Twenty Years of Lattice Reduction in Cryptology

Phong Nguyen and **Jacques Stern**

Département d'Informatique

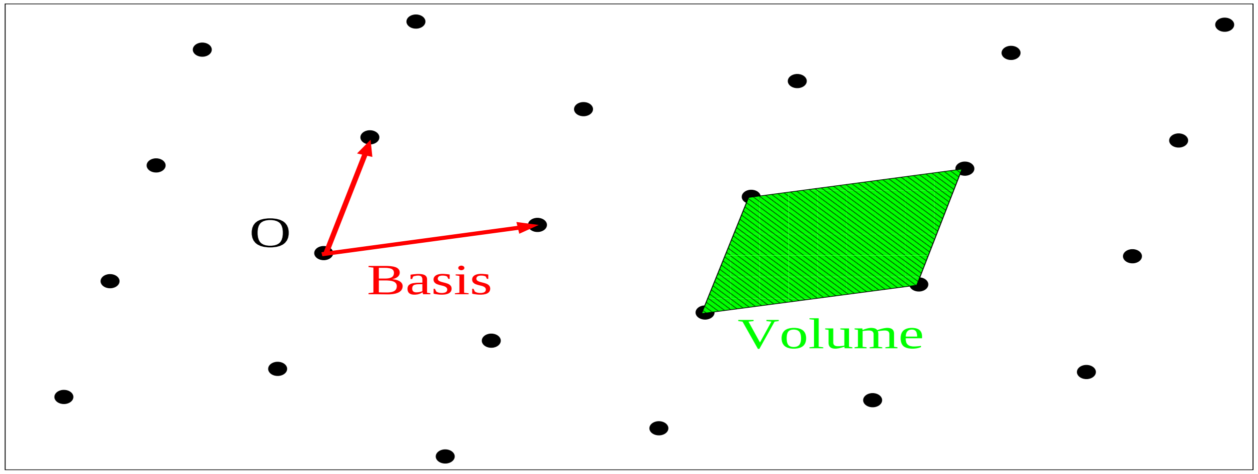
École Normale Supérieure

March 2000

Summary

- Lattices and algorithmic geometry of numbers
- Knapsack cryptography
- Ajtai and lattice cryptography
- Coppersmith and attacks on RSA

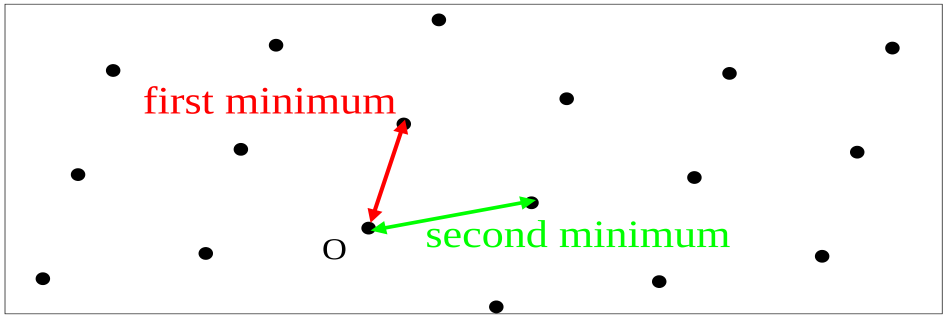
Integer Lattices



- Integer lattice = subgroup L of $(\mathbb{Z}^n, +)$.
- Dimension of L = dimension d of the vector subspace $\mathbb{R}.L$.
- Basis of L = free family $(\vec{b}_1, \ldots, \vec{b}_d)$ such that

  $L = \mathbb{Z}.\vec{b}_1 \oplus \cdots \oplus \mathbb{Z}.\vec{b}_d$ (there are infinitely many)
- Volume of L = d-dimensional volume of a basis.

Successive minima



- Let L be a d-dimensional lattice of $\mathbb{Z}^n$.
- $\lambda_1(L)$ = Euclidean norm of a shortest nonzero vector of L.

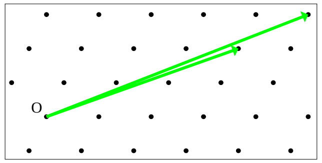
  $\lambda_k(L)$ = radius of the smallest ball including

  k linearly independent vectors of L.
- Minkowski : $\lambda_1(L) \leq \sqrt{d} \cdot \mathrm{vol}(L)^{1/d}$
- Lattice gap $= \lambda_2(L)/\lambda_1(L)$

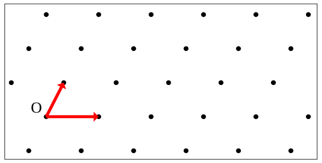
The Shortest Vector Problem (SVP)

Given a basis of L, find a shortest vector of L.

