

- Shor's Algorithm, for factorization
- Grover's Algorithm, for unstructured database search
- Simon's Algorithm, for period finding

Quantum supremacy

Quantum Algorithm + Quantum Computer $\Longrightarrow$

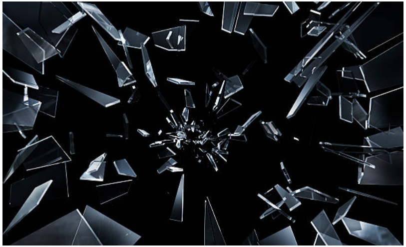


Figure: Classical Cryptography

Post-Quantum

So, what solutions should we adopt?

Hard problems in presence of Quantum Computer

Possible candidates

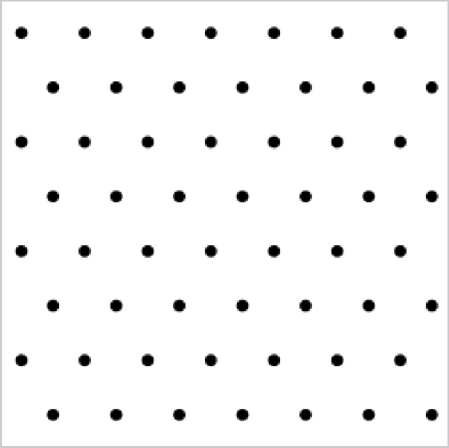
- Lattice-Based Cryptography
- Code-Based Cryptography
- Isogeny Based Cryptography

Basically started looking into,

"***Lattice-Based Cryptography***"

What is a Lattice?

- An infinite arrangement of "regularly spaced" points



Definition

- a discrete additive subgroup of $\mathbb{R}^n$ or,
- $\mathcal{L}(B) \triangleq \{B \cdot \vec{x} : \vec{x} \in \mathbb{Z}^n\} = \{\sum_{i=1}^{k} x_i \vec{b}_i : x_i \in \mathbb{Z}\}$, where $B = [\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_k]$ is k linearly independent vectors in $\mathbb{R}^n$
- For example, The lattice generated by $(1,0)^\intercal$ and $(0,1)^\intercal$ is $\mathbb{Z}^2$, the lattice of all integers points

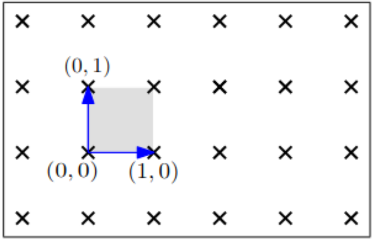


Figure: A basis of $\mathbb{Z}^2$

Seems to be Hard

So, what would a hard problem in lattice looks like?

The Shortest Vector Problem

- Given: A basis for a lattice $\mathcal{L}$
- Find: A non-zero lattice point in $\mathcal{L}$ as close as possible origin point

The Closest Vector Problem

- Given: A basis for a lattice $\mathcal{L}$ and a target vector
- Find: A non-zero lattice point in $\mathcal{L}$, closest to that target vector

No efficient algorithm is known to solve SVP and CVP exactly in arbitrary high dimension

Seems to be Hard

Typically, for **cryptographic purpose** we consider approximate variant of SVP and CVP

SVPγ

- Given: A basis for a lattice $\mathcal{L}$
- Find: A non zero lattice vector whose length is at most some approximation factor γ times the length of the shortest nonzero vector, for $\gamma = \gamma(n) \geq 1$

CVPγ

- Given: A basis for a lattice $\mathcal{L}$ and a target vector
- Find: A non zero lattice vector whose length is at most some approximation factor $\gamma(n)$ times the length of the closest nonzero vector

Motivation

- If the basis is orthogonal, solving SVP and CVP are easy

Example

In $\mathbb{R}^3$,

- basis $B = \{(1,0,0)^\intercal, (0,1,0)^\intercal, (0,0,1)^\intercal\}$
- target vector $t = (4.6, 2.3, 6.8)$
- closest vector $(4, 2, 6)$

