Lattice-based cryptography: Enumeration of vectors for SVP

Artur Mariano
Institute for Scientific Computing
TU Darmstadt
artur.mariano@sc.tu-darmstadt.de

Agenda

- Lattices
- Lattice-based cryptography (LBC)
- Problems in LBC
    - Enumeration algorithms
- The project

Notation

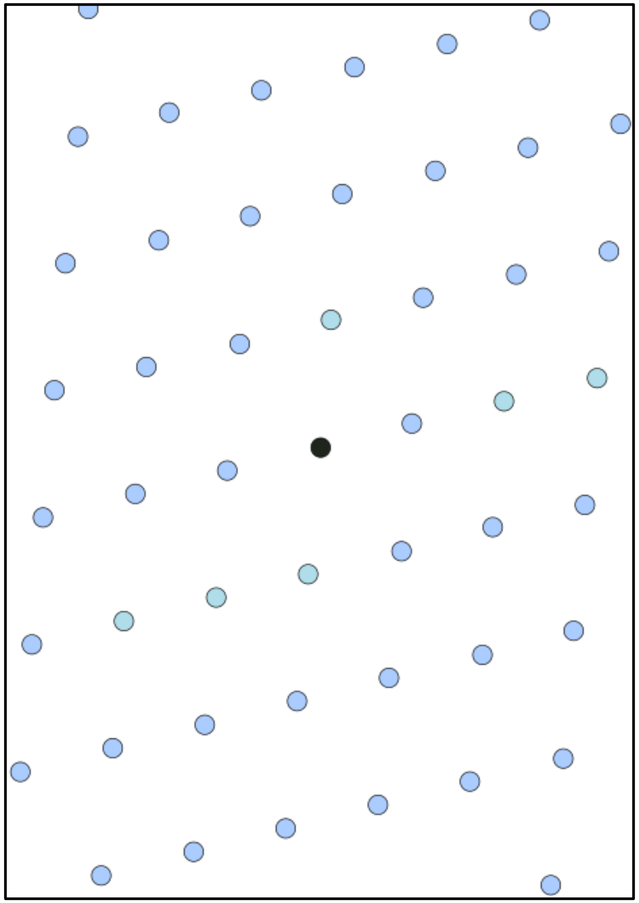
- Vectors are always in bold face and never capitalized
  - Might appear in italic due to the MPP equation feature
- Matrices are always in bold face and capitalized
  - Might appear in italic due to the MPP equation feature
- Lattices are represented by Λ and bases by Ḃ
- ||**v**|| represents the Euclidean norm of a vector **v**
  - Distance spanned from the origin to the point given by **v**

Lattices

- A lattice Λ is generated by a basis Ḃ
  - Set of linearly independent vectors;
  - Lattice points are linear combinations of vectors in Ḃ with integer coefficients:

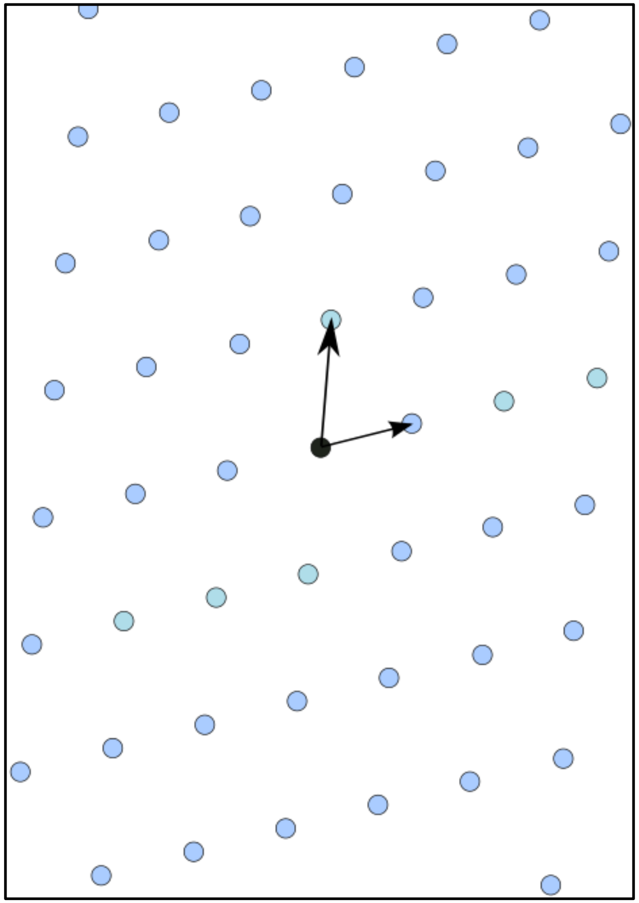
$$\Lambda = \mathbf{B}Z = \sum_{i=0}^{n} \boldsymbol{z}_i \boldsymbol{b}_i,\ \boldsymbol{z}_i \in \mathbb{Z}$$

