Sealing the Leak on Classical NTRU Signatures

C. Aguilar Melchor[1] X. Boyen[2]
J.C. Deneuville [1] P. Gaborit[1]

[1]XLIM-DMI, Université de Limoges, France

[2]Queensland University of Technology, Brisbane, Australia

POST-QUANTUM CRYPTOGRAPHY 2014
October the 1st

Outline

Outline

Definitions

Lattice

A m-dimensional lattice is a discrete subgroup of $\mathbb{R}^m$. Formally, if $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the lattice $\Lambda(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is the set

$$\Lambda = \{\sum_{i=1}^{n} x_i \mathbf{b}_i ; x_i \in \mathbb{Z}\} \subset \mathbb{R}^m$$

Vocabulary

- rank n (main security parameter)
- dimension m ($m = \mathcal{O}(n \cdot \log n)$)
- basis $\mathbf{B} = (\mathbf{b}_1 | \cdots | \mathbf{b}_n)$ (multiple basis)

Definitions

Lattice

A m-dimensional lattice is a discrete subgroup of $\mathbb{R}^m$. Formally, if $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the lattice $\Lambda(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is the set

$$\Lambda = \{\sum_{i=1}^{n} x_i \mathbf{b}_i ; x_i \in \mathbb{Z}\} \subset \mathbb{R}^m$$

Vocabulary

- rank n (main security parameter)
- dimension m ($m = \mathcal{O}(n \cdot \log n)$)
- basis $\mathbf{B} = (\mathbf{b}_1 | \cdots | \mathbf{b}_n)$ (multiple basis)

Definitions

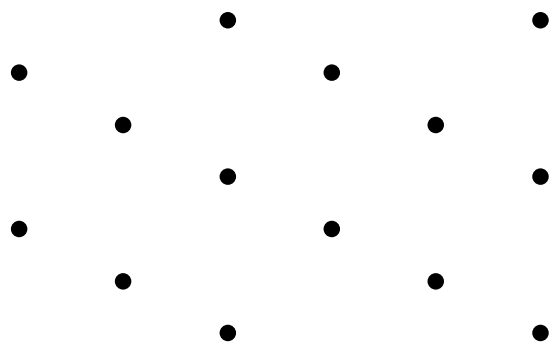
Lattice

A m-dimensional lattice is a discrete subgroup of $\mathbb{R}^m$. Formally, if $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the lattice $\Lambda(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is the set

$$\Lambda = \{\sum_{i=1}^{n} x_i \mathbf{b}_i ; x_i \in \mathbb{Z}\} \subset \mathbb{R}^m$$

Vocabulary

- rank n (main security parameter)
- dimension m ($m = \mathcal{O}(n \cdot \log n)$)
- basis $\mathbf{B} = (\mathbf{b}_1 | \cdots | \mathbf{b}_n)$ (multiple basis)



Definitions

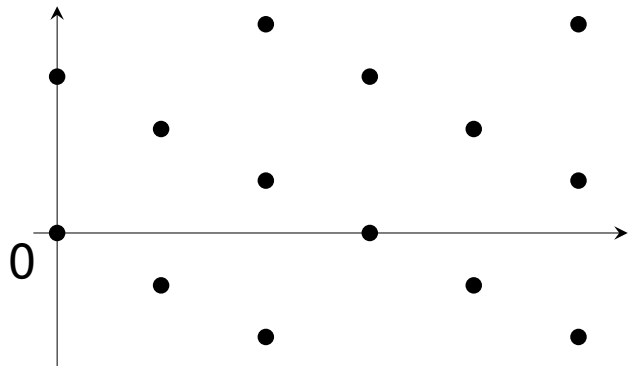
Lattice

A m-dimensional lattice is a discrete subgroup of $\mathbb{R}^m$. Formally, if $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the lattice $\Lambda(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is the set

$$\Lambda = \{\sum_{i=1}^{n} x_i \mathbf{b}_i ; x_i \in \mathbb{Z}\} \subset \mathbb{R}^m$$

Vocabulary

- rank n (main security parameter)
- dimension m ($m = \mathcal{O}(n \cdot \log n)$)
- basis $\mathbf{B} = (\mathbf{b}_1 | \cdots | \mathbf{b}_n)$ (multiple basis)



