Lattice-Based Cryptography: An overview

Phong Nguyễn

CNRS - École normale supérieure

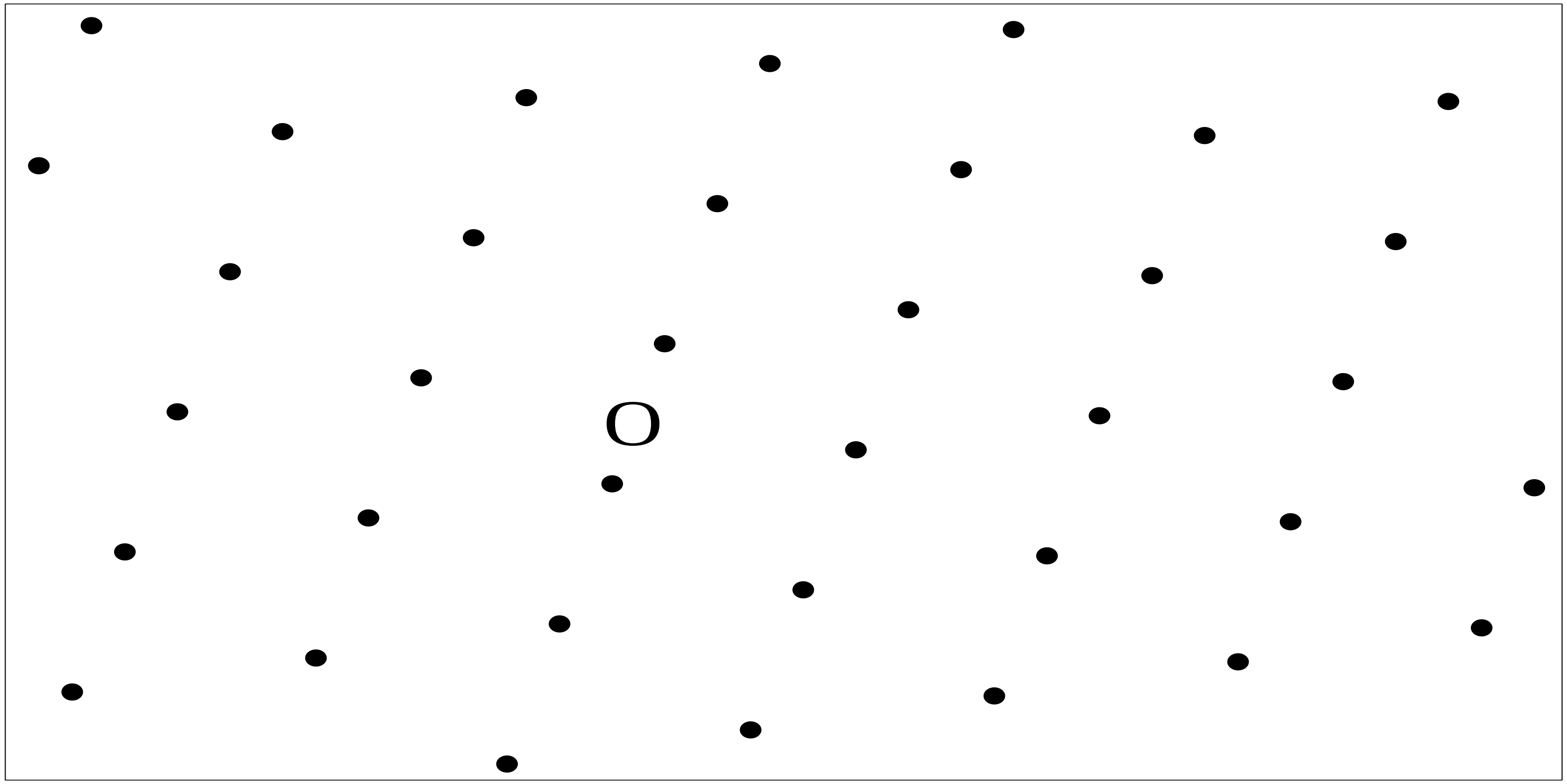
France

Essen, Sept 24th, 2002

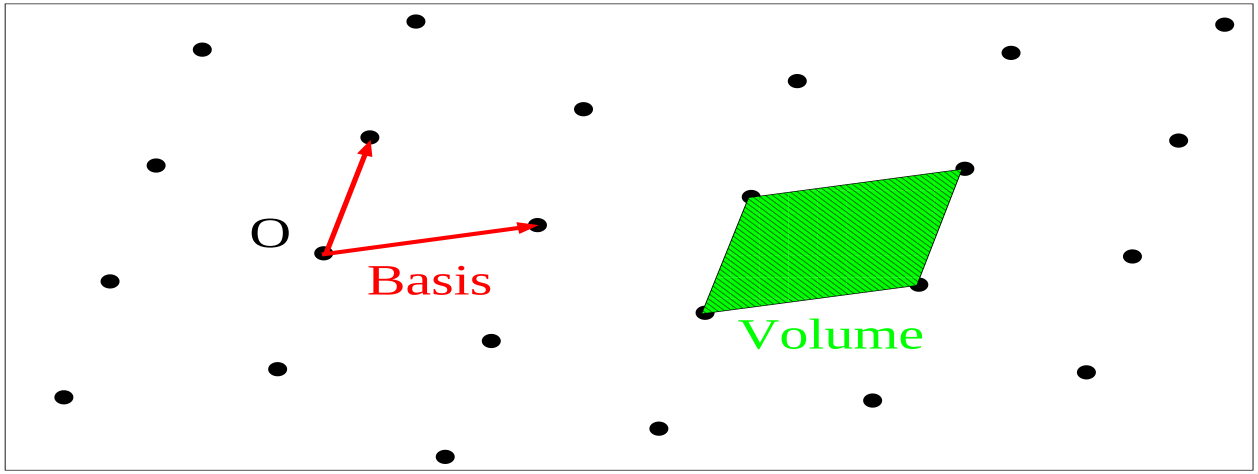
Summary

- Lattices and hard lattice problems
- The Ajtai-Dwork cryptosystem
- The GGH cryptosystem: Encryption and signature
- The NTRU cryptosystem: Encryption and signature