- NP-hard under randomized reductions (Ajtai, 1998).
- Approximation in $\sqrt{\dim}$ not NP-hard (Goldreich *et al.*, 1998).
- Best polynomial-time algorithms : exponential bounds.
- In practice, much better results.



Lattice reduction

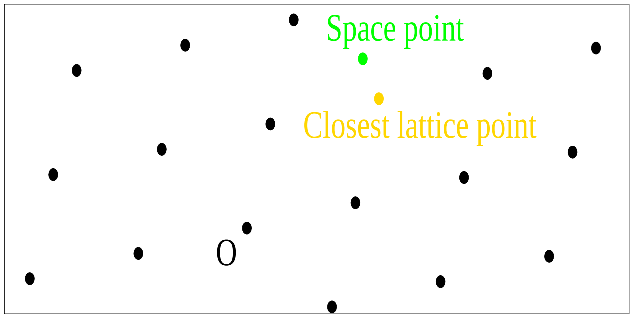


Find a lattice basis reasonably orthogonal
approaching the successive minima.

The Closest Vector Problem (CVP)

Given a basis of L and $\vec{c} \in \mathbb{R}^n$,
find a lattice point closest to $\vec{c}$.

- NP-hard (van Emde Boas, 1981).
- Approximation in $\sqrt{\dim}$ not NP-hard (Goldreich *et al.*, 1998)
- Considered "harder" than SVP, despite certains links.
- Best polynomial-time algorithms : exponential bounds.



Lattice reduction algorithms

- Lenstra-Lenstra-Lovász (LLL, 1982) :
  computes in polynomial time a basis $(\vec{b}_1, \ldots, \vec{b}_d)$ of L
  such that $\|\vec{b}_i\| \leq 2^{(d-1)/2}\lambda_i(L)$ and $\|\vec{b}_1\| \leq 2^{(d-1)/4}(\det L)^{1/d}$.
- Variants by Schnorr (1987-95) : approximation in $(1+\varepsilon)^d$.
- In practice, the bounds are much better.
- Applications : algorithmic number theory,
  integer programming, cryptanalysis, *etc.*

The Knapsack Problem

Given positive integers $a_1, a_2, \ldots, a_n$ and
a sum of the form $s = \sum_{i=1}^{n} x_i a_i$, where $x_i \in \{0, 1\}$,
the problem is to recover all x_i's.

- It is NP-hard (Karp).
- But there are easy instances.
- If the density $\dfrac{n}{\max \log_2 a_i}$ is $\leq 0.94\ldots$
  the problem can be reduced to SVP (Coster *et al.*, 1986-91).

Knapsack Cryptography

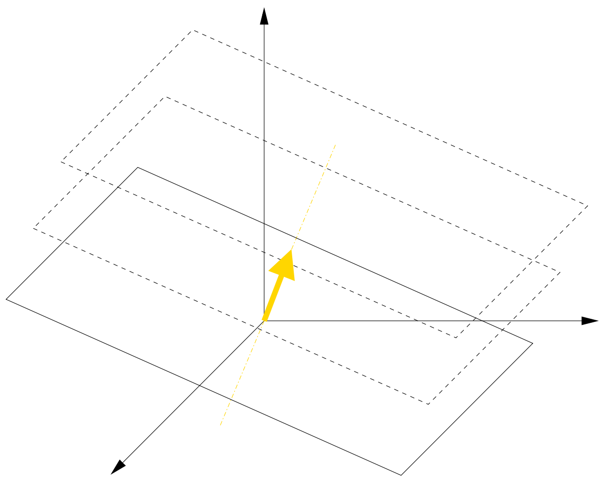
- Initiated by Merkle and Hellman (1978).
- Many propositions broken in the eighties, thanks to lattice reduction, considered easy.
- The low-density attacks are the most powerful attacks in practice.
- In 1997, knapsack cryptography is dead, but revival with lattice cryptography.

Lattice Cryptography

- Based on the hardness of lattice problems.
- Inspired by Ajtai's breakthroughs.
  - 1996 : worst case/average case equivalence for SVP* (SVP when the gap is polynomial in the dimension).
  - 1997 : NP-hardness (randomized) of SVP.
- Main propositions :
  - Ajtai-Dwork (STOC'97) $\rightarrow$ SVP.
  - Goldreich-Goldwasser-Halevi (Crypto'97) $\rightarrow$ CVP.
  - NTRU (ANTS'98) $\rightarrow$ SVP.