Definitions

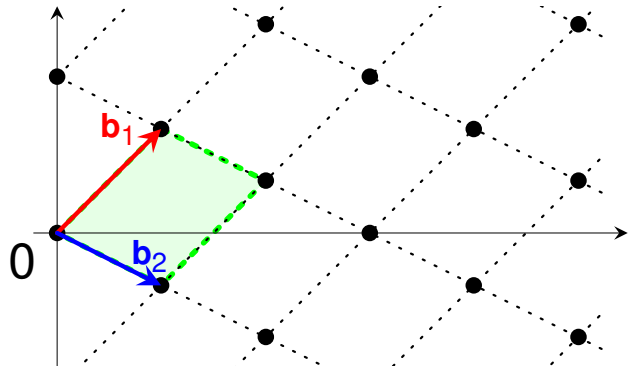
Lattice

A m-dimensional lattice is a discrete subgroup of $\mathbb{R}^m$. Formally, if $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the lattice $\Lambda(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is the set

$$\Lambda = \{\sum_{i=1}^{n} x_i \mathbf{b}_i ; x_i \in \mathbb{Z}\} \subset \mathbb{R}^m$$

Vocabulary

- rank n (main security parameter)
- dimension m ($m = \mathcal{O}(n \cdot \log n)$)
- basis $\mathbf{B} = (\mathbf{b}_1 | \cdots | \mathbf{b}_n)$ (multiple basis)



Definitions

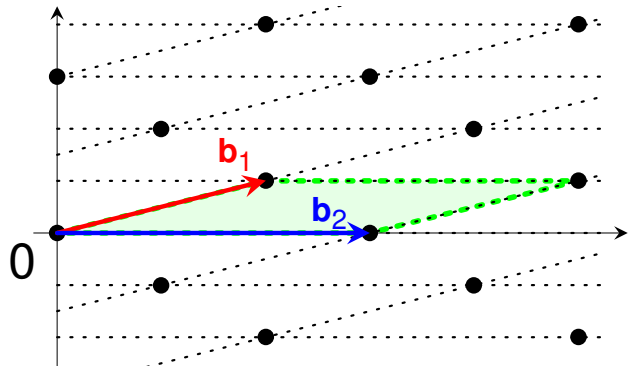
Lattice

A m-dimensional lattice is a discrete subgroup of $\mathbb{R}^m$. Formally, if $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the lattice $\Lambda(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is the set

$$\Lambda = \{\sum_{i=1}^{n} x_i \mathbf{b}_i ; x_i \in \mathbb{Z}\} \subset \mathbb{R}^m$$

Vocabulary

- rank n (main security parameter)
- dimension m ($m = \mathcal{O}(n \cdot \log n)$)
- basis $\mathbf{B} = (\mathbf{b}_1 | \cdots | \mathbf{b}_n)$ (multiple basis)



Definitions

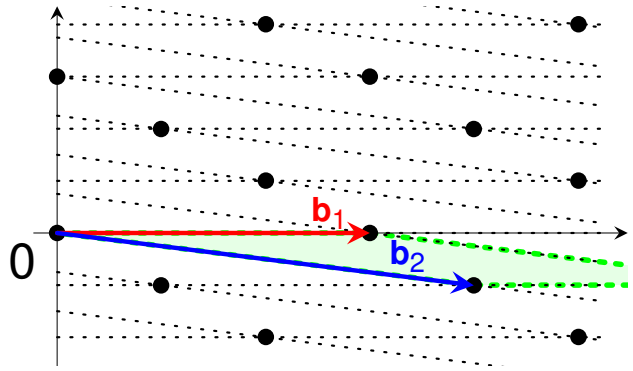
Lattice

A m-dimensional lattice is a discrete subgroup of $\mathbb{R}^m$. Formally, if $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the lattice $\Lambda(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is the set

$$\Lambda = \{\sum_{i=1}^{n} x_i \mathbf{b}_i ; x_i \in \mathbb{Z}\} \subset \mathbb{R}^m$$

Vocabulary

- rank n (main security parameter)
- dimension m ($m = \mathcal{O}(n \cdot \log n)$)
- basis $\mathbf{B} = (\mathbf{b}_1 | \cdots | \mathbf{b}_n)$ (multiple basis)



