# Lattice Problems

Daniele Micciancio
UC San Diego

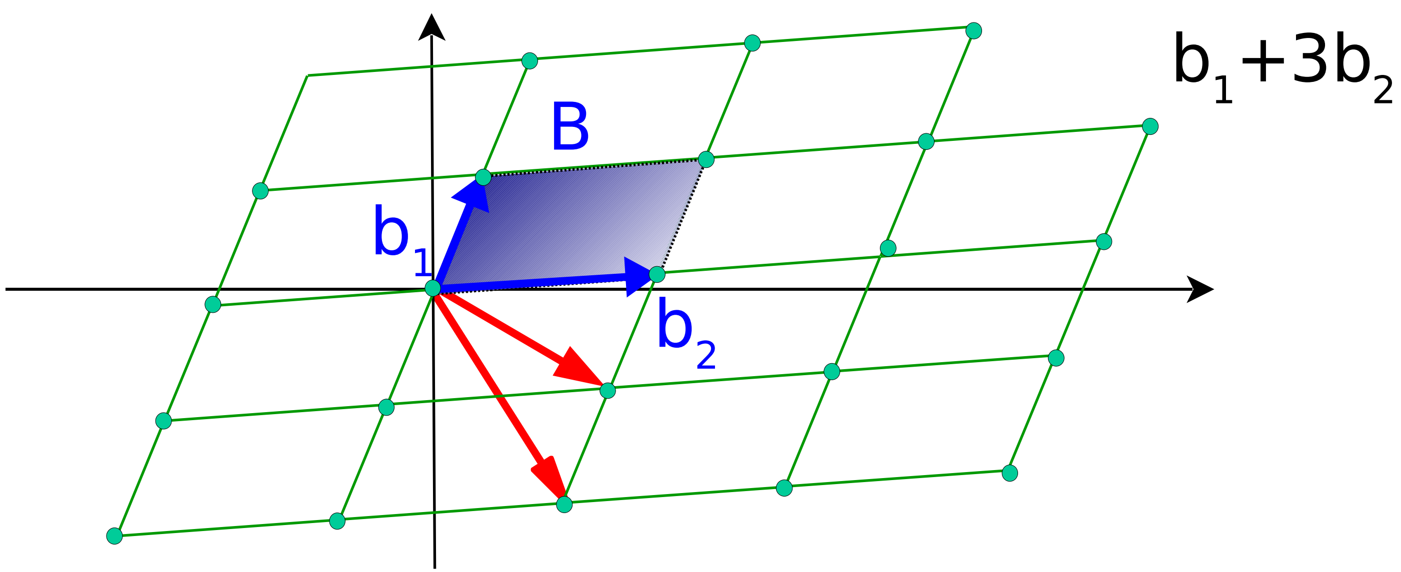
TCC 2007 Special Event:
Assumptions for cryptography

# Outline

- Lattice Problems
  - Introduction to Lattices, SVP, SIVP, etc.
- Cryptographic assumptions
  - Average-case vs. worst-case complexity
- Example Application
- Issues/Discussion
  - Choosing security parameters
  - Using lattices with special properties