The Ajtai-Dwork cryptosystem (STOC'97)

The secret key is $\vec{u} \in \mathbb{R}^n$ of norm ≤ 1, inducing a distribution $\mathcal{H}$ of points with dot product with $\vec{u}$ close to $\mathbb{Z}$.



The public key is formed by points from $\mathcal{H}$:

- $\vec{v}_1, \ldots, \vec{v}_m$ where $m = n^3$.
- $\vec{w}_1, \ldots, \vec{w}_n$ spanning a parallelepiped $P(w)$ not too flat.

Encryption

- Ciphertexts of '0' : reduction modulo $P(w)$ of a sum of $\vec{v_i}$'s.
- Ciphertexts of '1' : random point in $P(w)$.

Decryption

The dot product of the ciphertext with $\vec{u}$ indicates whether it is close to one of the hyperplanes.

Security proof

If SVP* is hard in the worst case,
then Ajtai-Dwork is provably secure.

Attack to recover the secret key

- Each $\langle \vec{u}, \vec{v}_i \rangle$ is close to some integer V_i.
- If the λ_i's are small integers such that $\sum_{i=1}^{m} \lambda_i \vec{v}_i$ is short, then $\lambda_1 V_1 + \cdots + \lambda_m V_m = 0$.
- By means of lattice reduction, one finds many such λ_i's.
- This discloses the V_i's and an approximation of $\vec{u}$.
- The attack is likely to work up to $n \leq 32$: one needs a public key of at least several Mbs.

Reduction to CVP and SVP

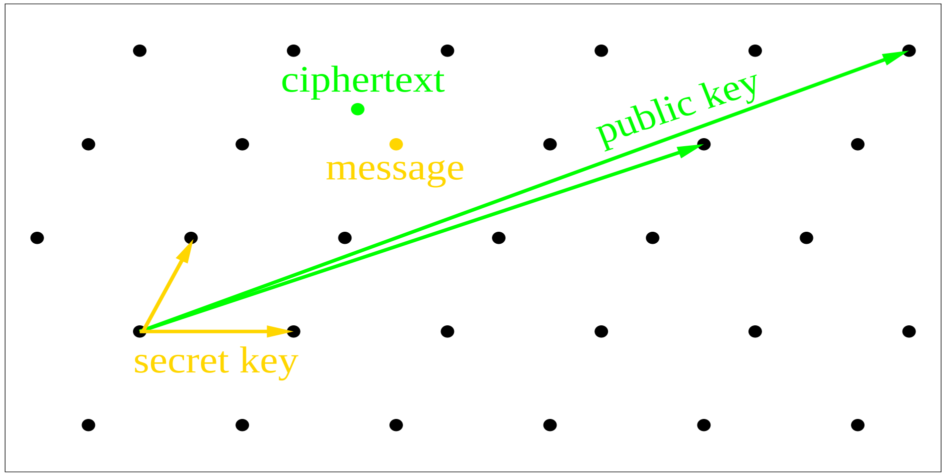
- A ciphertext of '0' is a sum of $\vec{v}_i$'s
  and a small (real) combination of the $\vec{w}_j$'s.
  It is close to a certain $(n+m)$-dimensional lattice.
- One deduces a polynomial Turing machine with CVP-oracle,
  taking as input a public key and a ciphertext,
  which decrypts with error probability polynomially small.
- But the corresponding CVP approximation problem
  is (unlikely) to be NP-hard.
- The proof can be adapted to a SVP-oracle.

Results on AD (Nguyen & Stern, Crypto'98)

- Ajtai-Dwork is insecure for any "realistic" key.
- One can decrypt if one can sufficiently well approximate CVP or SVP.
- Ajtai-Dwork security is equivalent to the hardness of SVP approximation.
- The decryption problem is (unlikely) to be NP-hard.

It is not clear whether SVP* should be hard.

The GGH Cryptosystem (Crypto'97)



- Security parameters $n \geq 200$ and $\sigma = 3$.
- Secret key = a very "good" basis of a full-rank lattice L in $\mathbb{Z}^n$.
- Public key = a "bad" basis B of L.
- $\vec{m} \in \mathbb{Z}^n$ is enciphered by $\vec{c} = \vec{m}B + \vec{e}$ where $\vec{e} \in_R \{\pm\sigma\}^n$.