Definitions

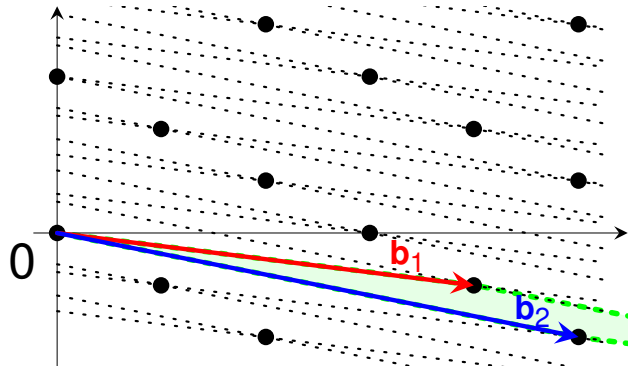
Lattice

A m-dimensional lattice is a discrete subgroup of $\mathbb{R}^m$. Formally, if $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the lattice $\Lambda(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is the set

$$\Lambda = \{\sum_{i=1}^{n} x_i \mathbf{b}_i ; x_i \in \mathbb{Z}\} \subset \mathbb{R}^m$$

Vocabulary

- rank n (main security parameter)
- dimension m ($m = \mathcal{O}(n \cdot \log n)$)
- basis $\mathbf{B} = (\mathbf{b}_1 | \cdots | \mathbf{b}_n)$ (multiple basis)



Matrix Representation and q-ary Lattices

Matrix Representation

Given $\mathbf{B} = (\mathbf{b}_1 | \cdots | \mathbf{b}_n) \in \mathbb{Z}^{m \times n}$, the lattice generated by $\mathbf{B}$ is

$$\Lambda(\mathbf{B}) = \{\mathbf{B} \cdot \mathbf{x}; \mathbf{x} \in \mathbb{Z}^n\}$$

q-ary Lattices

Let $\mathbf{B} = (\mathbf{b}_1 | \cdots | \mathbf{b}_n) \in \mathbb{Z}_q^{m \times n}$ for some prime q, and let

$$\Lambda_q(\mathbf{B}) = \{\mathbf{B} \cdot \mathbf{x} \mod q : \mathbf{x} \in \mathbb{Z}^n\}, \text{ and}$$

$$\Lambda_q^{\perp}(\mathbf{B}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y}^t \mathbf{B} = \mathbf{0} \mod q\}.$$

Matrix Representation and q-ary Lattices

Matrix Representation

Given $\mathbf{B} = (\mathbf{b}_1 | \cdots | \mathbf{b}_n) \in \mathbb{Z}^{m \times n}$, the lattice generated by $\mathbf{B}$ is

$$\Lambda(\mathbf{B}) = \{\mathbf{B} \cdot \mathbf{x}; \mathbf{x} \in \mathbb{Z}^n\}$$

