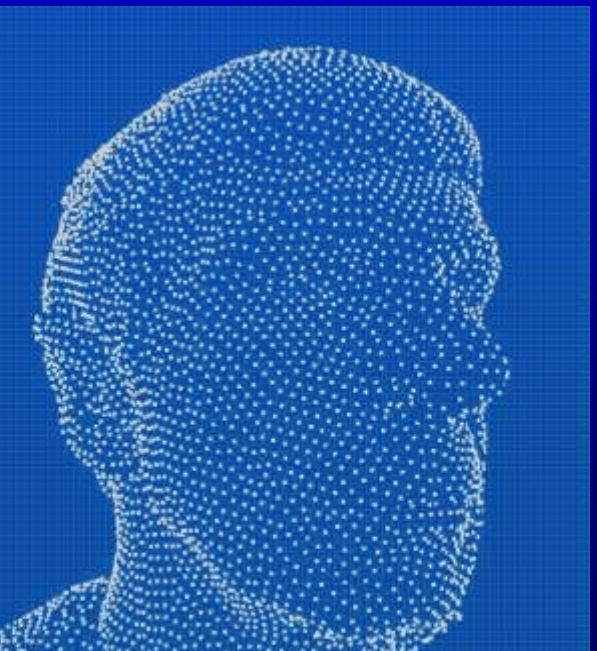
Winter School on Lattice-Based Cryptography and Applications
Bar-Ilan University, Israel 20/2/2012

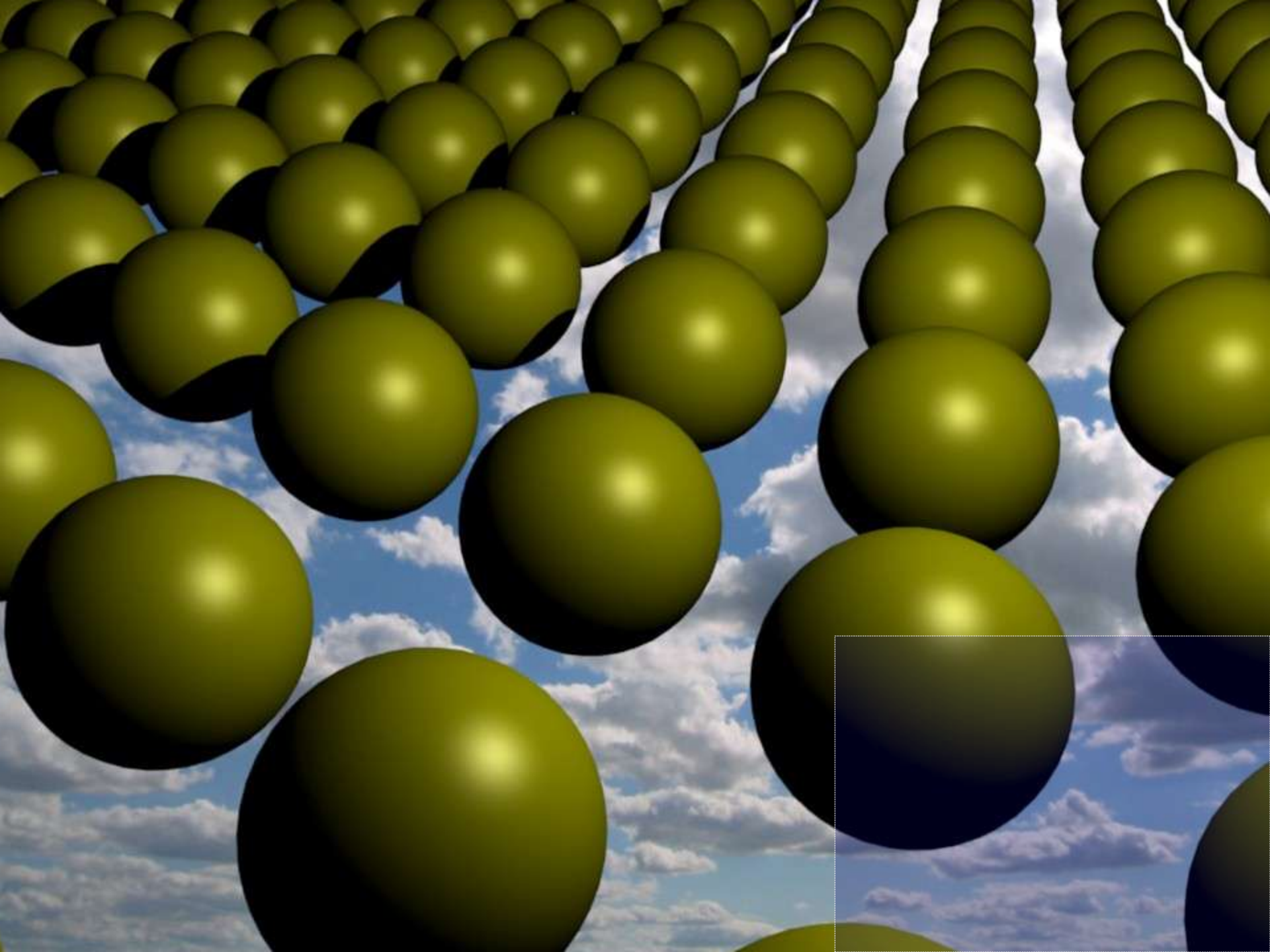
Bar-Ilan University
Dept. of Computer Science

Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures

Oded Regev
(Tel Aviv University and CNRS, ENS-Paris)



Based on work with Phong Q. Nguyen
(École normale supérieure)
[Eurocrypt'06]



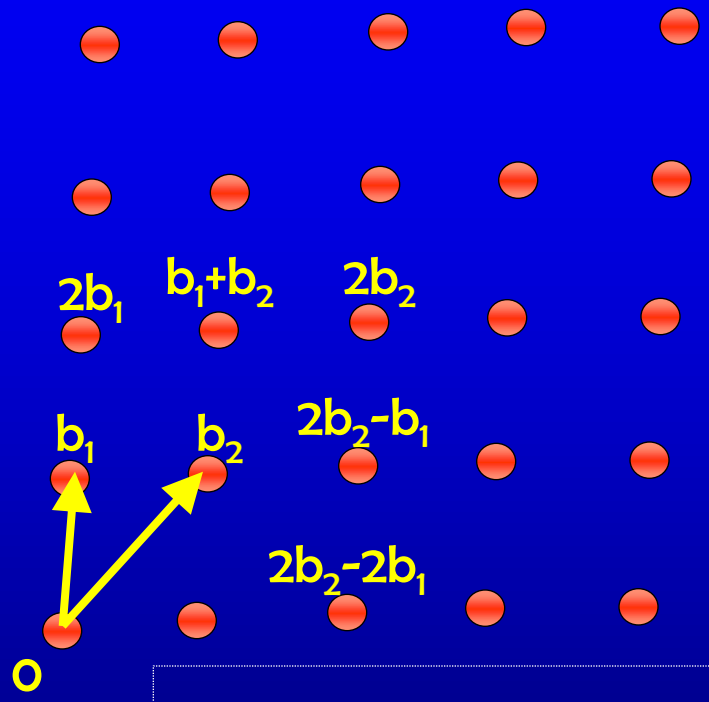
Lattices

Basis:

$b_1,\ldots,b_n$ vectors in R^n

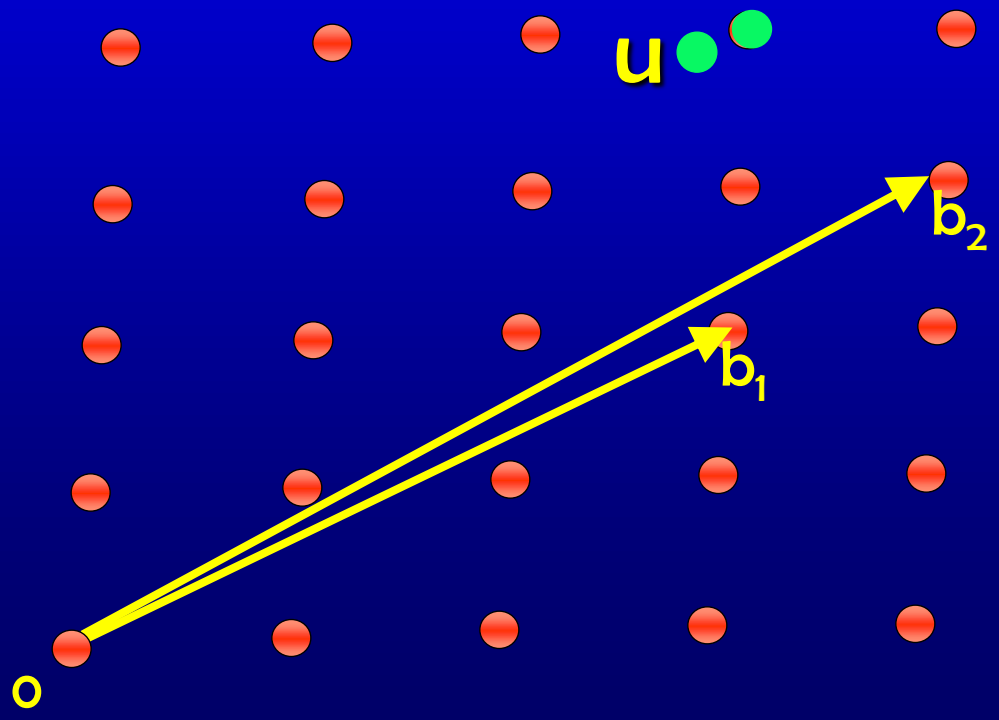
The lattice L is

$L=\{a_1b_1+\ldots+a_nb_n \mid a_i \text{ integers}\}$



Closest Vector Problem (CVP)

- CVP: Given a lattice and a target vector, find the closest lattice point
- Seems very difficult; best algorithms take time 2^n
- However, checking if a point is in a lattice is easy



Babai's (rounding) CVP Algorithm

- Babai's algorithm: given a point u, write

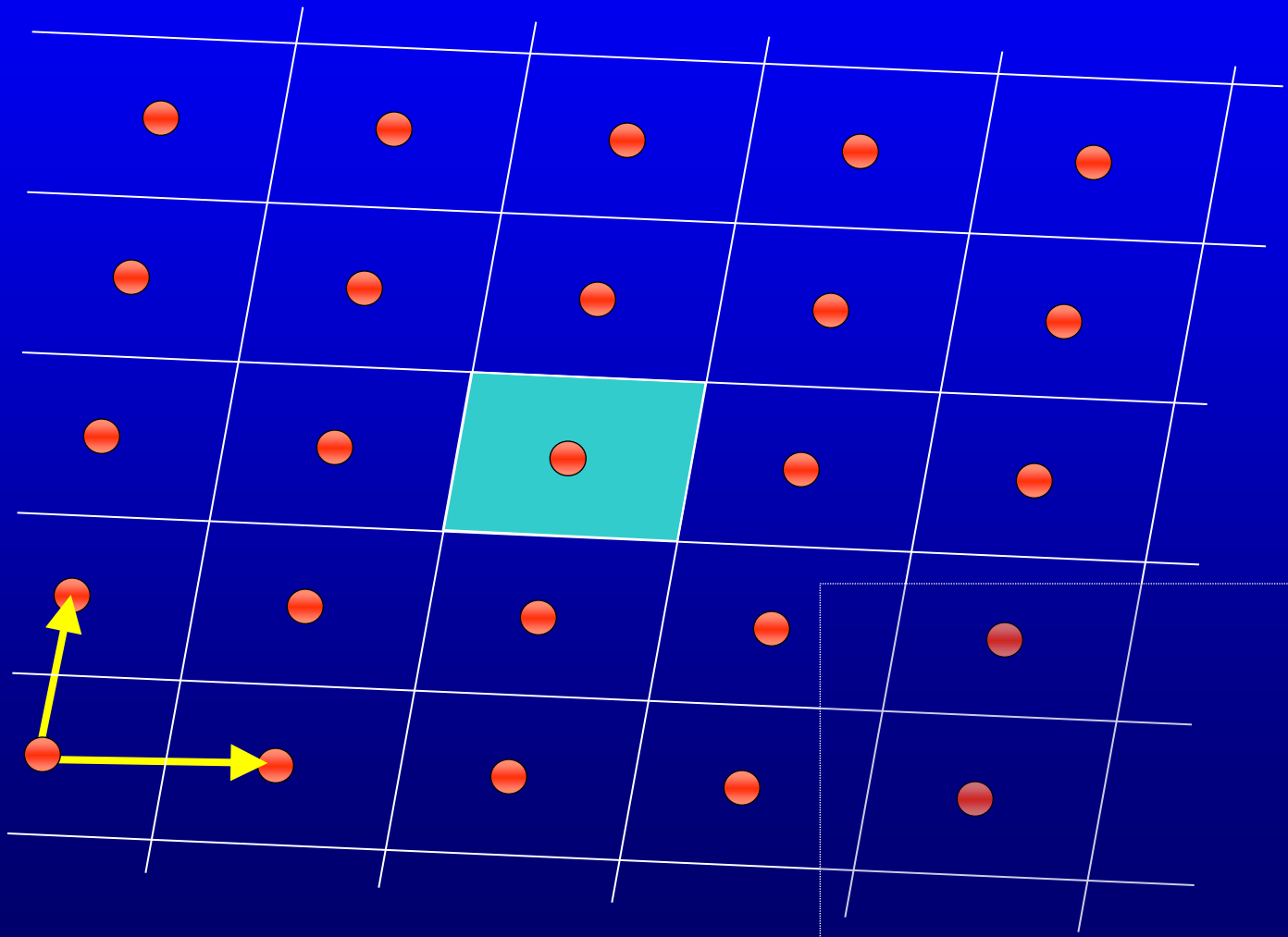
$$u = \alpha_1 b_1 + \cdots + \alpha_n b_n$$

