CS641

Modern Cryptology

Lecture 13

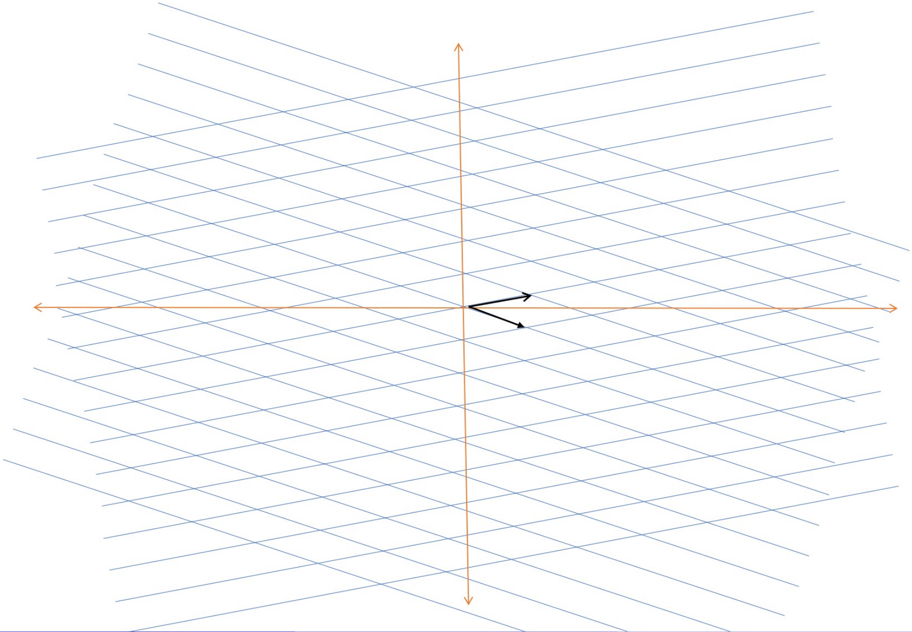
Integer Lattice

Given a set of linearly independent vectors $v_1, v_2, \ldots, v_D \in \mathbb{R}^D$, integer lattice generated by them is

$$\mathcal{L} = \{\sum_{i=1}^{D} a_i v_i \mid a_i \in \mathbb{Z}\}.$$

- $\mathcal{L}$ is a vector space consisting of integer linear combinations of vectors $v_1, \ldots, v_D$.
- Vectors $v_1, \ldots, v_D$ form a basis of the lattice.

Example in $\mathbb{R}^2$



Volume of Integer Lattices

Volume of lattice $\mathcal{L}$, denoted $v(\mathcal{L})$, is defined as $|\det(V)|$ where $V = [v_1\ v_2\ \cdots\ v_D]$.

Lemma

$v(\mathcal{L})$ is independent of the basis.

- Let u_1, u_2, ..., u_D be another basis for $\mathcal{L}$.
- Then u_i's can be written as integer linear combination of v_j's and vice versa.
- Let $U = [u_1\ u_2\ \cdots\ u_D] = AV$ and $V = BU$ with $A, B \in \mathbb{Z}^{D \times D}$.

Volume of Integer Lattices

- Then, $\det(U) = \det(A)\det(V) = \det(A)\det(B)\det(U)$, giving $\det(A)\det(B) = 1$.
- Since A and B have integer entries, $\det(A), \det(B) \in \mathbb{Z}$.
- Therefore, $\det(A) = \det(B) = \pm 1$ and $|\det(U)| = |\det(V)|$.

Shortest Vector

Shortest vector of lattice $\mathcal{L}$ is the minimum length non-zero vector in $\mathcal{L}$. Length of shortest vector is denoted as $\lambda_1(\mathcal{L})$.

- Finding shortest vector of a lattice is known to be a hard-to-solve problem.
- Even finding a vector of length within $\sqrt{2}\lambda_1(\mathcal{L})$ is known to be hard.
- However, it is possible to efficiently find a vector of length within $2^{(D-1)/2}\lambda_1(\mathcal{L})$.

Short Vectors

Minkowski's Theorem

For any lattice $\mathcal{L}$, $\lambda_1(\mathcal{L}) \leq \sqrt{D} v(\mathcal{L})^{1/D}$.