- So, our target is convert a given basis to an orthogonal and short basis

Motivation

Apply Gram-Schimdt orthogonalization

- span the same space
- may not be a basis for $\mathcal{L}$

Modify the Gram-Schimdt process

Reduced the basis so that

- The vectors will be as short as possible
- The first vector will be the shortest vector and then the length of the other consecutive vectors increase slowly

Algorithmic solution to SVP and CVP

Some known polynomial time lattice reduction algorithms

- **LLL Algorithm**, solves SVPγ
- **Babai Algorithm**, solves CVPγ

LLL Algorithm

- Published in 1982
- Authors were A. K. Lenstra, H. W. Lenstra and L. Lovasz
- Designed to solve "Factoring Polynomials With Rational Co-efficients"
- Widely used To find short lattice vectors

Theorem (Lenstra, Lenstra, Lovasz)

There is a polynomial time algorithm that finds a basis for $\mathcal{L}$ satisfying both the ***Size Condition*** *and* ***Lovasz condition****:*

1. ***(Size Condition)***, $|\mu_{i,j}| \leqslant \frac{1}{2}, \forall i > j$
2. ***(Lovasz Condition)***, $(\delta - \mu_{i+1,i}^2)\|v_i^*\|^2 \leqslant \|v_{i+1}^*\|^2$, *for any pair of consecutive vectors* v_i^*, v_{i+1}^* ***(Gram–Schmidt process*** *orthogonal basis vectors) and* $\delta \in (\frac{1}{4}, 1)$

Such basis is called ***LLL reduced basis****.*

LLL Algorithm

LLL algorithm,

- solves SVPγ, for $\gamma = (\frac{2}{\sqrt{3}})^n$, where n is the rank of the input lattice
- finds a so-called reduced basis of relatively short lattice vectors for the lattice

- SVPγ and CVPγ are *seems to be hard* for
  - exactly $\gamma = 1$ or,
  - even approximate versions for small values of γ

Application of LLL

Coppersmith's Algorithm

- Designed to find "small integer roots of **univariate** polynomial modulo a given integer"

Coppersmith's Algorithm

Univariate Modular Polynomial

- Basic setup: a univariate monic polynomial
  $F(x) = x^d + a_{d-1}x^{d-1} + ... + a_2x^2 + a_1x + a_0$, over $\mathbb{Z}[x]$ with degree $d > 1$
  and, a modulus M of unknown factorization
- Goal - To find "small roots" x_0 such that $|x_0| < B$, for a suitable bound B and $F(x_0) \equiv 0 \pmod{M}$

Coppersmith proposed a method, where $B = M^{\frac{1}{d}}$

Coppersmith's Algorithm

The Central Problem

Suppose $\exists$ atleast one solution x_0 to $F(x) \equiv 0 \pmod{M}$ and that $|x_0| \leq M^{\frac{1}{d}}$

Coppersmith's Algorithm

Coefficients are small enough

- Find roots over $\mathbb{Z}$:
  - Get roots over $\mathbb{R}$: Newton's method
  - Round approximation of the roots to nearest integer x_0
  - Check whether $F(x_0) = 0$ over $\mathbb{Z}$
- Go to **mod** M

Coppersmith's Idea

Coefficients are not small

- Build $G(x) \in \mathbb{Z}[x]$ from $F(x)$ such that

$$F(x_0) \equiv 0 \pmod{M} \implies G(x_0) = 0 \text{ over } \mathbb{Z}$$

Important theorems and Background

Theorem

(Howgrave-Graham): *Let* $M, X \in \mathbb{N}$ *and let* $F(x) = \sum_{i=0}^{d} a_i x^i \in \mathbb{Z}[x]$*. Suppose* $x_0 \in \mathbb{Z}$ *is a solution of* $F(x) \equiv 0 \pmod{M}$ *such that* $|x_0| \leq X$*. We associate with the polynomial the row vector*

$$b_F = (a_0, a_1 X, a_2 X^2, \cdots, a_d X^d)$$

If $\|b_F\| < \frac{M}{\sqrt{d+1}}$*, then* $F(x_0) = 0$*.*

Important theorems and Background

Definition

Let $G_i(x) = Mx^i, for 0 \leq i < d$ be $d+1$ polynomials that has the root $x_0 \pmod{M}$. Then we define a basis B corresponds to these polynomials $G_i(x)$ together with $F(x)$ for a lattice $\mathcal{L}$ as follows:

$$B = \begin{bmatrix} M & 0 & 0 & \cdots & 0 & 0 \\ 0 & MX & 0 & \cdots & 0 & 0 \\ 0 & 0 & MX^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & MX^{d-1} & 0 \\ a_0 & a_1X & a_2x^2 & \cdots & a_{d-1}X^{d-1} & X^d \end{bmatrix}$$