and output

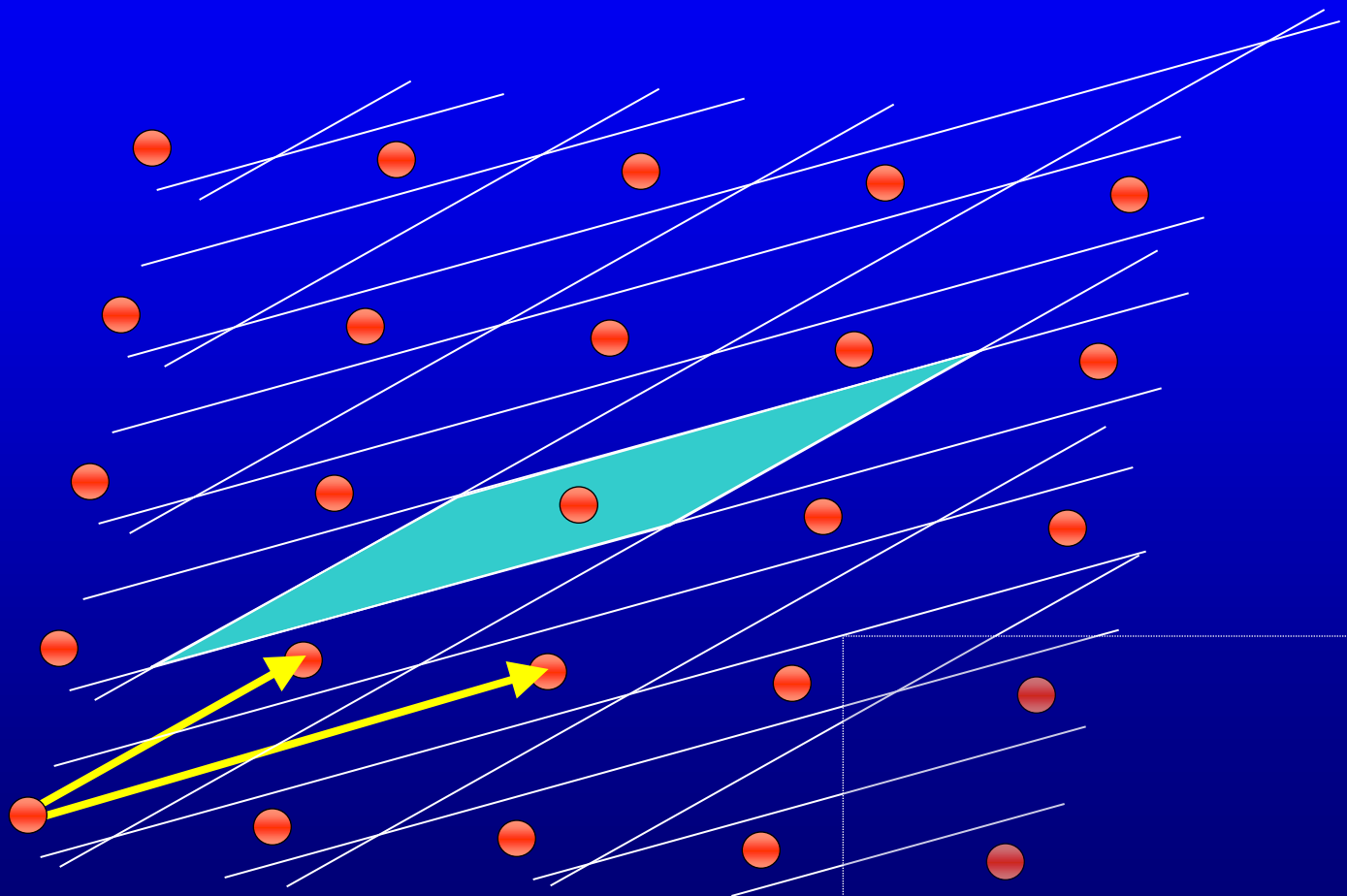
$$\lceil \alpha_1 \rfloor b_1 + \cdots + \lceil \alpha_n \rfloor b_n$$

- Works well for "good" bases

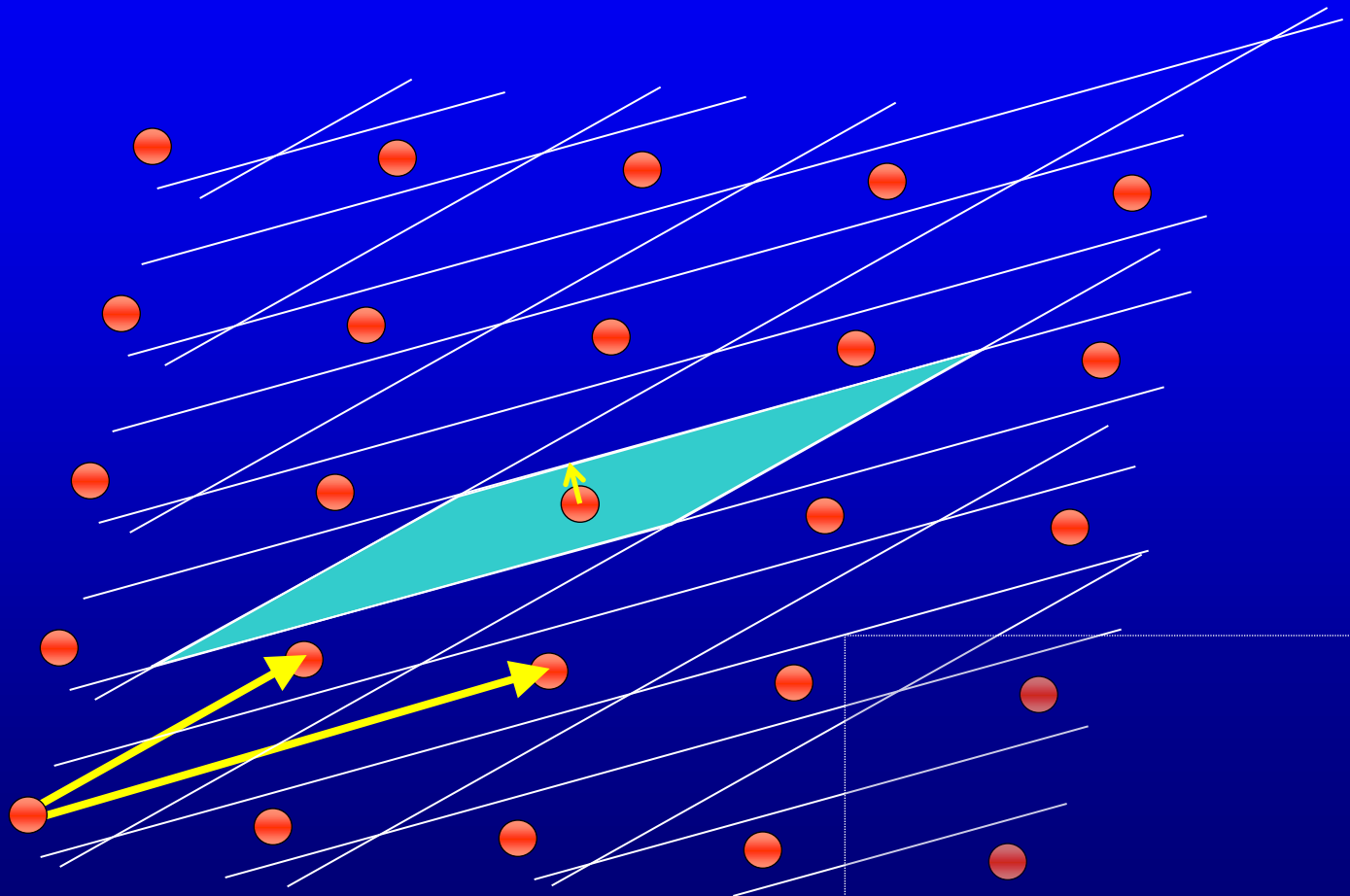
Babai's CVP Algorithm



Babai's CVP Algorithm



Babai's CVP Algorithm: Analysis



Babai's CVP Algorithm: Analysis

- For a basis $b_1,\ldots,b_n$, define the dual basis $b_1^*,\ldots,b_n^*$ by taking b_i^* to be the vector satisfying $\langle b_i^*, b_i\rangle = 1$ and $\langle b_i^*, b_j\rangle = 0$ for all $i \neq j$.
- In matrix notation, if $B=(b_1,\ldots,b_n)$, then $B^*=(B^{-1})^T$
- Notice that if $u = \alpha_1 b_1 + \cdots + \alpha_n b_n$ then $\alpha_i = \langle u, b_i^* \rangle$
- We can therefore equivalently write Babai's algorithm as:
  - Given a point u, output

$$\lceil \langle u, b_1^* \rangle \rfloor b_1 + \cdots + \lceil \langle u, b_n^* \rangle \rfloor b_n$$