Lenstra-Lenstra-Lovasz (L^3) Algorithm

Given a lattice $\mathcal{L}$, it computes a vector v, in time polynomial in D, such that $|v| \leq 2^{(D-1)/2} \lambda_1(\mathcal{L})$.

Special Case: Small e

- In order to save time during encryption, e may be chosen to be small.
- The smallest possible value for e is 3 (when 3 does not divide $\phi(n) = (p-1)(q-1)$).
- For small e, however, the system can be broken without d.

Special Case: Small e

- Set $e = 3$.
- Suppose m is a 128 bit key of AES, and n is 1024 bits long.
- Let $c = m^e = m^3 \pmod{n}$.
- Then $c = m^3$ over integers and can be computed easily!
- To avoid this, we can pad m with all 1's to make it a 1023 bit number.
- Let this new $m' = a + m$ where a is the number corresponding to padding and is known.
- Let $R(x) = (a + Kx)^3 - c$ where K is an upper bound on m (we can take $K = 2^{128}$).
- $R(x)$ has a "small" root in Z_n since $R(m/K) = 0 \pmod{n}$ and $m/K \leq 1$.

Special Case: Small e

- Define polynomials $R_j(x)$ for $0 \leq j \leq 4$ as: $R_j(x) = nK^j x^j$ for $0 \leq j \leq 2$, $R_3(x) = R(x)$, and $R_4(x) = KxR(x)$.
- Every R_j satisfies the property that $R_j(m/K) = 0 \pmod{n}$.
- Define vector $v_j \in \mathbb{Z}^5$ to contain the coefficients of polynomial R_j.
- Let $\mathcal{L} \subset \mathbb{Z}^5$ be the lattice generated by vectors v_j, $0 \leq j \leq 4$.
- Let $R(x) = K^3x^3 + c_2K^2x^2 + c_1Kx + c_0$.
- Then, we have

$$v(\mathcal{L}) = \det \begin{bmatrix} K^4 & c_2K^3 & c_1K^2 & c_0K & 0 \\ 0 & K^3 & c_2K^2 & c_1K & c_0 \\ 0 & 0 & K^2n & 0 & 0 \\ 0 & 0 & 0 & Kn & 0 \\ 0 & 0 & 0 & 0 & n \end{bmatrix} = n^3K^{10}.$$

Special Case: Small e

- Use L^3 algorithm to find a short vector $u \in \mathcal{L}$.
- We have:

$$|u| \leq 4\lambda_1(\mathcal{L}) \leq 4\sqrt{5}n^{3/5}K^2.$$

- Let $S(x)$ be the polynomial whose coefficients are given by vector u.
- Since S is an integer linear combination of R_j's, $S(m/K) = 0 \pmod{n}$.
- Moreover,

$$\begin{aligned} |S(m/K)| &\leq 20\sqrt{5}n^{3/5}K^2 \\ &< 2^6 2^{3072/5} 2^{256} \\ &< 2^{877} \\ &< n. \end{aligned}$$

Special Case: Small e

- Therefore, $S(m/K) = 0$ over $\mathbb{Z}$.
- It is easy to find roots of polynomial $S(x)$ over $\mathbb{Z}$ (using Newton's method for example).
- Root finding methods will give a close approximation of m/K.
- This approximation can be multiplied by K and closest integer to the result gives the value of m.

Special Case: Small e

- Similar attack breaks other small values of e too.
- Therefore, e must be chosen to be large.
- This implies we cannot save time during encryption.
- Can we save time during decryption by choosing small d?

Special Case: Small d

- Suppose $d < n^\epsilon$ for some $\epsilon > 0$ and $e > \Omega(n)$.
- We have:

$$de = 1 + r(p-1)(q-1) = 1 + r(n+1) - r(p+q).$$

- Since $d < n^\epsilon$ and $e < n$, we have $r = O(n^\epsilon)$. Let $r \leq K$.
- Let $s = r(p+q)$. Then $s < L = O(n^{1/2+\epsilon})$.