q-ary Lattices

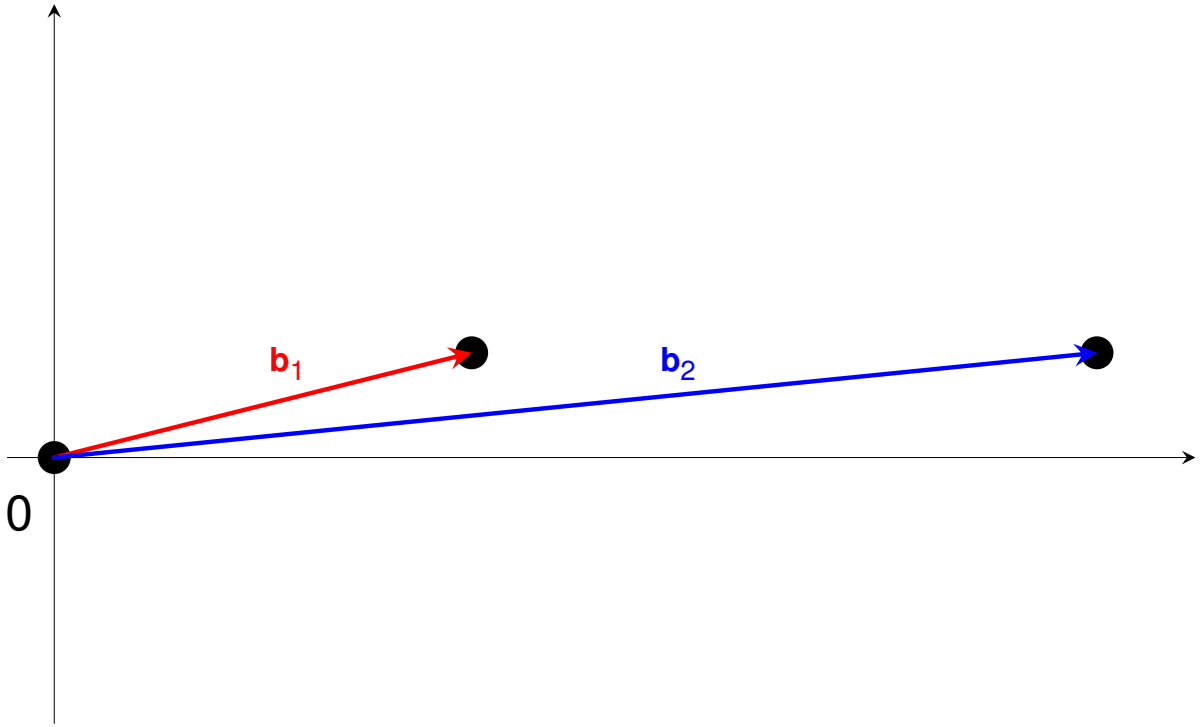
Let $\mathbf{B} = (\mathbf{b}_1 | \cdots | \mathbf{b}_n) \in \mathbb{Z}_q^{m \times n}$ for some prime q, and let

$$\Lambda_q(\mathbf{B}) = \{\mathbf{B} \cdot \mathbf{x} \mod q : \mathbf{x} \in \mathbb{Z}^n\}, \text{ and}$$

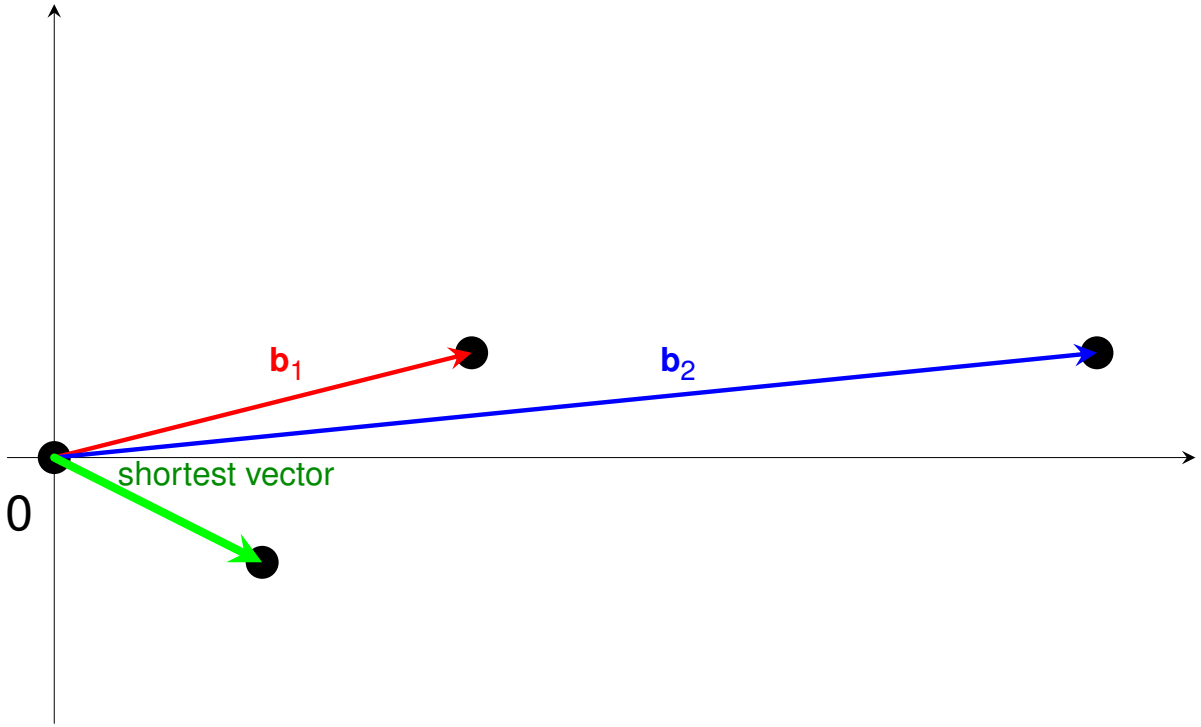
$$\Lambda_q^{\perp}(\mathbf{B}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y}^t \mathbf{B} = \mathbf{0} \mod q\}.$$