- So the radius of correct decoding is:

$$\frac{1}{2 \max ||b_i^*||}$$

- The lattice generated by $b_1^*, \ldots, b_n^*$ is called the dual lattice

Signature Scheme

- Consists of:
  - Key generation algorithm: produces a (public-key,private-key) pair
  - Signing algorithm: given a message and a private-key, produces a signature
  - Verification algorithm: given a pair (message,signature) and a public key, verifies that the signature matches

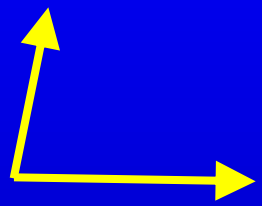
- Although can be built from any one-way function, efficient constructions are very important and still a main open question

The GGH Signature Scheme [1997]

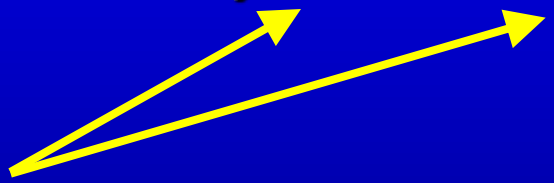
- Suggested in [GoldreichGoldwasserHalevi97]; no security proof
- Idea: CVP is hard, but easy with good basis
- The scheme:
  - Key generation algorithm: choose a lattice with some good basis
    - Private-key = good basis
    - Public-key = bad basis
  - Signing algorithm: given a message and a private key,
    - Map message to a point in space
    - Apply Babai's algorithm with good basis to obtain the signature
  - Verification algorithm: given message+signature and a public key, verify that
    - Signature is a lattice point, and
    - Signature is close to the message

GGH Signature Scheme:

Private-key:

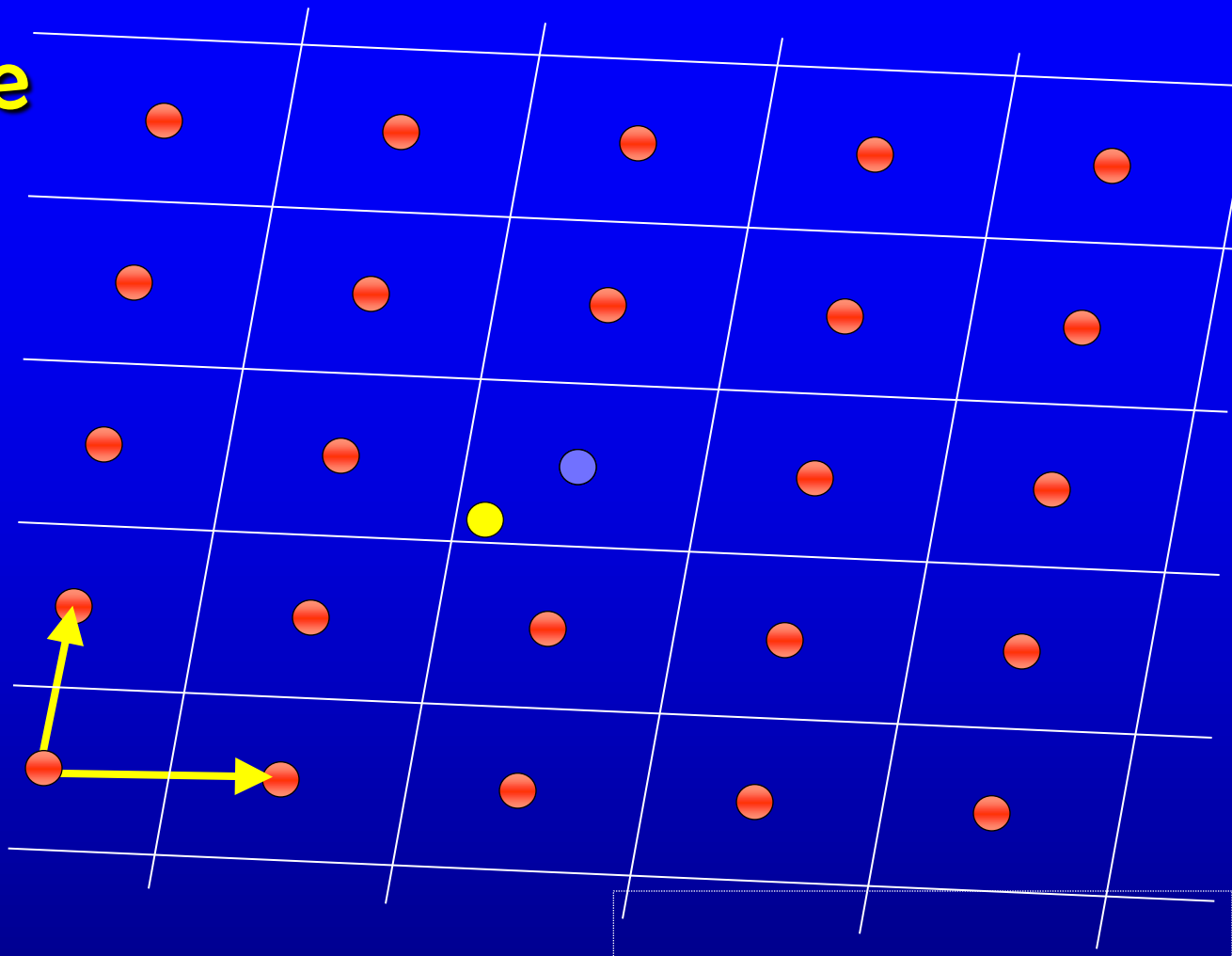


Public-key:

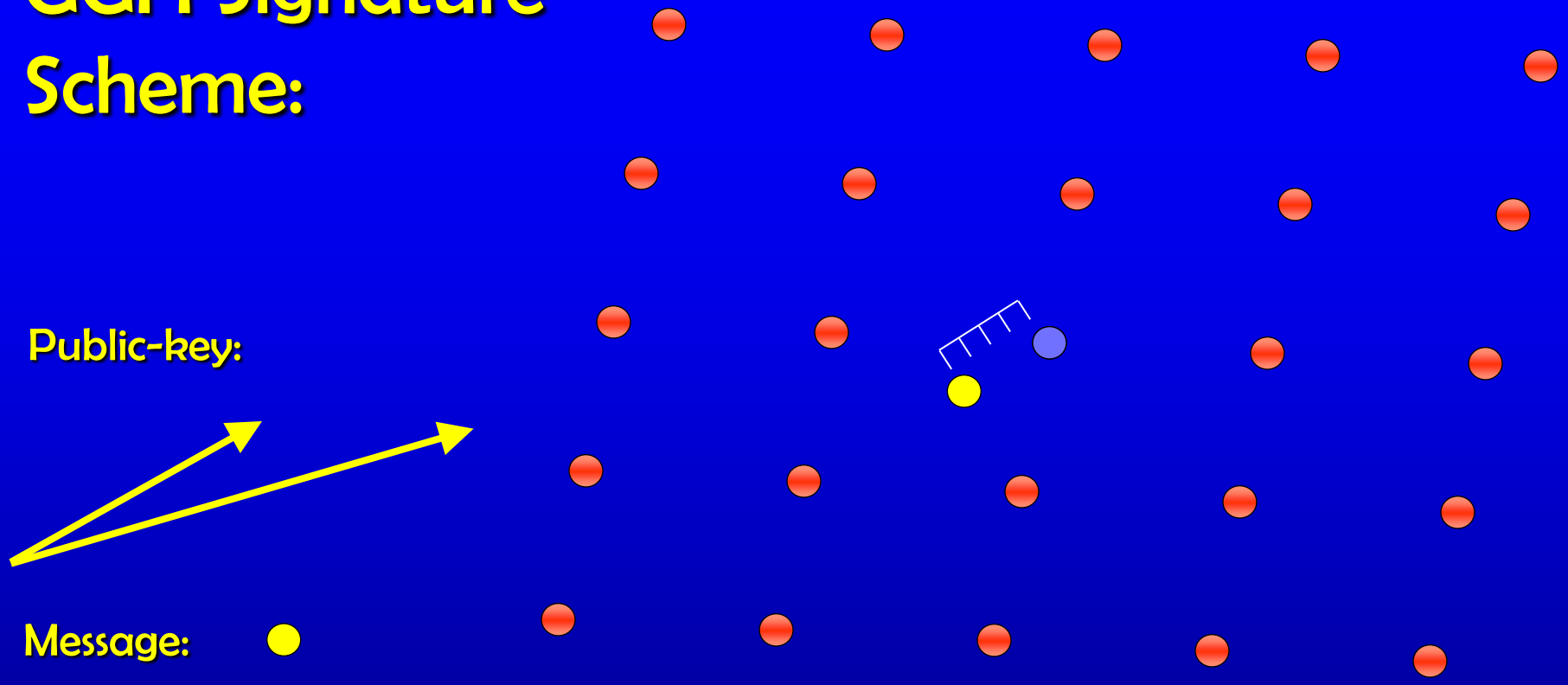


Message:

Signature:



GGH Signature Scheme:



Public-key:

Message:

Signature:

Verification: 1. ● should be a lattice point

2. distance between ● and ● should be small

NTRUSign

From Wikipedia, the free encyclopedia

NTRUSign, also known as the **NTRU Signature Algorithm**, is a public key cryptography digital signature algorithm based on the GGH signature scheme. It was first presented at the rump session of Asiacrypt 2001 and published in peer-reviewed form at the RSA Conference 2003. The 2003 publication included parameter recommendations for 80-bit security. A subsequent 2005 publication revised the parameter recommendations for 80-bit security, presented parameters that gave claimed security levels of 112, 128, 160, 192 and 256 bits, and described an algorithm to derive parameter sets at any desired security level. NTRU Cryptosystems, Inc. have applied for a patent on the algorithm.

NTRUSign involves mapping a message to a random point in $2N$-dimensional space, where N is one of the NTRUSign parameters, and solving the close vector problem in a lattice closely related to the NTRUEncrypt lattice. This lattice has the property that a private $2N$-dimensional basis for the lattice can be described with 2 vectors, each with N coefficients, and a public basis can be described with a single N-dimensional vector. This enables public keys to be represented in $O(N)$ space, rather than $O(N^2)$ as is the case with other lattice-based signature schemes. Operations take $O(N^2)$ time, as opposed to $O(N^3)$ for elliptic curve cryptography and RSA private key operations. NTRUSign is therefore claimed to be faster than those algorithms at low security levels, and considerably faster at high security levels.

NTRUSign is not a zero-knowledge signature scheme and a transcript of signatures leaks information about the private key, as first observed by Gentry and Szydlo. The current proposals use *perturbations* to increase the transcript length required to recover the private key: the effect of this is that the point representing the message is displaced by the signer by a small secret amount before the signature itself is calculated. The contribution of the perturbations to the transcript is designed to be difficult to distinguish from the contribution of the private key. NTRU claim that at least 2^{30} signatures are needed, and probably considerably more, before a transcript of perturbed signatures enables any useful attack.

NTRUSign is under consideration for standardization by the IEEE P1363 working group.

External links

[edit]

- NTRU Cryptosystems, Inc.
- NTRU Cryptosystems's technical website, containing specifications, tutorials and analysis of NTRUEncrypt.
- NTRUSign specification from CT-RSA 2003 (pdf)
- Extended version of CT-RSA paper (pdf)
- Most recent NTRUSign paper, including parameter sets for multiple security levels
- Paper by Gentry and Szydlo describing transcript attack on the original, unperturbed NTRUSign

Public-key cryptography edit

Algorithms: Cramer-Shoup | DH | DSA | ECDH | ECDSA | EKE | ElGamal | GMR | MQV | NTRUEncrypt | NTRUSign | Paillier | Rabin | Rabin-Williams | RSA | Schnorr |

WIKIPEDIA
The Free Encyclopedia

navigation
- Main Page
- Community Portal
- Featured articles
- Current events
- Recent changes
- Random article
- Help
- Contact Wikipedia
- Donations

search
Go | Search

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link
- Cite this article

The NTRUsign Signature Scheme

[HoffsteinHowgraveGrahamPipherSilvermanWhyte01]

- Essentially a very efficient implementation of the GGH signature scheme
  - Signature length only 1757 bits
  - Signing and verification are faster than RSA-based methods
- Based on the NTRU lattices (bicyclic lattices generated from a polynomial ring)
- Developed by the company NTRU and was under IEEE P1363.1
- Some flaws pointed out in [GentrySzydlo'02]