A Lattice



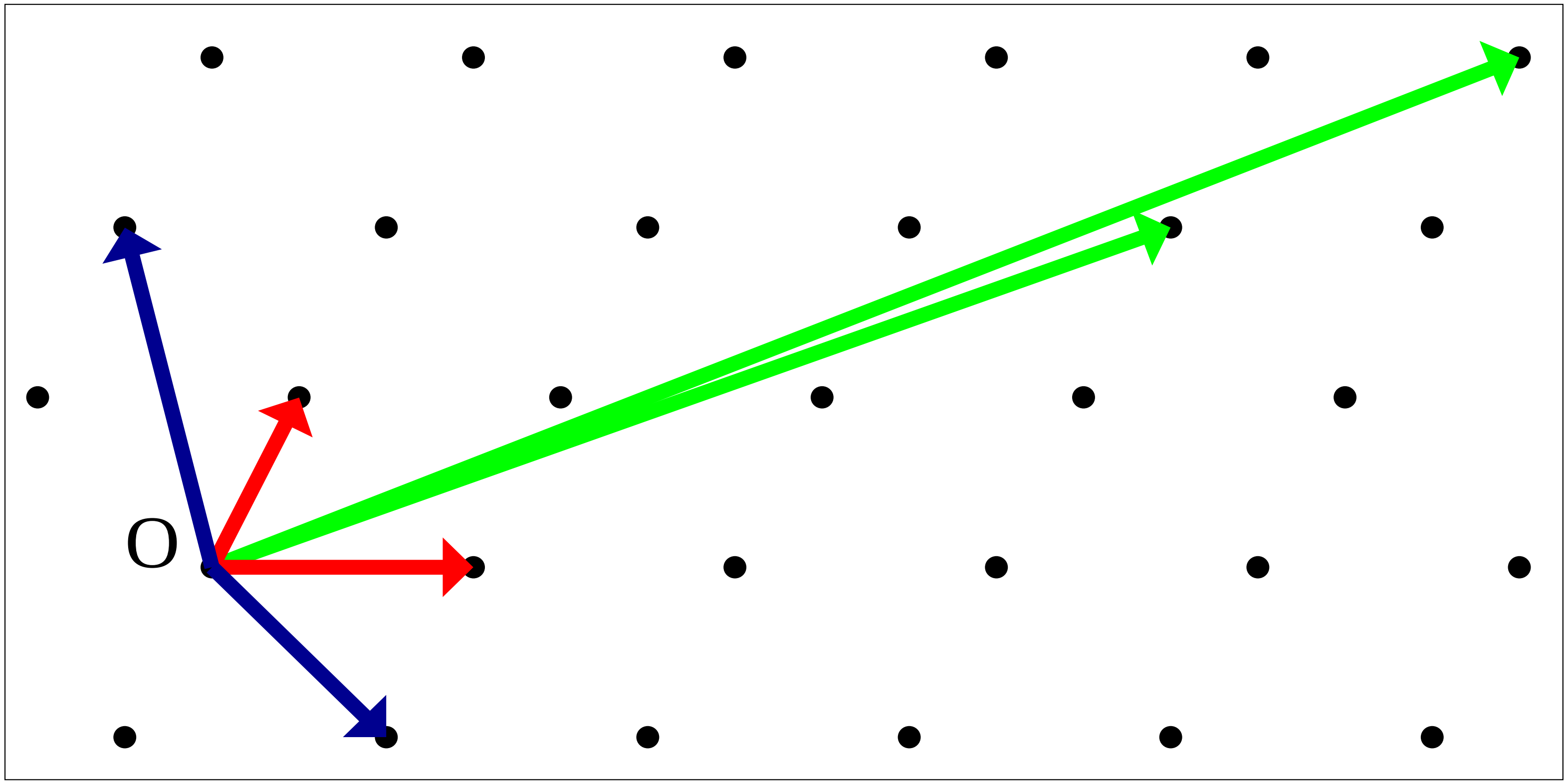
Point Lattices



- Point lattice := discrete subgroup L of $(\mathbb{R}^n, +)$.

  [discrete := for every point, there is a ball "isolating" the point].
- Basis of L := any linearly independent $\vec{b}_1, \ldots, \vec{b}_d$ such that $L = \mathbb{Z}.\vec{b}_1 + \cdots + \mathbb{Z}.\vec{b}_d$.

Infinitely many bases



Lattice invariants

- Let L be a lattice in $\mathbb{R}^n$.
- All the bases of L have the same number of elements,

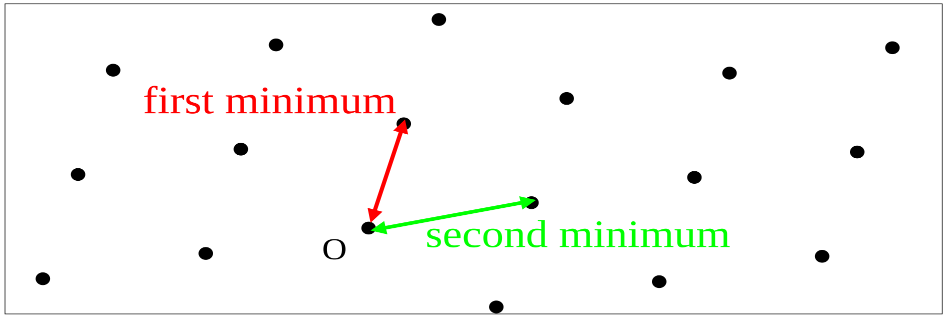
  called the dimension or rank of L.
- All the bases $\vec{b}_1, \ldots, \vec{b}_d$ of L

  have the same Gram determinant $\det\langle \vec{b}_i, \vec{b}_j \rangle$.

  Its square root is called the volume or determinant of L.
- Among all the many bases, some are more interesting than others.

  How "good" can a basis be ?

Successive Minima



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

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

  k linearly independent vectors of L, where $1 \leq k \leq d$.
- Minkowski's theorems: $\lambda_1(L) \leq \sqrt{d} \cdot \mathrm{vol}(L)^{1/d}$

  and for all $r \leq d$, $\left(\prod_{k=1}^{r} \lambda_k(L) \right)^{1/r} \leq \sqrt{d} \cdot \mathrm{vol}(L)^{1/d}$

Links between minima and bases

- In dimension ≤ 3, a set of linearly independent vectors reaching the minima is always a basis.
- This is no longer true in dimension ≥ 4.

  Let L be the lattice of vectors in $\mathbb{Z}^4$

  such that the sum of the coordinates is even.

  L has volume 2, and minima all equal to $\sqrt{2}$.

  Now consider the following row vectors :

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

Links between minima and bases

- In dimension ≤ 4, there always exists a basis reaching the minima.
- This is no longer true in dimension ≥ 5.