where **B** is the matrix with column vectors (matrices k $*$ 1) $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$, for a basis with *n* vectors.



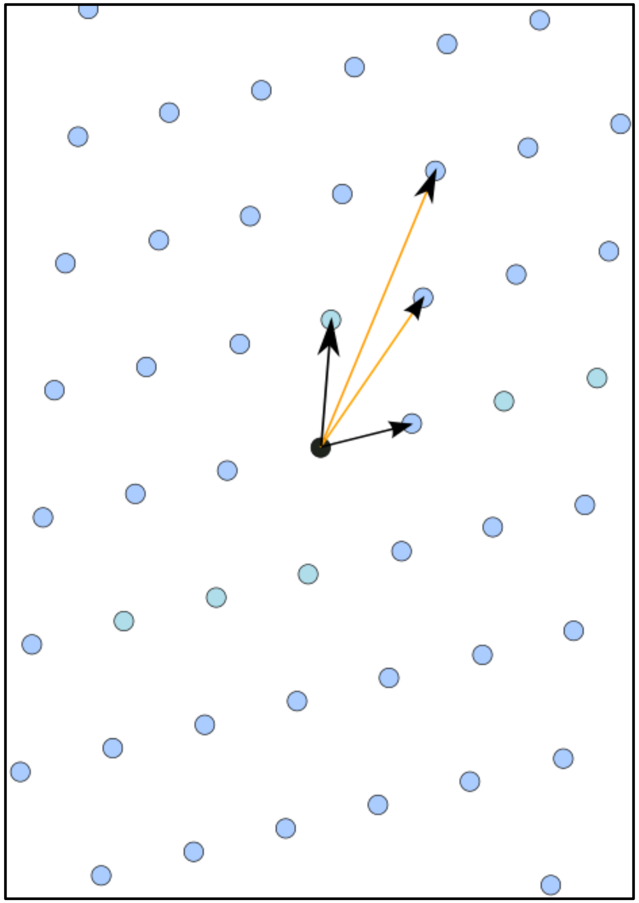
Lattices

- Let us assume a Basis Ḃ with 2 vectors drawn in the picture on the right side



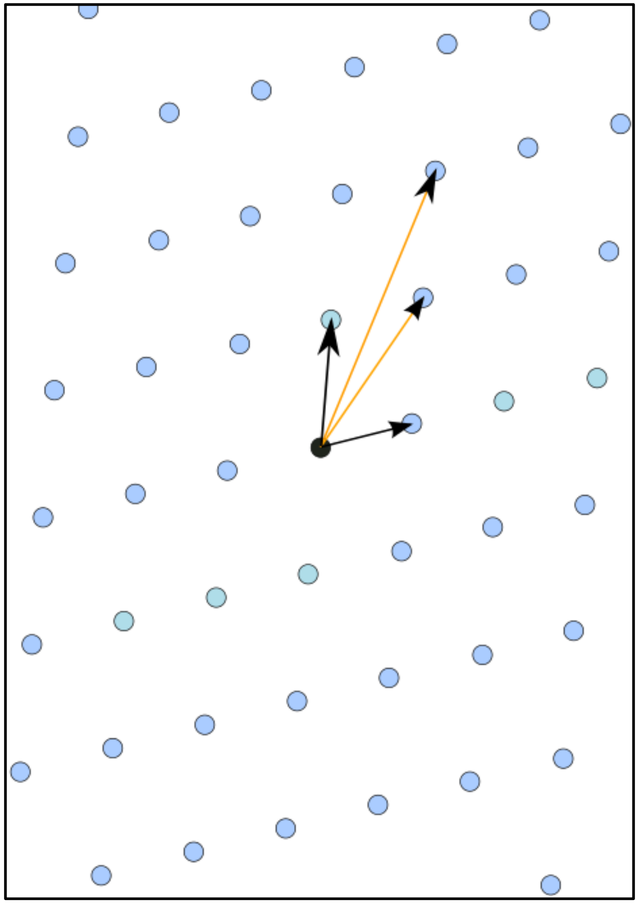
Lattices

- Let us assume a Basis Ḃ with 2 vectors drawn in the picture on the right side
- The same lattice has generally more than one possible basis!