Simplifying Decryption

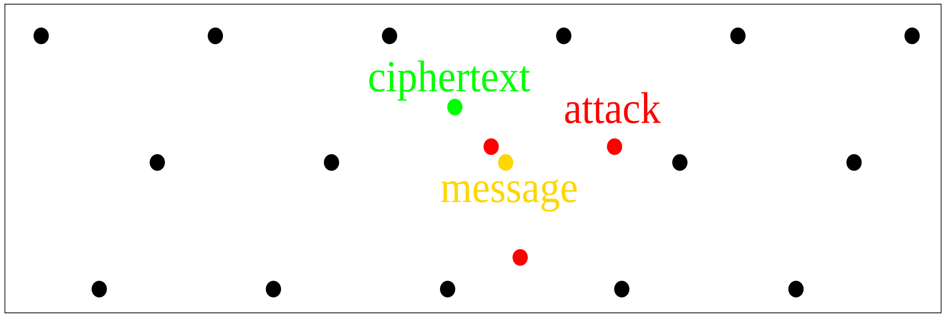
One has $\vec{c} = \vec{m}B + \vec{e}$ with $\vec{e} \in_R \{\pm\sigma\}^n$.

- Therefore $\vec{c} \equiv \vec{m}B \mod \sigma$.

  If B is invertible mod σ, one can compute $\vec{m}$ mod σ.

  Then, one can get much closer to the lattice.

- And a random matrix has a very small kernel mod σ.



Results on GGH (Nguyen, Crypto'99)

- Encryption must be modified : no special form for $\vec{e}$.
- Dimension must be ≥ 400 : 4 of 5 GGH challenges solved.
- GGH is then impractical (public key of several Mbs)

Modifications to GGH (Micciancio, 1999)

- Choose the HNF as the public basis.
- Encode the message into the error vector, and reduce it modulo the public basis.
- Keysize down to $O(n^2 \log n)$: worse than McEliece.
- GGH is not a real alternative to RSA.

Status of Lattice-based Cryptography

- Only NTRU is a real alternative (but lack of security proofs).
- All other PKCs have the same problems as knapsack systems. Keysize grows at least quadratically, which is impractical, because lattice reduction is efficient up to dimension at least 300.
- Hardness of SVP/CVP needs to be better understood.

The RSA Cryptosystem (notations)

- Public key : $N = p \cdot q$ with p and q primes,

  and public exponent e coprime with $\phi(N) = (p-1) \cdot (q-1)$.
- Secret key : d such that $e \cdot d \equiv 1 \mod \phi(N)$.
- A message $m \in \mathbb{Z}/N\mathbb{Z}$ is enciphered into $c = m^e \mod N$.

  Decryption : $m = c^d \mod N$.
- In general, the best attack is factoring N.

Integer Factorization

- Best general algorithm = number field sieve (NFS).

  Current record for an RSA number : 512 bits (August 1999),

  that is 155 decimal digits.

- NFS has 3 steps :
  - Sieving, in parallel on many computers.
  - Linear algebra, on a large computer with a lot of RAM.
  - Algebraic square root, on a "small" computer.

    LLL (in low dimension) is the core of the last stage.

Small roots of polynomial equations (Coppersmith, Eurocrypt'96)

- Let P be a monic polynomial of small degree δ and N a number of unknown factorization.
- In time polynomial in $(\log N, 2^{\delta})$, LLL can find all integers $x_0 \leq N^{1/\delta}$ such that $P(x_0) \equiv 0 \mod N$.
- Heuristic extension to multivariate polynomials.
- Analogous result for bivariate integer polynomials.

  One can factor $N = pq$ if one knows half of the bits of p.

Applications to RSA

- If $e = 3$, one can sometimes recover the plaintext (steoreotyped messages, broadcast, padding), using Coppersmith for univariate modular polynomials.
- If $d \leq N^{0.292}$, one can heuristically factor $N = pq$ (Boneh-Durfee, Eurocrypt'99) using Coppersmith for bivariate modular polynomials.
- If enough bits of d are known, one can recover all of d (Boneh *et al.*, Asiacrypt'98) using Coppersmith for bivariate integer polynomials.

Other applications of Coppersmith's result

- Factoring $N = p^r q$ with large r (Boneh *et al.*, Crypto'99)
- Finding smooth integers in small intervals (Boneh, STOC'2000)
- Noisy Chinese remaindering
  (Bleichenbacher & Nguyen, Eurocrypt'2000)

Conclusion

- Applications
  - positive (new schemes, security proofs).
  - negative (cryptanalysis).
- Lattice reduction solves fairly well :
  - "linear" problems (knapsacks, SVP/CVP).
  - certain non-linear problems.

    (small roots of low-degree polynomial equations)
- Paradox between theory and practice :
  - Algorithms behave much better than expected.
  - More and more hardness results.

Open problems

- Secure and efficient lattice cryptosystem ?
- New lattice reduction algorithms ?
- Improved analysis of LLL and variants ?
- Complexity of SVP* and
  approximation of SVP within a polynomial factor ?
- NP-hardness (deterministic) of SVP ?