Outline

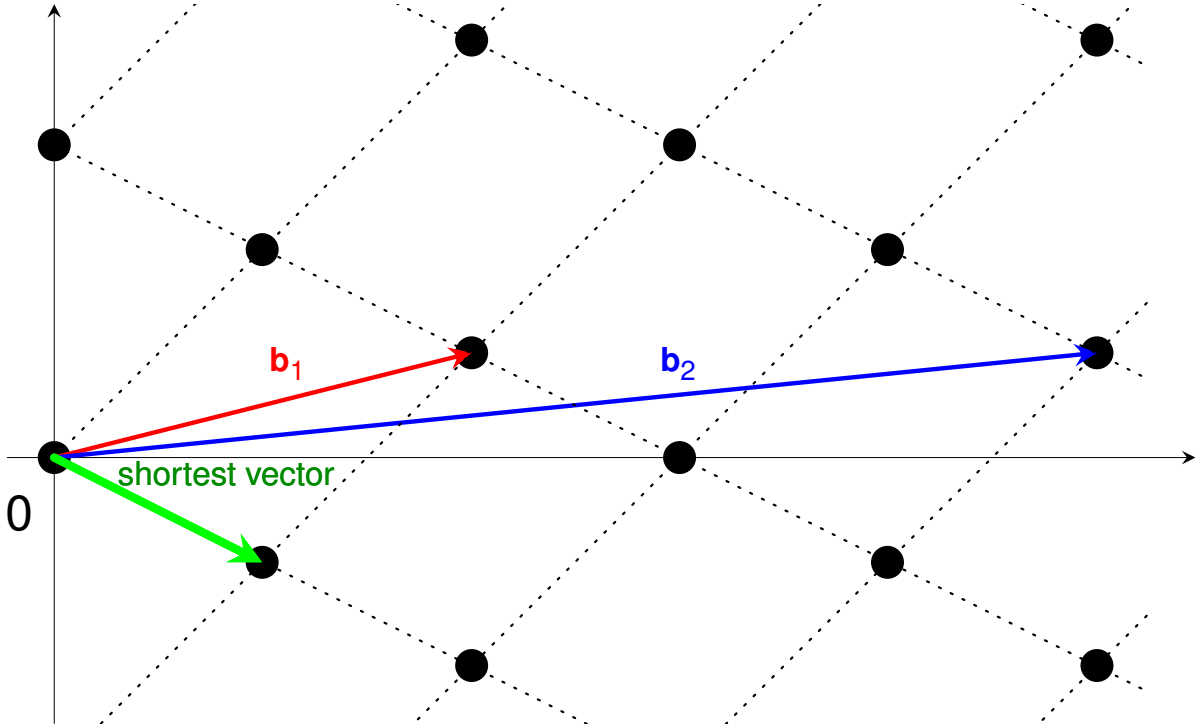
Shortest Vector Problem



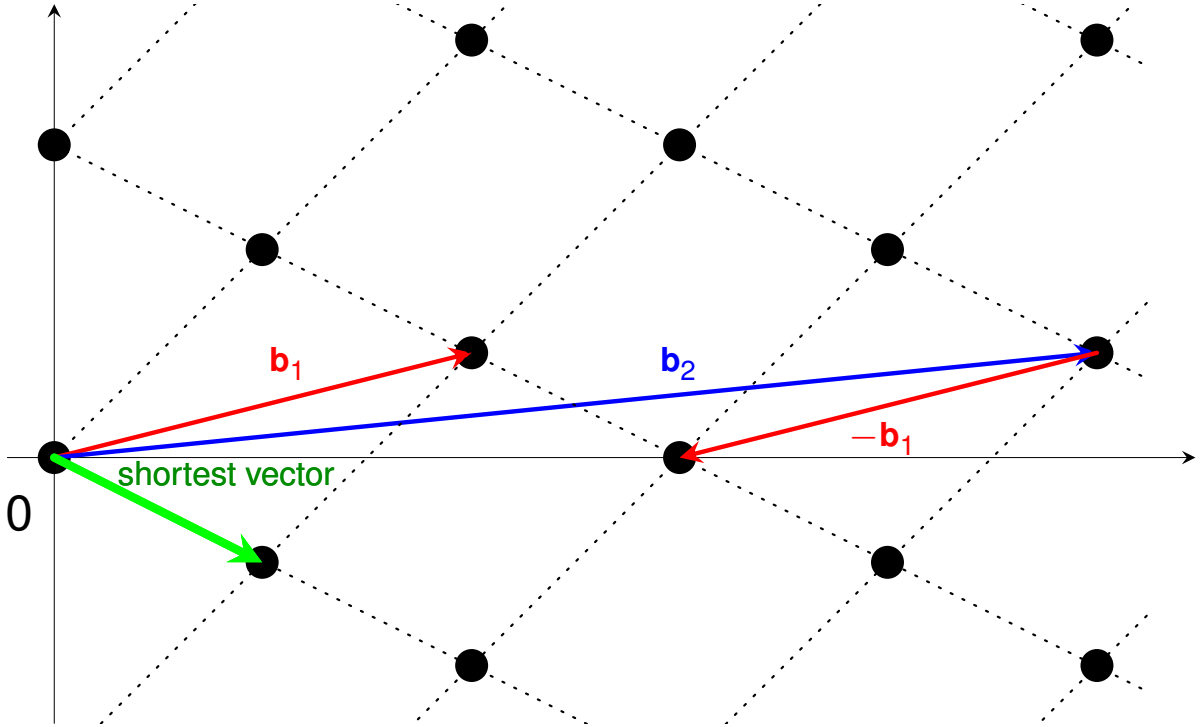
Shortest Vector Problem



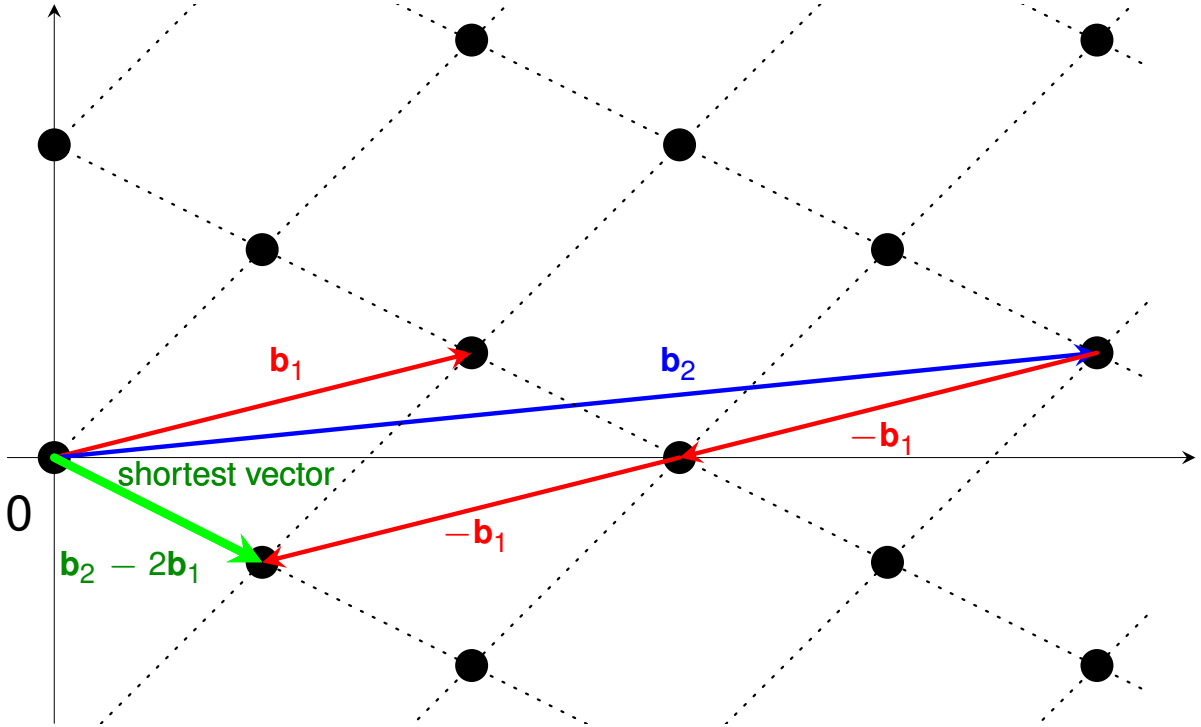
Shortest Vector Problem



Shortest Vector Problem

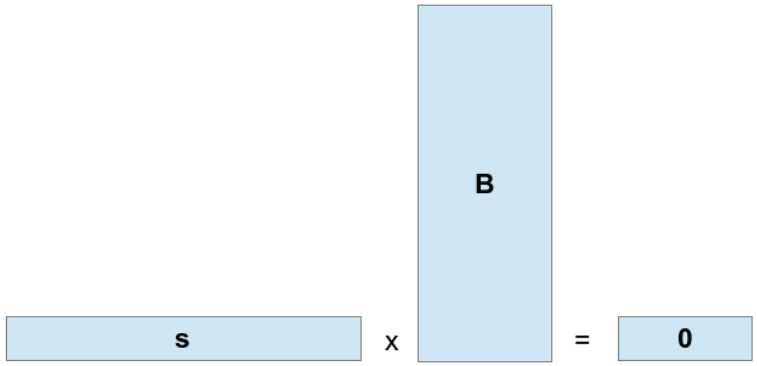


Shortest Vector Problem



Small Integer Solution