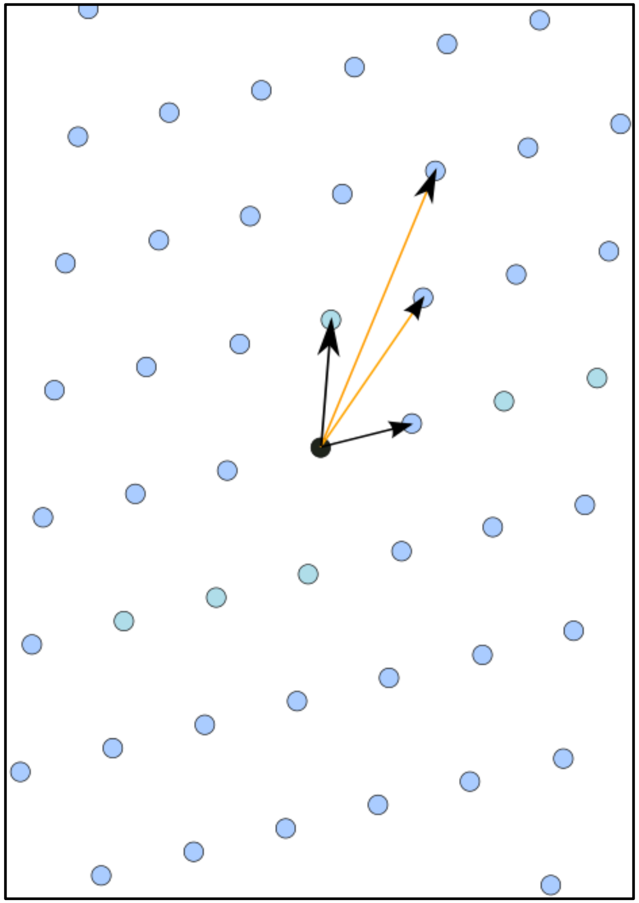
Lattices

- Let us assume a Basis Ḃ with 2 vectors drawn in the picture on the right side
- The same lattice has generally more than one possible basis!
  - Solve problems in reduced (shorted and orthogonal) bases is simpler



Lattices

- Let us assume a Basis Ḃ with 2 vectors drawn in the picture on the right side
- The same lattice has generally more than one possible basis!
  - Solve problems in reduced (shorted and orthogonal) bases is simpler
  - Solvers of several problems call basis reduction algorithms before executing