  Consider the lattice L of $\mathbb{Z}^5$ spanned by the following row vectors:

$$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Lattice reduction

- There always are linearly indep. vectors reaching the minima, but in general, there is no basis reaching the minima.
- There is no basis which is "obviously better" than others. Many definitions of "reduced basis" have appeared, and keep appearing.
- In general, a reduced basis satisfies the following:
  - reasonably short vectors: norm $\approx \sqrt{n}(\operatorname{vol} L)^{1/n}$.
  - almost orthogonal vectors.

The Shortest Vector Problem (SVP)

Given a basis of L, find a shortest non-zero lattice vector.

- Hard in theory. (NP-hardness for randomized reductions)
- Best polynomial-time algorithms: subexponential approximations.
- In practice, reasonably good results
  with usual lattice basis reduction algorithms.

The gap of the SVP-instance is $\frac{\|\text{second-to-shortest vector}\|}{\|\text{shortest non-zero vector}\|} = \frac{\lambda_2(L)}{\lambda_1(L)}$.

Experimental fact: the larger the gap is, the easier SVP becomes.

The Closest Vector Problem (CVP)

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

- Hard in theory. (NP-hardness even for approximation)
- Best polynomial-time algorithms: subexponential approximations.
- Easy when one has a very "good" basis and the distance is small enough (Babai).
- In practice, the best method is the embedding technique to heuristically transform CVP to SVP.

The gap of the CVP-instance is $\dfrac{\|\text{shortest non-zero vector of } L\|}{\text{dist}(\vec{c}, L)}$.

The embedding technique works better when the gap is large.

Links between SVP and CVP

- Approximating CVP $\Rightarrow$ approximating SVP with the same factor.
- Approximating SVP $\Rightarrow$ approximating CVP with a factor loss.
- More NP-hardness results for CVP.
- CVP is believed to be harder.
- In practice, SVP and CVP have about the same difficulty with the embedding technique.

The Embedding Technique

From a CVP-instance with $\vec{c}$ and a basis B of a lattice L in $\mathbb{Z}^n$, build the lattice M spanned by the rows of:

$$\begin{pmatrix} B & 0 \\ \vec{c} & 1 \end{pmatrix}$$

- A shortest vector of M "should" give a closest vector in L: $(\vec{c} - \vec{x}, 1) \in M$ is very short when $\vec{x} \in L$ minimizes $\|\vec{c} - \vec{x}\|$.
- The gap of L's CVP-instance "should" be the gap of M's SVP-instance.

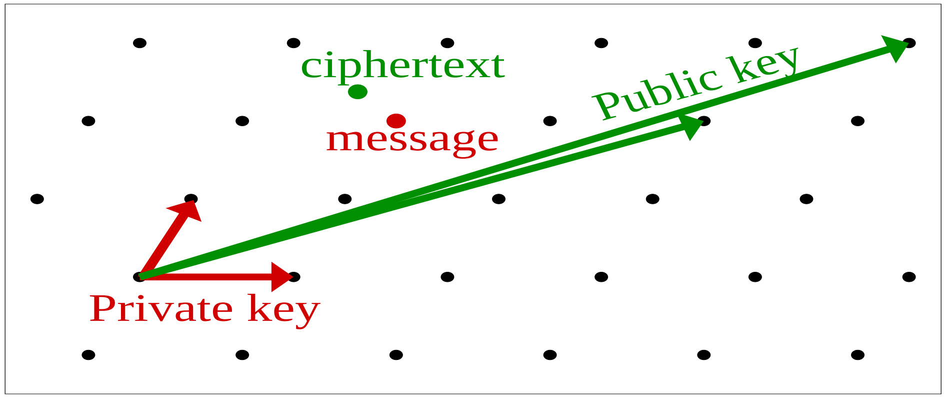
Lattice Cryptography

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

The Ajtai-Dwork Cryptosystem

- Semantically secure if SVP* is hard in the worst-case.
- Mainly of theoretical interest [NgSt98]:
  - Ajtai-Dwork is insecure for any “realistic” key.
  - One can decrypt if one can approximate CVP or SVP sufficiently well.
  - Decryption is “not” NP-hard.
- It is not clear why SVP* should be hard.

The GGH Cryptosystem



- Security parameters $n \geq 300$ and $\sigma = 3$.
- Private key := a very "good" basis of a full-rank lattice L in $\mathbb{Z}^n$.