Given $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, find "small" $\mathbf{s} \in \mathbb{Z}^m$ such that $\mathbf{s}^t\mathbf{B} = \mathbf{0} \mod q$

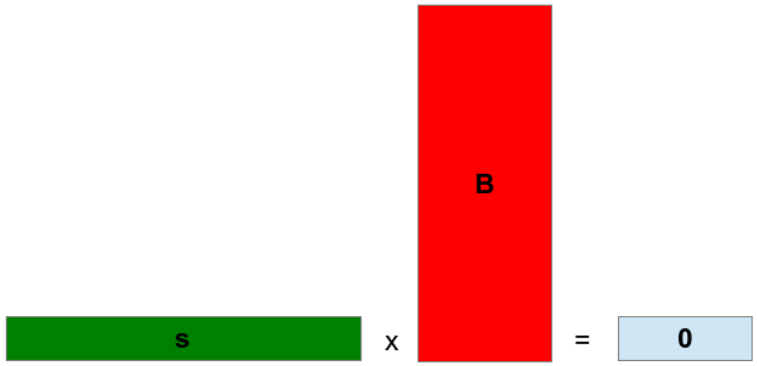


Relationship to Lattices

Solving **SIS** in random lattices **B** is "close" to solving **SVP** in $\Lambda_q^{\perp}(\mathbf{B})$

Small Integer Solution