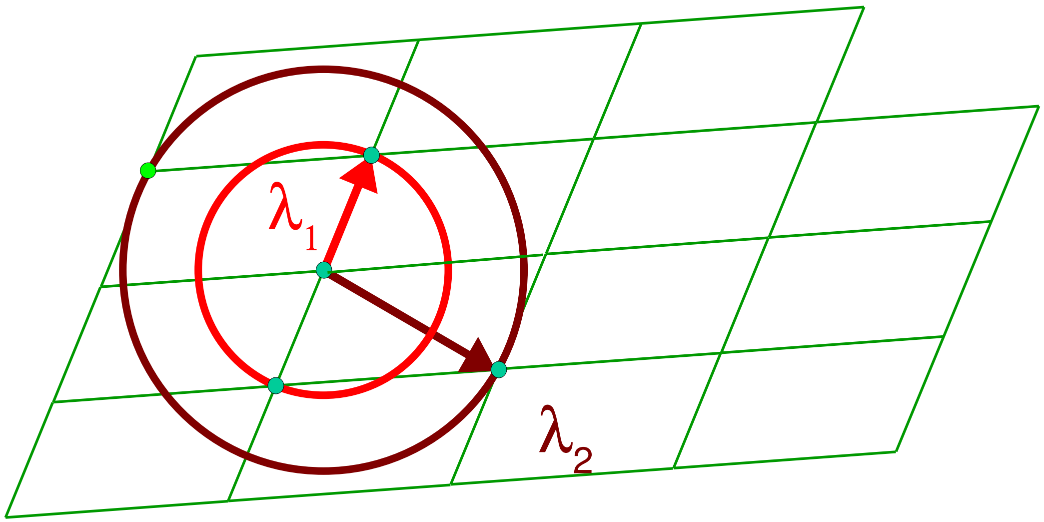
# Point Lattices

- Set of all integer linear combinations of basis vectors $B = [b_1,\ldots,b_n] \subset R^n$
- $L(B)=\{Bx: x \in Z^n\} \subset \text{span}(B)=\{Bx: x \in R^n\}$



# Successive Minima

- For every n-dimensional lattice L, and *i*=*1*,…,*n*, the *i*th successive minimum $\lambda_i(L)$ is the smallest radius r such that Ball(0,r) contains *i* linearly independent lattice vectors



# Lattice problems

- Shortest Vector Problems (SVP)
  - Given a lattice $L$, find the nonzero lattice vector $v$ closest to the origin ($\|v\| \leq \gamma \lambda_1(L)$)
- Shortest Independent Vect. Prob. (SIVP)
  - Given a lattice $L$, find n lin. independent vectors $v_1,\ldots,v_n$ of length $\max_i \|v_i\| \leq \gamma \lambda_n(L)$
- Approximation factor $\gamma(n)$ usually a function of the lattice dimension n.

# More lattice problems

- Closest Vector Problem (CVP):
  - Given lattice $L$ and target point $t$, find lattice vector $v$ closest to $t$: $||v - t|| \leq \gamma \,\mathrm{dist}(t,L)$
- Bounded Distance Decoding (BDD):
  - CVP with promise that $\mathrm{dist}(t,L) < \lambda_1(L)/2$
- Covering Radius Problem (CRP):
  - (Approximately) compute $\rho(L)=\max_t \mathrm{dist}(t,L)$
- ... but no *bilinear generalized decisional gap longest uber sublattice problem*, yet.

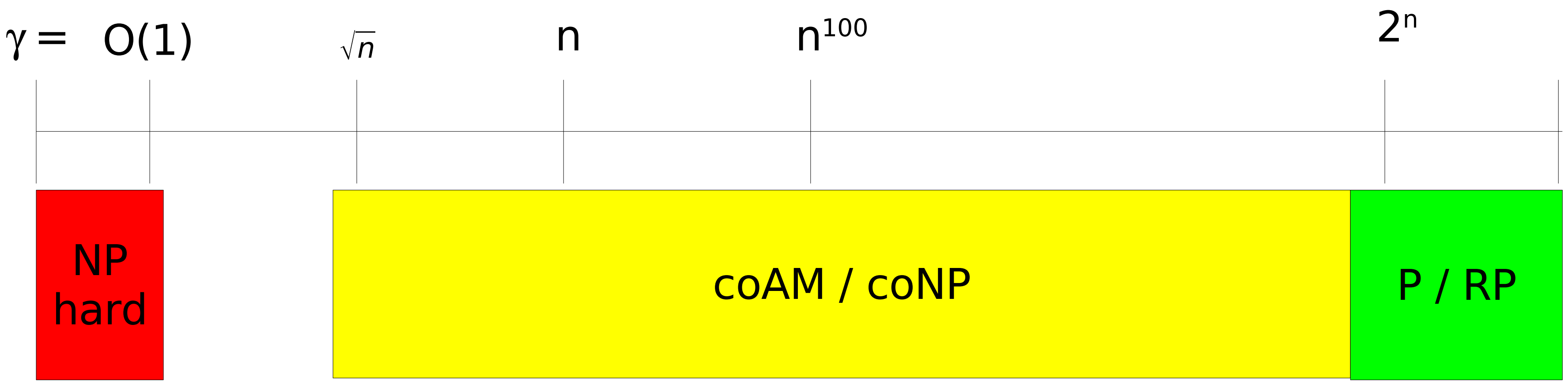
# Relations among problems

- Approximation preserving reductions
  - $SVP_\gamma$ reduces to $CVP_\gamma$ [GMSS]
  - Also, approx. $\lambda_1$ reduces to approx. dist(t,L)
- Exact solution [K, BS]
  - SVP reduces to computing $\lambda_1$
  - CVP reduces to computing dist(t,L)
  - Computing dist(t,L) reduces to $\lambda_n(L)$
- Approximate reductions [K]
  - $CVP_{\gamma'}$ reduces to $SVP_\gamma$ where $\gamma' = \mathrm{poly}(\gamma, n)$