Lattice-based cryptography

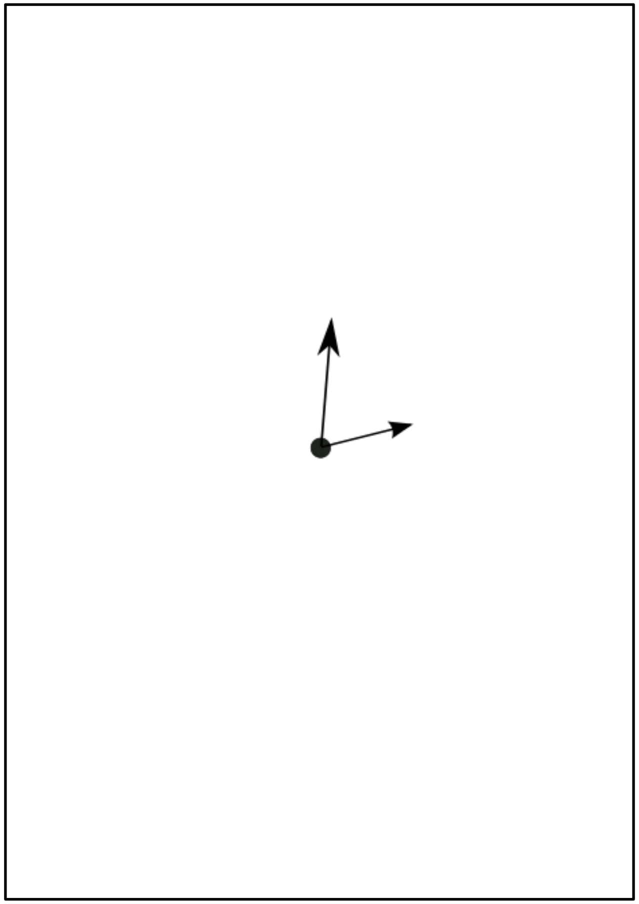
TECHNISCHE UNIVERSITÄT DARMSTADT

- Current cryptographic schemes (e.g. RSA) become vulnerable in the presence of quantum computers
  - This poses real risk, as you might guess!
- Lattices benefit from unique, interesting properties for cryptography
  - The most proeminent type of quantum-resistant cryptography
    - Chances are that this will be the standard type of cryptosystems!
  - NP-Hard problems, that are used as the underlying mathematical problems
  - The average-case of lattice problems is still hard to solve
  - Enables the use of fully homomorphic encryption, the holy grail of crypto

SCIENTIFIC COMPUTING

Lattices in LBC

- Lattices in $\mathbb{R}^n$ whose basis Ḃ has n elements are called full-rank lattices
  - The most common type in LBC;
- It is also common to work with integer lattices in LBC
  - Whose problems are proved to be as hard as in floating-point lattices
  - Easier to work computationally



Problems in LBC

- Lattice-based cryptosystems become vulnerable only if specific lattice problems are solved in a timely manner
  - One of which is to find the shortest non-zero vector(s) in a given lattice, referred to as the Shortest Vector Problem (SVP)
  - The SVP is known to be NP-hard in random reductions
    - No polinomial time algorithms are expected to be found
- The shortest vector problem is, in lattice based cryptography, the most relevant problem:

$$\text{find } ||\mathbf{s}|| < ||\mathbf{p}||, \forall \mathbf{p} \in \Lambda, \mathbf{s} \in \Lambda$$

Problems in LBC

- It might be enough to solve an approximation of SVP (aSVP) if lattice based cryptosystems are to be broken
- SVP are still of vital importance since they are used in aSVP solvers, as a way of improving the final solution
- There are virtually no aSVP solvers, lattice-reduction algorithms are used instead when solving the aSVP
  - and these use SVP solvers as part of their logic!

Science on LBC

- Relatively recent, yet radiply growing field
- Large number of groups working on LBC -> a lot of papers published
- Led to the creation of challenges to announce what can be broken

HALL OF FAME

Position	Dimension	Euclidean Norm	Seed	Contestant	Solution	Algorithm	Subm. Date	Approx. Factor
1	138	3077	0	Kenji KASHIWABARA and Tadanori TERUYA	vec	Other	2014-12-7	1.03516
2	134	2976	0	Kenji KASHIWABARA and Tadanori TERUYA	vec	Other	2014-07-13	1.01695
3	132	3012	0	Kenji Kashiwabara and Masaharu Fukase	vec	Other	2014-04-24	1.03787
4	130	2883	0	Yoshinori Aono and Phong Nguyen	vec	ENUM,BKZ	2014-10-9	0.99871