Important theorems and Background

Theorem

Suppose given a basis B as defined in ***Definition****, and $G(x)$ be the polynomial corresponding to the first vector in the* ***LLL*** *- reduced basis for $\mathcal{L}$. If*

$$X < \frac{M^{\frac{2}{d(d+1)}}}{\sqrt{2}(d+1)^{\frac{1}{d}}},$$

then any root x_0 of $F(x) \pmod{M}$ such that $|x_0| \leq X$ satisfies $G(x_0) = 0$ in $\mathbb{Z}$.

Coppersmith's Method

Coppersmith's Technique

- **Input**: $F(x)$, M, $M^{\frac{1}{d}}$
- Output: All solutions of $F(x) \equiv 0 \pmod{M}$ satisfying $|x_0| \leq M^{\frac{1}{d}}$

- **Step 1**: Define a sequence of $d+1$ polynomials $G_i(x)$ and if $F(x_0) \equiv 0 \pmod{M}$ $\implies G_i(x_0) \equiv 0 \pmod{M}$
- **Step 2**: Find a polynomial $G(x) \in \mathcal{L}(G_i)$ having small norm by using **LLL alorithm**
- **Step 3**: Solve the equation $G(x) = 0$ numerically; Output all integer roots within the target range

Some Applications of Coppersmith's Method

Some Attack variants of RSA

- ***Fixed Padding Scemes in RSA***
- ***Factoring*** $N = pq$ ***with Partial knowledge of*** p

Fixed Padding Schemes in RSA

- Given:
  - A κ-bit RSA moduli and a κ' bit message with $(\kappa' << \kappa)$
  - Fixed padding: Put $(\kappa - \kappa' - 1)$ 1's to the left hand side of the message
- Encryption:
  - Step 1: Suppose $\kappa = 1024$ and $\kappa' = 128$ (128 bit AES key K)
  - Step 2: Then

$$m = 2^{1024} - 2^{128} + K$$

  - Step 3: Suppose the encryption exponent is $e = 3$
  - Step 4: Then the ciphertext is $c = m^3 \pmod{N}$

Fixed Padding Schemes in RSA

- Idea:
  - Step 1: If the cryptanalyst get access the ciphertext then he only needs to find the value K
  - Step 2: We know K is a solution to the polynomial

$$F(x) = (2^{1024} - 2^{128} + x)^3 - c \equiv 0 \pmod{N}$$

  - Step 3: The polynomial is of degree 3 with a root modulo N
  - Step 4: Apply **Coppersmith's method** to find the solution K in polynomial time

Factoring $N = pq$ with Partial knowledge of p

- Given:
  - $N = pq$, with $p < q < 2p$
  - An approximation $\widetilde{p}$ of p. So, $p = \widetilde{p} + x_0$, x_0 is small
- Idea:
  - Step 1: Consider $F(x) = \widetilde{p} + x$. It has a small solution x_0 modulo p.
  - Step 2: Construct a sequence of polynomials $N, F(x), xF(x), x^2F(x), \cdots$ that have the root $x_0 \pmod{p}$
  - Step 3: Form a **lattice** corresponding to polynomials and apply **LLL**
  - Step 4: Get the polynomial $G(x)$ from the reduced basis
  - Step 5: Solve $G(x)$ over $\mathbb{Z}$ and check for solution of $F(x) \pmod{p}$
  - Step 6: Compute **p** as $\gcd(N, F(x_0))$

Future direction

- Explore Coppersmith's method for Bivariate Integer Polynomials
- Explore different attack variants for RSA
- Analyze different cryptographic algorithms based on Lattice


THANK YOU