# Open problems

- Reduce search to decision
  - Reduce $SVP_\gamma$ to approximating $\lambda_1$
  - Reduce $CVP_\gamma$ to approximating dist(t,L)
- Missing reductions
  - Reduce $CVP_\gamma$ to $SIVP_\gamma$
  - Reduce approx. $\lambda_n(L)$ to approx. dist(t,L)
- Remark
  - $\lambda_n(L)$ --> $SIVP_\gamma$ --> $CVP_\gamma$ -?-> dist(t,L)

# Complexity of SVP, SIVP, CVP

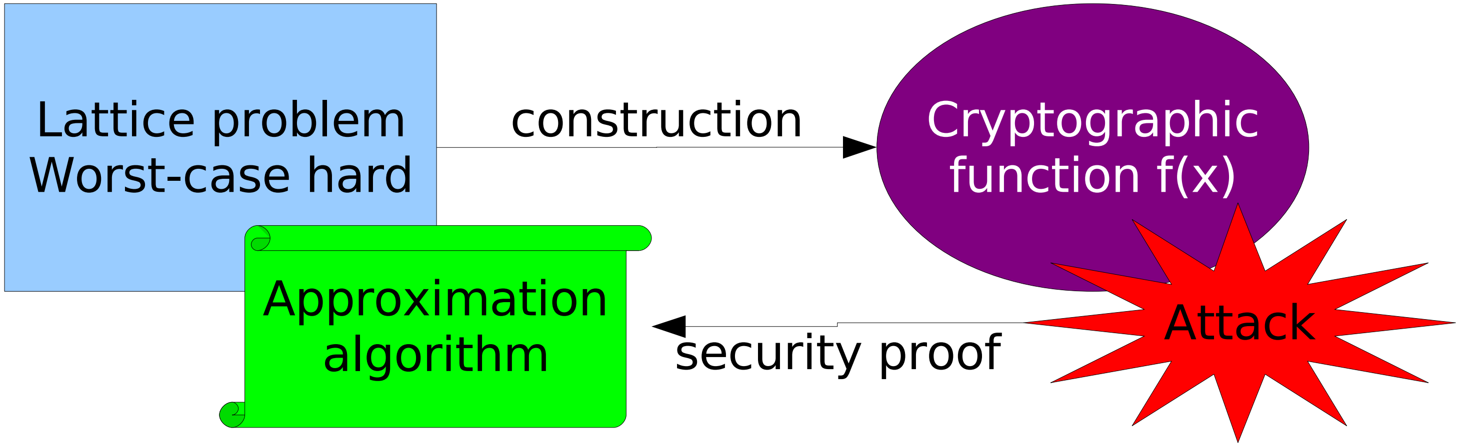



- NP-hard [vEB, Aj, ABSS, M, BS, K]
- coAM, coNP [GG, AR, GMR]
- P, RP [LLL, S, AKS]
- Open problem: $\gamma = n^{O(1)}$ factors