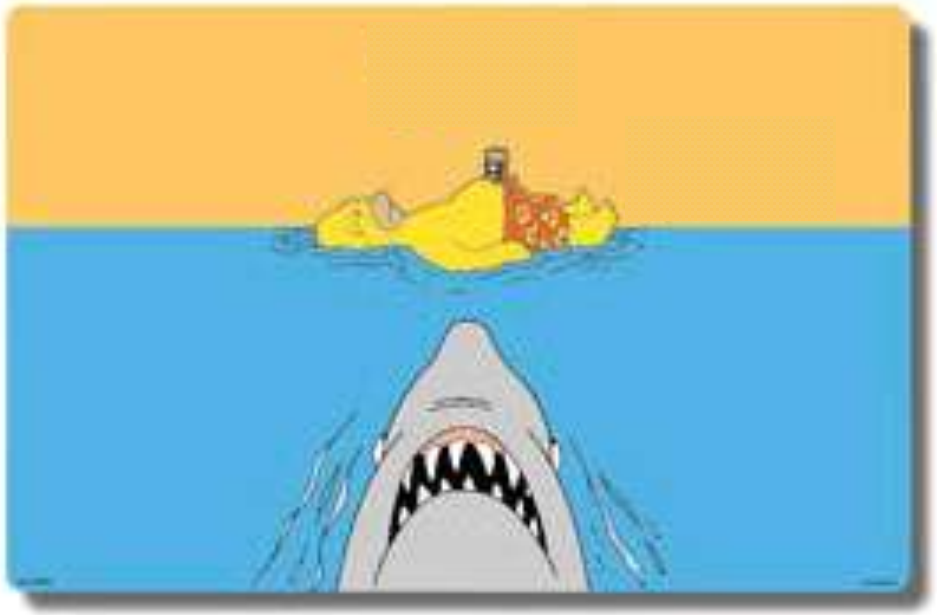
Main Result

- An inherent security flaw in GGH-based signature schemes
- Demonstrated a practical attack on:
  - GGH
    - Up to dimension 400
  - NTRUsign
    - Dimension 502
    - Applies to half of the parameter sets in IEEE P1363.1
    - Only 400 signatures needed!
- The attack recovers the private key
- Running time is a few minutes on a 2Ghz/2GB PC

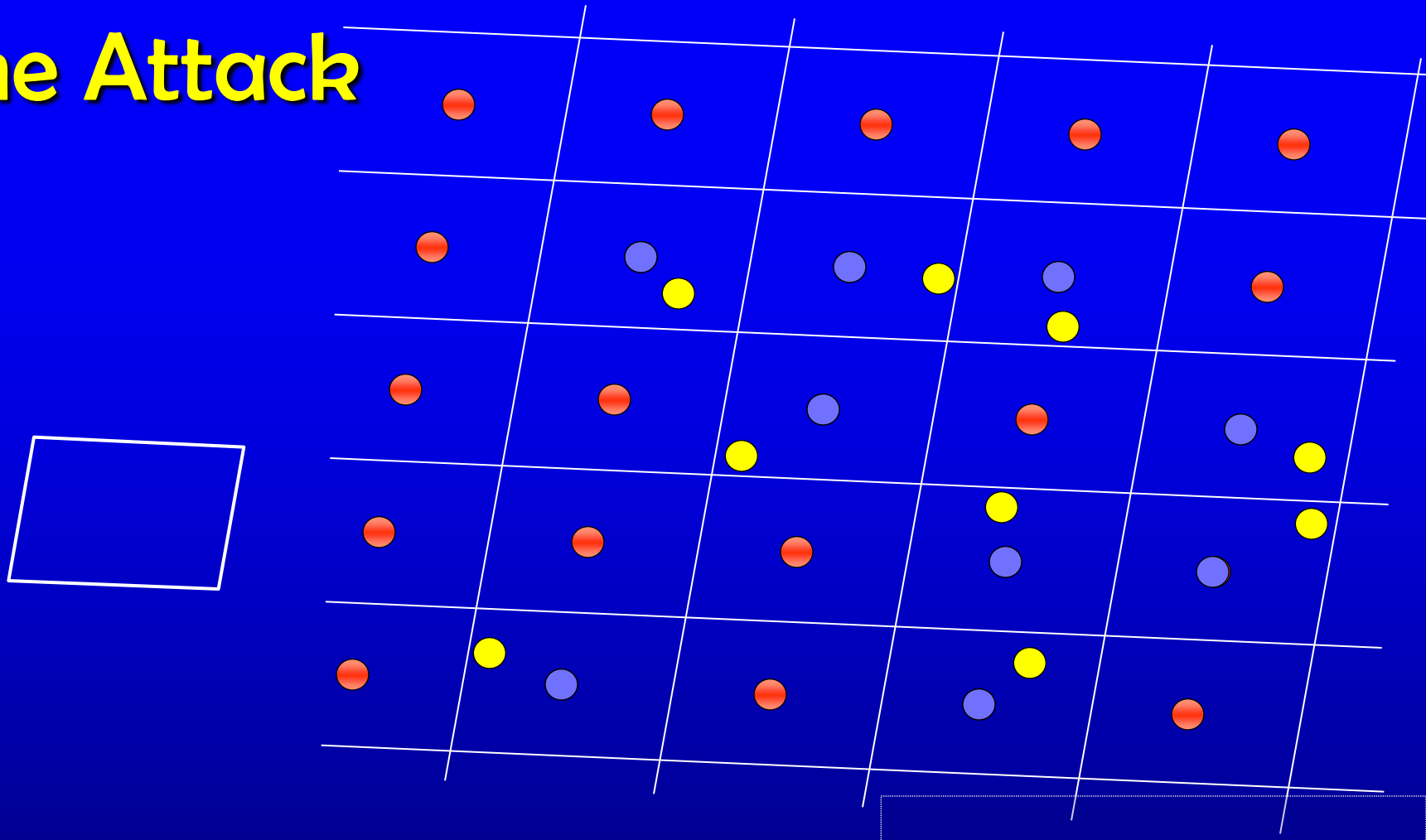
Main Result

- Possible countermeasures:
  - Pertubations, as suggested by NTRU in several of the IEEE P1363.1 parameter sets
  - Larger entries in private key
  - It is not clear if the attack can be extended to deal with these extensions
  - Use provably secure alternatives!!
- NTRUEncrypt is still secure, as is all provably secure lattice-based crypto!

THE ATTACK

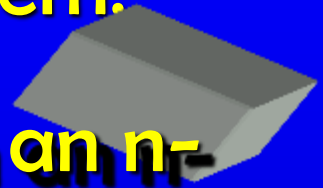


The Attack



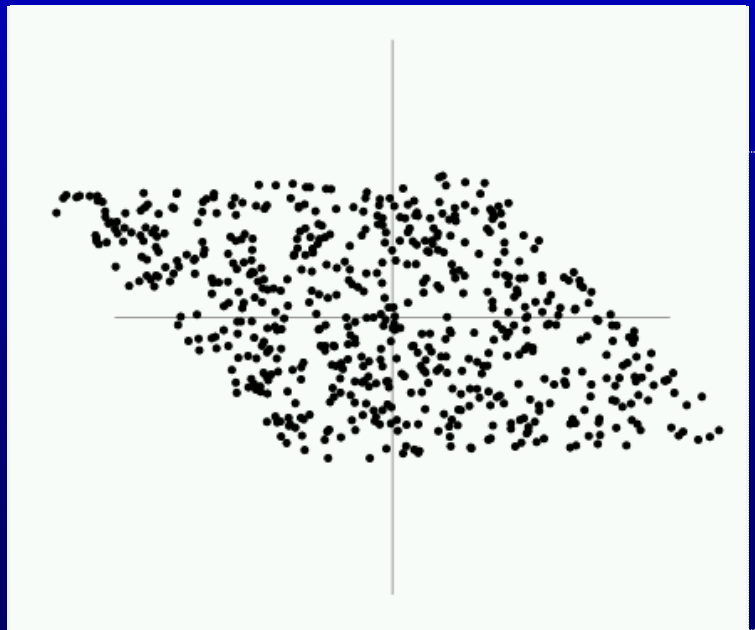
Hidden Parallelepiped Problem

- So it is enough to solve the following problem:

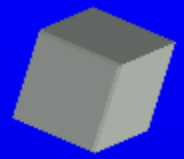


Given points sampled uniformly from an n-dimensional centered parallelepiped, recover the parallelepiped

- This would enable us to recover the private key



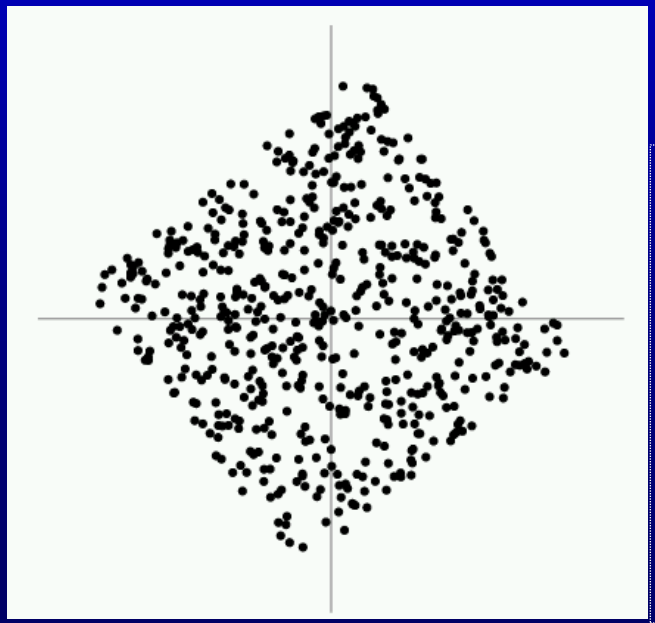
Hidden Hypercube Problem



- Let's try to solve an easier problem:

  Given points sampled uniformly from an n-dimensional centered unit hypercube, recover the hypercube

- We will later reduce the general case to the hypercube



HHP: First Attempt

- For a unit vector u define the variance in the direction u as

$$\mathrm{Var}(u) = \mathbf{E}_x[\langle u, x\rangle^2]$$

- Perhaps by computing Var(u) for many u's we can learn something

- The samples x can be written as $x = Uy$ for y chosen uniformly from $[-1,1]^n$ and an orthogonal matrix U, so

$$\begin{aligned}\mathrm{Var}(u) &= \mathbf{E}[\langle u, x\rangle^2] = \mathbf{E}[u^T x x^T u] \\ &= \mathbf{E}[u^T U y y^T U^T u] = u^T U \mathbf{E}[yy^T] U^T u \\ &= u^T U (I/3) U^T u = u^T u/3 = 1/3.\end{aligned}$$

HHP: Second Attempt

- So let's try the fourth moment instead:

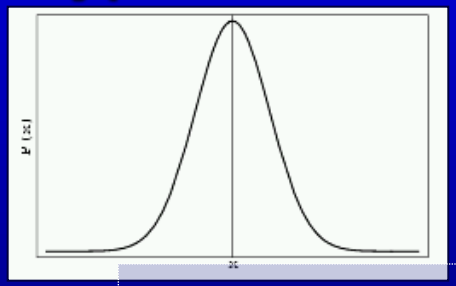
$$\mathrm{Kur}(u) = \mathbf{E}_x[\langle u, x\rangle^4]$$

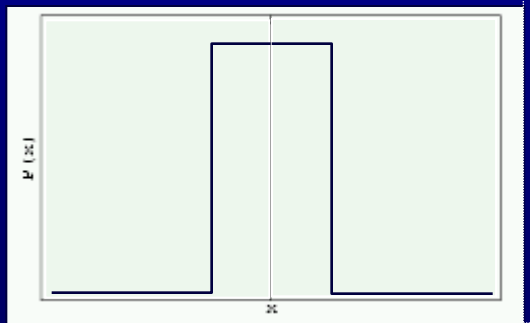
- A short calculation shows that

$$\mathrm{Kur}(u) = \frac{1}{3} - \frac{2}{15}\sum_{i=1}^{n} u_i^4$$

where u_i are u's coordinates in the hypercube basis

- Therefore:
  - In direction of the corners the kurtosis is ~1/3
  - In direction of the faces the kurtosis is 1/5

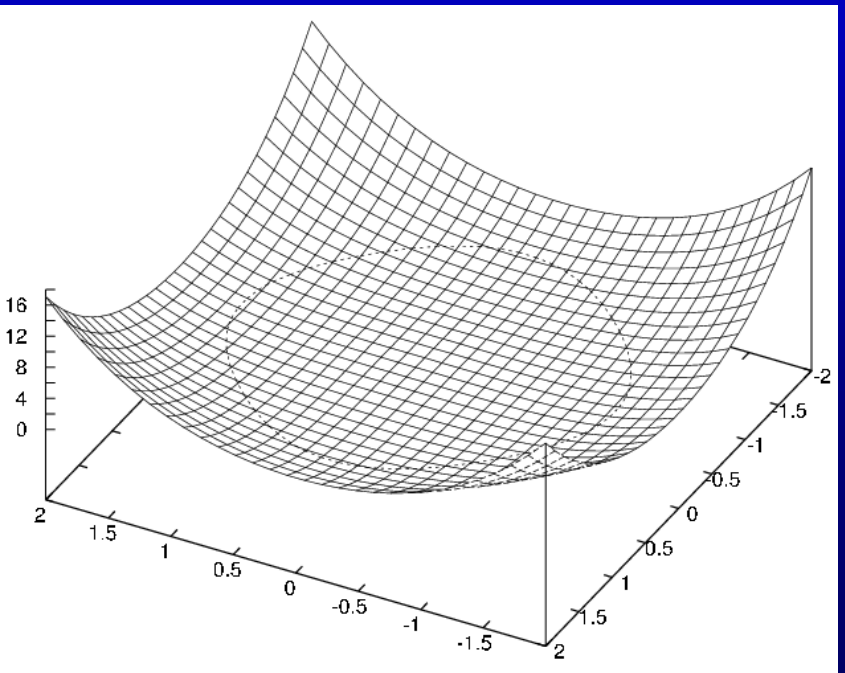




HHP: The Algorithm

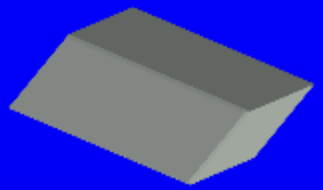
The algorithm repeats the following steps:

- Choose a random unit vector *u*
- Perform a gradient descent on the sphere to find a local minimum of Kur(u)
- Output the resulting vector



Each application randomly yields one of the 2n face vectors

Back to HPP



- Now the samples can be written as $x = Ry$ where y is chosen uniformly from $[-1,1]^n$ and R is some matrix
- Consider the average of the matrix xx^T

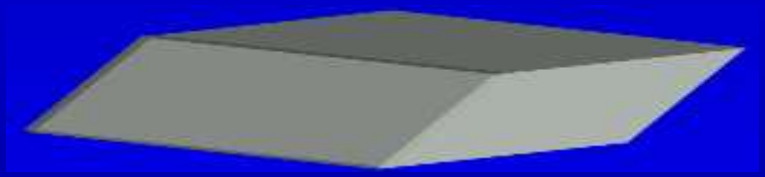
$$\mathbf{E}[xx^T] = \mathbf{E}[Ryy^T R^T]$$
$$= R\mathbf{E}[yy^T]R^T = RR^T/3.$$