Special Case: Small d

- Define polynomial $R(x, y) = Ly - (n+1)Kx - 1$.
- We have $R(r/K, s/L) = 0 \pmod{e}$ and $r/K, s/L \leq 1$.
- How do we define a lattice using a polynomial in two variables?
- We use $K^i x^i R(x, y)$ for additional vectors: there will be t such vectors for $0 \leq i < t$.
- Terms of these polynomials are $x^i y$ and x^j for $0 \leq i < t$, $0 \leq j \leq t$.
- Therefore, there are a total of $2t + 1$ terms.
- We can get the same number of vectors by taking additional polynomials $K^i x^i e$ for $0 \leq i \leq t$.
- All the additional polynomials are 0 modulo e.

Special Case: Small d

- These polynomials give rise to a lattice $\mathcal{L}$ in $\mathbb{Z}^{2t+1}$.
- Vectors given by polynomials above will form an upper triangular matrix when terms are ordered as yx^{t-1}, yx^{t-2}, …, y, x^t, x^{t-1}, …, 1.
- Volume of $\mathcal{L}$ equals

$$\prod_{i=0}^{t-1}(LK^i)\prod_{i=0}^{t}(K^i e) = L^t K^{t(t-1)/2} K^{t(t+1)/2} e^{t+1} = L^t K^{t^2} e^{t+1}.$$

- The vector in $\mathcal{L}$ computed by L^3 algorithm has length at most

$$\ell = 2^t \sqrt{2t+1} L^{t/(2t+1)} K^{t^2/(2t+1)} e^{(t+1)/(2t+1)}.$$

Special Case: Small d

- Using bounds $L = c_L n^{1/2+\epsilon}$, $K = n^{\epsilon}$, and $e < n$, we get that:

$$\begin{aligned} \ell &< 2^t \sqrt{(2t+1)c_L}\, n^{t/2(2t+1)+\epsilon t/(2t+1)+\epsilon t^2/(2t+1)+(t+1)/(2t+1)} \\ &= 2^t \sqrt{(2t+1)c_L}\, n^{(2\epsilon t^2+2\epsilon t+3t+2)/2(2t+1)} \end{aligned}$$

- If $(2t+1)\ell < e$ then we have the desired property: the polynomial defined by L^3 vector is zero over integers for $x = r/K$ and $y = s/L$.
- This requires $2\epsilon t^2 + 2\epsilon t + 3t + 2 < 2(2t+1)$, or $\epsilon < 1/2(t+1)$.

Special Case: Small d

- Suppose we have $\epsilon = 1/2(t+1) - \delta$ for some $\delta > 0$.
- Then,

$$
\begin{aligned}
(2t+1)\ell &< 2^t(2t+1)^{3/2}\sqrt{c_L}\,n^{(t-2(t+1)\delta+3t+2)/2(2t+1)} \\
&< 2^t(2t+1)^{3/2}\sqrt{c_L}\,n^{1-\delta/2} \\
&< e
\end{aligned}
$$

  since $e = \Omega(n)$, provided t is chosen to be small.
- For $t = 1$, we get $\epsilon < 1/4$.
- The L^3 polynomial, $R_1(x, y)$, satisfies $R_1(r/K, s/L) = 0$.
- A bivariate polynomial has infinitely many roots, and so it is still not easy to identify the desired root.

Special Case: Small d

- We identify another polynomial with same property using the second smallest vector computed by L^3 algorithm:
  - This vector is linearly independent of first one and has length bounded by $2^{(D-1)/2}\sqrt{D}v(\mathcal{L})^{1/(D-1)}$.
- A similar calculation done for this vector gives condition $\epsilon < 1/2(t+1) - 1/t(t+1)$.
- For $t = 1, 2$: $\epsilon < 0$, which is of no use.
- The best bound is obtained for $t = 4$: $\epsilon < 1/20$.
- For this ϵ and $t = 4$, we get two bivariate polynomials R_1 and R_2 with $(r/K, s/L)$ as common root.
- GCD of these two polynomials is likely to give a univariate polynomial in x with root r/K.
- This can be used to find r and s, thus resulting in d.

Special Case: Small d

- Therefore, if $d < n^{1/20}$, it can be computed from e and n.
- With choice of more involved polynomials, one can show the same result for $d < n^{0.292}$.
- Hence, neither e nor d can be chosen small.