# Cryptographic Assumption

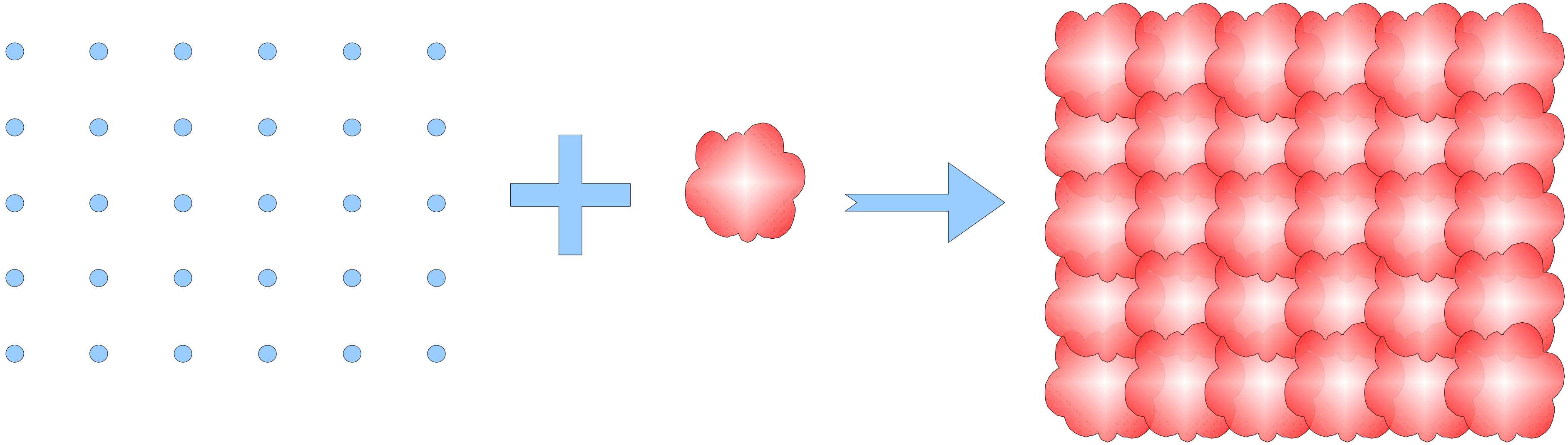
- NP-hardness for cryptography
  - Unnecessary: NP = P U NPC implies P=NP
  - Insufficient: need average-case hardness
- Cryptographic assumption:
  - SIVP is hard to approximate within $\gamma = n^c$ [Aj]
  - Best to date $\gamma = \omega(n \log(n))$ [MR]
- Remarks
  - Worst-case hardness assumption
  - Still implies cryptographic applications

# How to use lattices in cryptography




- Assumption: SIVP is worst-case hard
- Application: cryptographic function
- Proof of security:
  - Assume can break (e.g., invert) random f(x)
  - Use attack to solve SIVP on <u>any</u> lattice

# Intuition



LATTICE

random noise

$\mathbf{R}^n$

Every point in $R^n$ can be written as the sum

$$a = v + r$$

of a lattice point $v$ and small error vector $r$

# Lattice based Hash function (oversimplified version)

- Construction:
  - Key: random points $a_1,\ldots,a_m$ in $R^n$
  - Function: $f_A(x_1,\ldots,x_m) = \Sigma_i \, a_i x_i$ ,  ($x_i$ in $\{0,1\}$)
  - $f_A : \{0,1\}^m \dashrightarrow R^n$
- Technical problem
  - Range $R^n$ is infinite, so $f_A$ never compresses
  - Problem can be solved using $Z_M{}^n$ instead of $R^n$

# Security proof

- Proof of security:
  - Generate random key as $a_i = v_i + r_i$ (i=1,...n)
  - Find a collision $f_A(x_1,\ldots,x_m) = f_A(y_1,\ldots,y_m)$
  - Notice: $\Sigma_i\, a_i\, x_i = \Sigma_i\, a_i\, y_i$
- Substituting $a_i = v_i + r_i$ and rearranging:

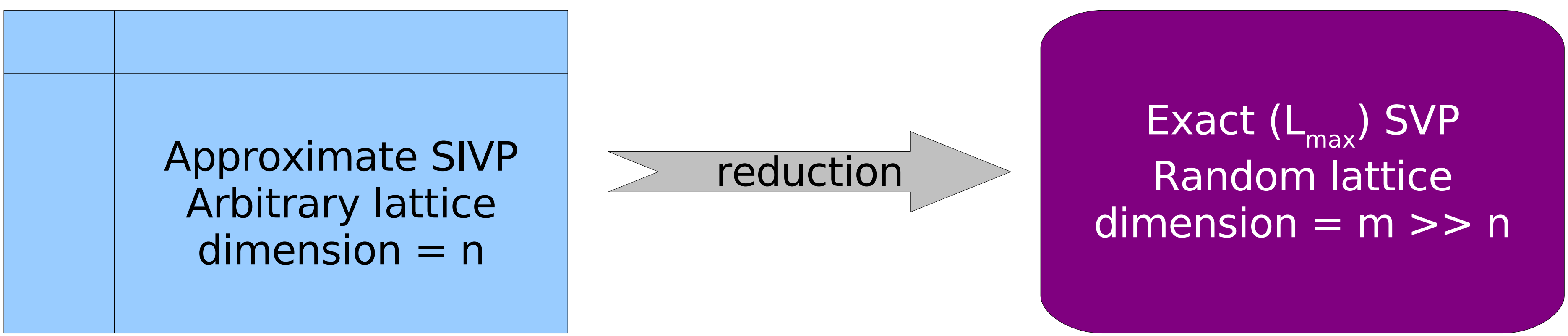
$$\Sigma_i\, v_i\, (x_i - y_i) = \Sigma_i\, r_i\, (y_i - x_i)$$