Enumeration techniques for the SVP*

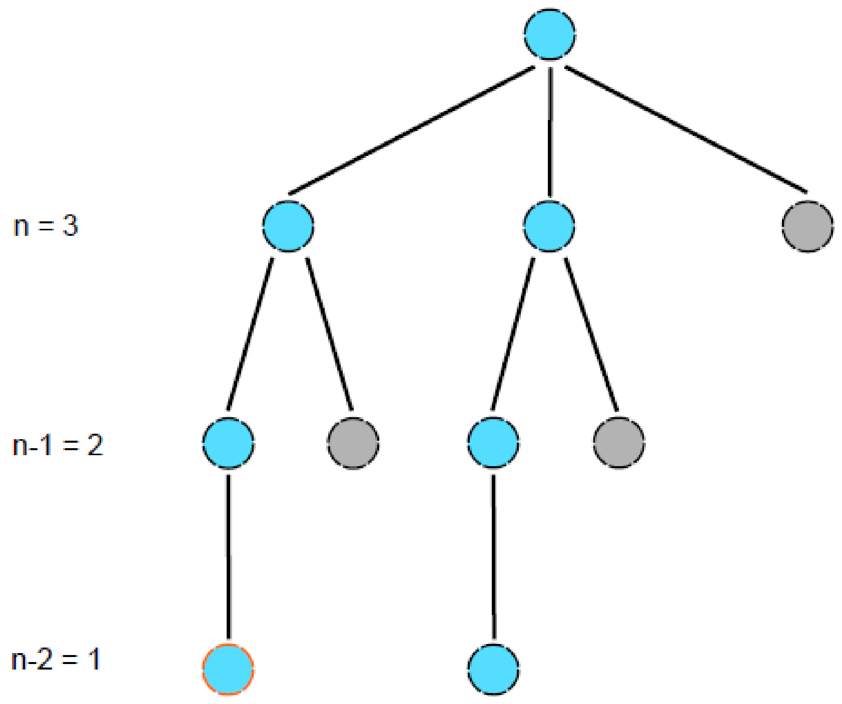
- Exaustive search algorithms with exponential time complexity but polynomial space complexity
  - Enumeration of all possible vectors within a ball around the origin
    - Depth search in a tree with the resultant vectors
- Extreme prunning techniques make of these algorithms <u>the faster in practice</u>
- Highly parallel
  - Implemented in CPU-chips, GPUs and FPGAs with quasi linear speedups
  - No implementations for heterogenous sys., no vectorized code, etc...

*we do try to break systems in Darmstadt, sorry! ☺

Enumeration

```
i = 1;
C = ||b_1||^2;
while true do
    computeSqrDist();
    if sqrDist_i < C then
        if i > 1 then
            moveDown();
        else
            updateC();
            updateBestVector();

        end
    else
        if i = n then
            return bestVector;
        else
            moveUp();
        end
end
```



Enumeration: what is it all about?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Let us focus on the algorithm rather than on the math
- The search is mapped onto a (virtual) search tree
  - The algorithm goes up and down on the tree, acording to some criteria
  - The levels of the tree represent parts of the final vector
  - Leaves are complete vectors, but the enumeration might abort at some early point on the branch and move onwards (to another branch or sibling)
  - Unbalanced tree: on CPUs, the problem is easy to solve as long as the enumeration tree is correctly balanced among the running threads
    - Problem was solved, with moderate success; we refer to [DS10]

SCIENTIFIC COMPUTING

Enumeration: GPUs and HetPlats

- If CPU code scales at a moderate rate, GPUs might be suited!
  - Trick is to choose operators that can be applied to many (active) nodes
  - Hint: a data driven approach might be useful;
    - Examples from domains with graphs (attend Cristiano's talk today!)
- And if both work, why not to think about heterogeneous CPU+GPU platforms?
  - Frameworks available (e.g. StarPU), although hand tuned code is desired; performance does matter in crypto!

The project

- Enumeration with pruning is the most efficient technique to solve the SVP
- Suboptimal solutions have been proposed to balance the tree
  - Very few details were given on the implementation
  - No heterogenous, high performance versions are known
- (1) Implement a parallel version of the algorithm for shared-memory CPUs
- (2) Port that implementation to GPUs
- (3) Implementation of a CPU+GPU version of the code (hand-tuned)

Working and living in Darmstadt

- Library for High Performance lattice algorithms
  - Lattice Unified Set of Algorithms (LUSA)
- Carry out thesis works in Darmstadt
  - Fábio Correia

Questions

Ask everything you want, even if it looks random!

References

- Images based on the presentations of Panagiotis Voulgaris and Fábio Correia
  - http://cseweb.ucsd.edu/~pvoulgar/files/

- [M11] Milde B. et al., “A Parallel Implementation of GaussSieve for the Shortest Vector Problem in Lattices”, Lecture Notes in Computer Science Volume 6873, 2011, pp 452-458; [A02] Agrell, E.; Eriksson, T.; Vardy, A.; Zeger, K., "Closest point search in lattices," *Information Theory, IEEE Transactions on* , vol.48, no.8, pp.2201,2214, Aug 2002
- [M10] Micciancio, Daniele and Voulgaris, Panagiotis, “A Deterministic Single Exponential Time Algorithm for Most Lattice Problems Based on Voronoi Cell Computations”, Proceedings of the 42Nd ACM Symposium on Theory of Computing, 2010