  Public key := a "bad" basis B of L.
- Encrypt a message $\vec{m} \in \mathbb{Z}^n$ by $\vec{c} = \vec{m}B + \vec{e}$ where $\vec{e} \in_R \{\pm\sigma\}^n$.

  $\vec{c}$ is a perturbation of the lattice vector $\vec{m}B \in L$.

Security

- Finding a basis as good as the secret key is hard.
- Decryption = solving a CVP-instance.
- GGH-Challenges in dimension 200, 250, 300, 350, 400.

  Conjectured intractable if dimension ≥ 300.

  The GGH-200 challenge was solved in 1997 [Ng97] with NTL.

Efficiency

- If both the private and public key have polynomial entries, then encryption/decryption is $O(n^2 \log n)$ [RSA/ECC is $O(n^3)$].
- But the keysize is also $O(n^2 \log n)$ [RSA/ECC is $O(n)$].

Leaking Remainders [Ng99]

Recall that: $\vec{c} = \vec{m}B + \vec{e}$ where $\vec{e} \in_R \{\pm\sigma\}^n$. Therefore:

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

Even better, by letting $\vec{s} = (\sigma, \ldots, \sigma)$:

$$\vec{c} + \vec{s} \equiv \vec{m}B \mod 2\sigma.$$

Thus, if B is invertible mod 2σ, then $\vec{m}$ mod 2σ is disclosed.

And most matrices have a tiny kernel mod 6,
so one can guess $\vec{m}$ mod 2σ.

Decryption made easier

Let $\vec{m_\sigma} = \vec{m} \bmod 2\sigma$. Then $\vec{m} - \vec{m_\sigma} = 2\sigma\vec{m'}$ where $\vec{m'} \in \mathbb{Z}^n$.

$$\vec{c} = \vec{m}B + \vec{e} \Longrightarrow \vec{c} - \vec{m_\sigma}B = (\vec{m} - \vec{m_\sigma})B + \vec{e}.$$

$$\Longrightarrow \frac{\vec{c} - \vec{m_\sigma}B}{2\sigma} = \vec{m'}B + \frac{\vec{e}}{2\sigma}$$

- This is a CVP-instance with error vector $\dfrac{\vec{e}}{2\sigma}$.
- The distance is $\sqrt{n/4}$ instead of $\sigma\sqrt{n} \Longrightarrow$ larger gap.
- If we solve this CVP-instance, we can obtain $\vec{m}$.

The Attack

- Compute all the possible values of $\vec{m} \bmod 2\sigma$, from the system of linear congruences.
- For each value, try to solve the corresponding CVP-instance, using the embedding technique.

Experiments on the GGH-Challenges

Dimension	200	250	300	350	400
# solutions mod 6	2	2	6	6	1
Expected gap	9.7	9.4	9.5	9.4	9.6
Time in minutes	30	60	165	270	Unsolved

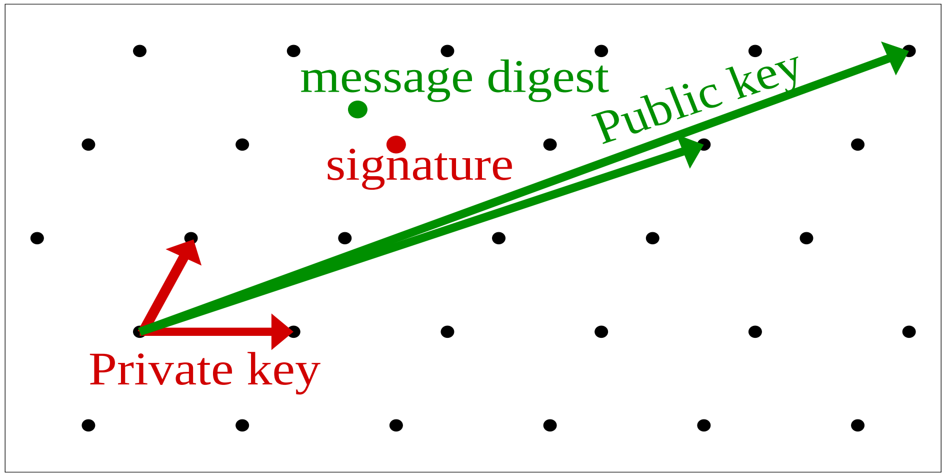
Repairing GGH

- Needs to choose an error vector $\vec{e}$
  which is not disclosed mod any number, but long enough.
- Possible choice: $\vec{e} \in_R \{-\sigma, \ldots, +\sigma\}^n$
- But the CVP-instance still has a gap.
- Would probably require dimension ≥ 400:
  - Public key ≈ 1.8 Mbytes.
  - Message expansion ≈ 17.
  - Encryption requires 2^{17} multiplications
    with numbers up to 129 bits.