Lattice vector

short vector

# Worst-case/Average-case connection

- The set $L = \{z \text{ in } Z^m \mid f_A(z)=0\}$ is a lattice
- Collisions: z=x-y in $L$ of norm $||z||_{max} = 1$
- Security proof:




Worst-case
complexity assumption

Average-case
cryptanalysis

# Setting security level

- Choose n large enough so that SIVP is hard to approximate
  - Worst-case hard is enough for security
  - How do we generate hardest (worst-case) challenge instances?
- Choose m large enough so that SVP is hard on average
  - Easy to generate meaningful challenges
  - But then, why prove security at all?

# How to falsify worst-case assumptions

- Algorithmic approach
  - Cryptanalyst comes up with SVP algorithm, and proves it achieves $\gamma$ approximation
  - Too much burden on cryptanalyst?
- Reverse challenge approach
  - Cryptanalyst comes up with SVP algorithm, and claims it achieves $\gamma$ approximation
  - Cryptographer gives counterexample showing the algorithm does not achieve $\gamma$
- Generic model for lattices?

# “Abstract” provable security

- Security proof as a qualitative statements
  - Attacks can be avoided by increasing security parameter
  - No conceptual security flaw in cryptographic function
  - Tell us what distribution should be used
- Use traditional cryptanalysis to determine suitable security parameters

# Summary

- Classic lattice assumptions (SVP, CVP)
  - All polynomially related up to polynomial factors
  - Minor issue: decision ($\lambda_1$) vs. search (SVP)
  - Main issue: determine concrete *worst-case* hardness bounds
- Next: “ad-hoc” lattice assumptions
  - Hardness of SVP, SIVP, etc. for special classes of lattices

# Other cryptographic primitives

- Public key encryption [AD, R]
  - Requires planting a trapdoor for decryption
  - Can be done by using lattices where $\lambda_1 << \lambda_2$
- Unique SVP (uSVP)
  - Solve SVP on special class of lattices such that $\lambda_1 << \lambda_2$
  - Still worst-case assumption, but over smaller class of lattices

# Faster cryptographic functions

- Subset-sum function $f_A(x_1,\ldots,x_m) = \Sigma_i\, a_i x_i$
  - Key size and time complexity: $|A| > mn > n^2$
- Generalized compact knapsack [M,LM,PR]
  - Use polynomial ring $Z[X]/(X^n-1)$ instead of $Z$
  - Key size and time complexity is $O(n \log n)$
  - Hard to invert on the average, based on worst-case hardness of SIVP over cyclic lattices

# Worst-case assumptions for lattices with special structure

- Geometric structure
  - E.g., $\lambda_1 << \lambda_2$
  - Application: embed trapdoor for PKE
- Algebraic structure
  - E.g., $Rot(L) = L$
  - Application: more efficient functions
- Question
  - Are these legitimate assumptions? Can we still call them "worst-case"?

# Conclusion

- Lattice based cryptography
  - Only requires worst-case hardness of underlying problem
  - Classic assumptions are fairly standard
- Less standard (ad-hoc) assumptions
  - Motivated by cryptographic applications or efficiency considerations
  - Worst-case assumptions for lattices with special structure

# Things I didn't talk about

- Cryptographic functions based on average-case lattice problems
  - E.g., [GGH], NTRU
- Unconditionally secure constructions
  - Zero-Knowledge proofs for SVP, CVP [MV]
- CVP with preprocessing [M,FM,R,AKKV]
  - Fixed lattice, only target is part of input
  - Interesting for efficient cryptography
- Quantum complexity assumptions [R]