- Hence, we can get an approximation of $S=RR^T$ (the Gram matrix of R)
- Now the matrix $S^{-1/2}R$ is orthogonal:

$$R^T S^{-1/2} S^{-1/2} R = I$$

Back to HPP

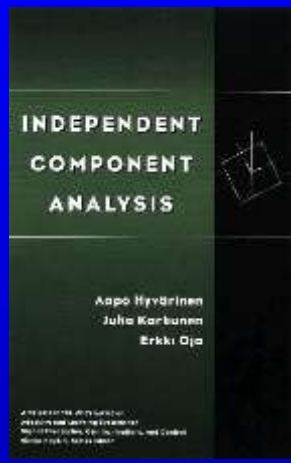
- Hence, by applying the transformation $S^{-1/2}$ to our samples x, we obtain samples from a unit hypercube, so we're back to HCP
- In other words, we have morphed a parallelepiped into a hypercube:

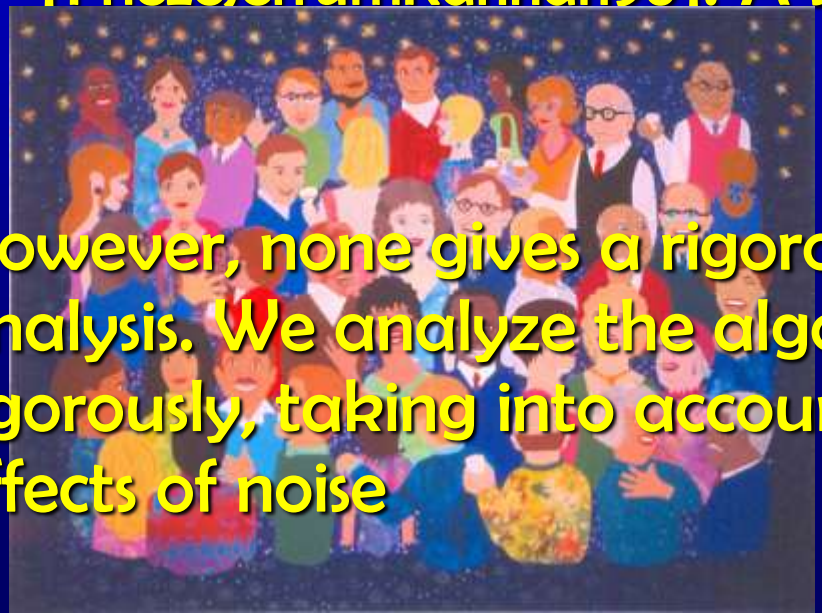


- Now run the HHP algorithm on the samples $S^{-1/2}x$. If U is the returned matrix, return $S^{1/2}U$ as the parallelepiped.

We're not alone

- The HPP has already been looked at:
  - In statistical analysis, and in particular Independent Component Analysis (ICA). The FastICA algorithm is very similar to ours [HyvärinenOja97]. Many applications in signal processing, neural networks, etc.
  - In the computational learning community, by [FriezeJerrumKannan96]. A somewhat different

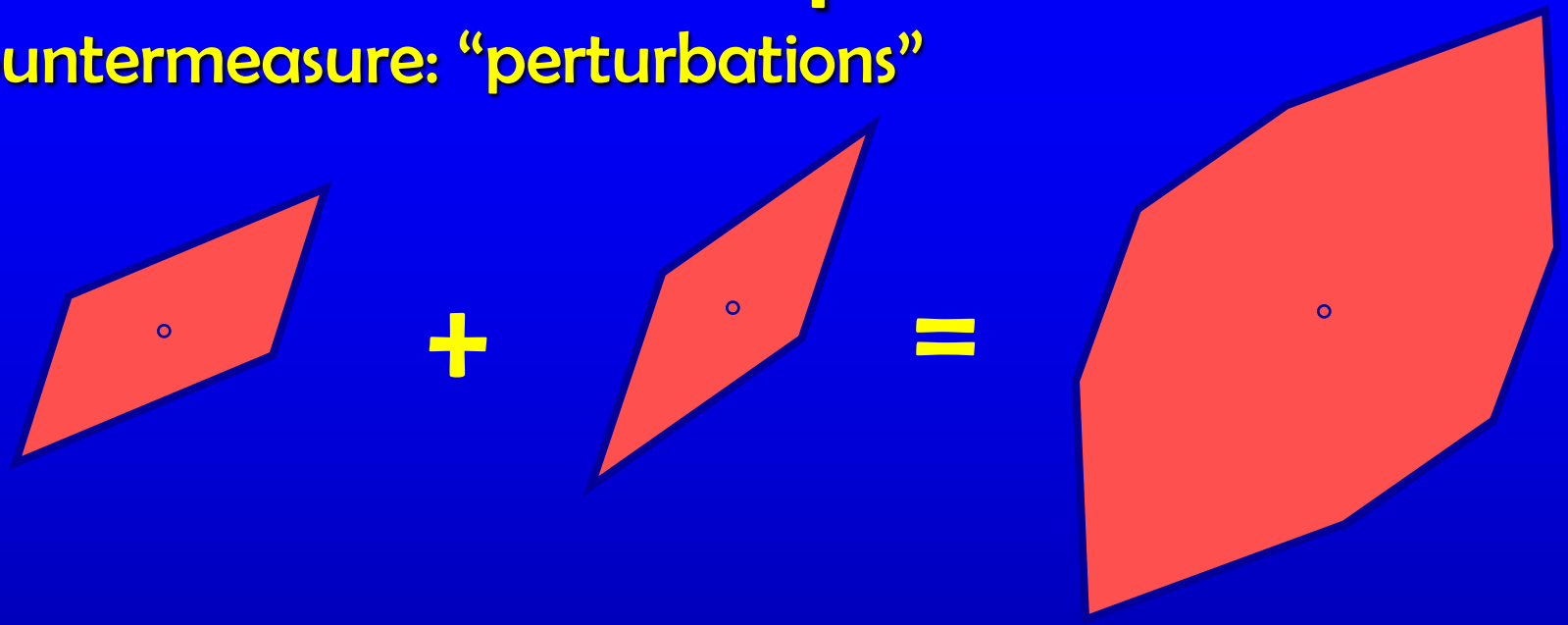




- However, none gives a rigorous analysis. We analyze the algorithm rigorously, taking into account the effects of noise

Followup Work

- Countermeasure: "perturbations"

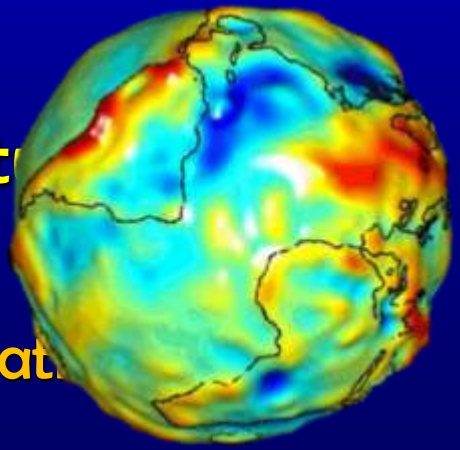


- Can the attack be extended to deal with pertubrations?
  - Yes, to some extent!

  [DucasNguyen12]
- Provably secure signat

  Gaussian sampler

  [GentryPeikertVaikuntanat



Thanks