Improving GGH by the HNF [Mi01]

- Choose the Hermite normal form as the public basis.
  The HNF is the one and only lattice basis H which is
  upper triangular with positive entries such that
  each diagonal element is $>$ all the elements of its column.
- Encode the message into the error vector,
  and reduce it modulo the HNF.
  Instead of $\vec{c} = \vec{m}B + \vec{e}$ where $\vec{m}$ is in a large cube,
  one takes $\vec{c} = \vec{m} \bmod H$ where $\vec{m}$ is small.
- Keysize is provably $O(n^2 \log n)$: not competitive with RSA/ECC.

The GGH Signature Scheme



- Security parameters $n \geq 300$ and $\sigma = 3$.
- Private key $:=$ a very "good" basis of a lattice L in $\mathbb{Z}^n$.

  Public key $:=$ a "bad" basis B of L.
- A message digest is mapped into a space point $\vec{m}$.

  A signature of $\vec{m}$ is a lattice point sufficiently close to $\vec{m}$.

Generating signatures

- Let L be a lattice spanned by a good basis (the matrix G).
  $L = \{\vec{z}G : \vec{z} \in \mathbb{Z}^n\}$.
- Signing is done with Babai's CVP-approximation algorithm.
  To sign the message digest $\vec{m}$:
  compute $\vec{y} \in \mathbb{Q}^n$ such that $\vec{m} = \vec{y}G$,
  and round $\vec{y}$ to an integer vector $\vec{x} \in \mathbb{Z}^n$.
  The signature is $\vec{s} := \vec{x}G$.

Security problem [GeSz01]

- Assume that one signs many messages $\vec{m}_i$.
- If the message digests $\vec{m}_i$'s are random, one can argue that $\vec{x}_i - \vec{y}_i$ is uniformly distributed over $[-1/2, +1/2]^n$.

  From the signatures $\vec{s}_i$'s, one derives vectors of the form $\vec{\varepsilon}_i G$ where $\vec{\varepsilon}_i$ is uniformly distributed over $[-1/2, +1/2]^n$.
- One can compute $G^\top \vec{\varepsilon}_i{}^\top \vec{\varepsilon}_i G$.

  Interestingly, the mean of $\vec{\varepsilon}_i{}^\top \vec{\varepsilon}_i$ is $\frac{1}{12} I$.
- Signatures eventually leak the Gram matrix $G^\top G$:

  The GGH-signature is not zero-knowledge.

Breaking the GGH signature ? [Sz02]

- Does the knowledge of $G^{\top}G$ help lattice reduction ?
  If one knows $G^{\top}G$ and HNF(G),
  can one recover G or another very reduced basis ?
- [Sz02] shows that this can be solved in poly-time,
  if one is given access to a distinguishing oracle:
  given integer matrices $A^{\top}A$ and $B^{\top}B$,
  decide if the lattices spanned by A and B are isomorphic.

The NTRU Cryptosystem (1996)

- Let $\mathcal{P}$ be the ring $\mathbb{Z}[X]/(X^N - 1)$ where N is a "small" prime: $N := 251$, 347 or 503.

  $\mathcal{P}$ is identified with the set of integer polynomials of degree $< N$.
- The multiplication $*$ in $\mathcal{P}$ is a convolution product.

  Convolutions can easily be computed thanks to $X^N - 1$.
- For a and b in $\mathcal{P}$, we write $\mathsf{a} \equiv \mathsf{b} \pmod{p}$ when the coefficients are pairwise congruent modulo p.

NTRU Parameters

- Let $\mathcal{S}$ be a subset of sparse polynomials with coeffs 0 and ± 1.

  q is a small power of 2, typically 128.

  p is a small odd number, typically 3.

- Private key := f and g in $\mathcal{S}$ such that

  f is invertible mod p and q:

  $\mathsf{f} * \mathsf{f}_p \equiv 1 \pmod{p}$ and $\mathsf{f} * \mathsf{f}_q \equiv 1 \pmod{q}$.

- Public key $\mathsf{h} := \mathsf{g} * \mathsf{f}_q \pmod{q}$. Note that $\mathsf{f} * \mathsf{h} \equiv \mathsf{g} \pmod{q}$.

The NTRUEncrypt primitive (1996)

- A message m is an element of $\mathcal{P}$ with coeffs 0 or ± 1.

  m is encrypted into $\mathsf{e} := \mathsf{m} + p\mathsf{r} * \mathsf{h} \pmod{q}$ where $\mathsf{r} \in_R \mathcal{S}$.
- To decrypt e, notice that $\mathsf{e} * \mathsf{f} \equiv \mathsf{m} * \mathsf{f} + p\mathsf{r} * \mathsf{g} \pmod{q}$.

  If the reduction is centered, this "should" be an equality over $\mathcal{P}$.

  By taking residues modulo p and dividing by f, one recovers m.
- For the recommended parameters, the decryption may fail,

  but the failure probability seems to be negligible.
- Encryption and decryption cost $O(N^2 \ln q)$. [RSA/ECC is $O(n^3)$]

  Keysize is $O(N \ln q)$ [RSA/ECC is $O(n)$].

  Ciphertext expansion is $O(\ln q)$.

Modifications to NTRU (2000)

- Replace $p = 3$ by the small polynomial $p = X + 2$.

  Ternary polynomials become binary polynomials.
- Special form for sparse polynomials: $\mathsf{f}, \mathsf{g}, \mathsf{r}$.

  For instance, $\mathsf{r} = \mathsf{r}_1 * \mathsf{r}_2 + \mathsf{r}_3$.
- These changes improve the efficiency, but may affect the security.

  Nevertheless, this is the version specified in NTRU standards.

Lattice attack on NTRU [CoSh97]

Let L be the lattice of all $(u, v) \in R^2 \simeq \mathbb{Z}^{2N}$ s.t. $u * h = v \pmod q$.
If $h(X) = h_0 + h_1 X + \cdots + h_{N-1} X^{N-1}$, the HNF of L is:

$$\begin{pmatrix}
1 & 0 & \dots & 0 & h_0 & h_1 & \dots & h_{N-1} \\
0 & 1 & \ddots & \vdots & h_{N-1} & h_0 & \dots & h_{N-2} \\
\vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & \vdots \\
0 & \dots & 0 & 1 & h_1 & \dots & h_{N-1} & h_0 \\
0 & \dots & \dots & 0 & q & 0 & \dots & 0 \\
\vdots & & & \vdots & 0 & q & \ddots & \vdots \\
\vdots & & & \vdots & \vdots & \ddots & \ddots & 0 \\
0 & \dots & \dots & 0 & 0 & \dots & 0 & q
\end{pmatrix}$$

Lattice attack on NTRU

- L is a full-rank lattice: $\mathrm{rank}(L) = 2N$.
- $\mathrm{vol}(L) = q^N$, so one might expect that a short vector is $\approx \sqrt{2Nq}$.
- The target $(f, g) \in L$: it should be a shortest vector of L,

  but it is not an exceptionnally short vector.
- NTRU claims that such lattice attacks are exp. in N.

  The old NTRU-167 challenge is still unsolved.

  New NTRU challenges start with $N = 251$.

Improved lattice attacks ?

- Since f and g are sparse, one may slightly reduce the rank [MS01].
- If N is not prime, the rank can be reduced because of non-trivial factors of $X^N - 1$. [Ge01]
- Attempts to divide the rank by two using properties of R, but the new lattices seem to be much denser.

The NTRUSign scheme (2000–02)

- NTRU has tried to develop a signature scheme.
  Almost all versions have been broken [Mi00,JGSS01,GeSz02].
- A new version, called NTRUSign, has recently been proposed.
  It follows the GGH design:
  the signer computes a reduced compact basis of the NTRU lattice (basis consisting of 4 circulant matrices).
  But like in GGH, signatures leak the Gram matrix:
  security becomes suspicious.
  Besides, N is chosen so that $N - 1$ is highly smooth.

Conclusion

- Among existing lattice cryptosystems, only NTRU is competitive, but it is still too early to tell whether NTRU is secure or not.
- 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 250.
- Hardness of SVP/CVP needs to be better understood. In particular, there has not been any large experiments.
- A "convincing" lattice signature scheme has yet to be found.