Given $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, find "small" $\mathbf{s} \in \mathbb{Z}^m$ such that $\mathbf{s}^t\mathbf{B} = \mathbf{0} \mod q$

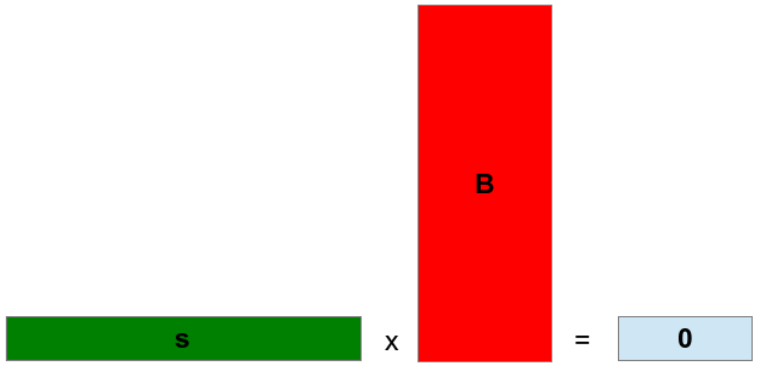


Relationship to Lattices

Solving **SIS** in random lattices **B** is "close" to solving **SVP** in $\Lambda_q^{\perp}(\mathbf{B})$

Small Integer Solution

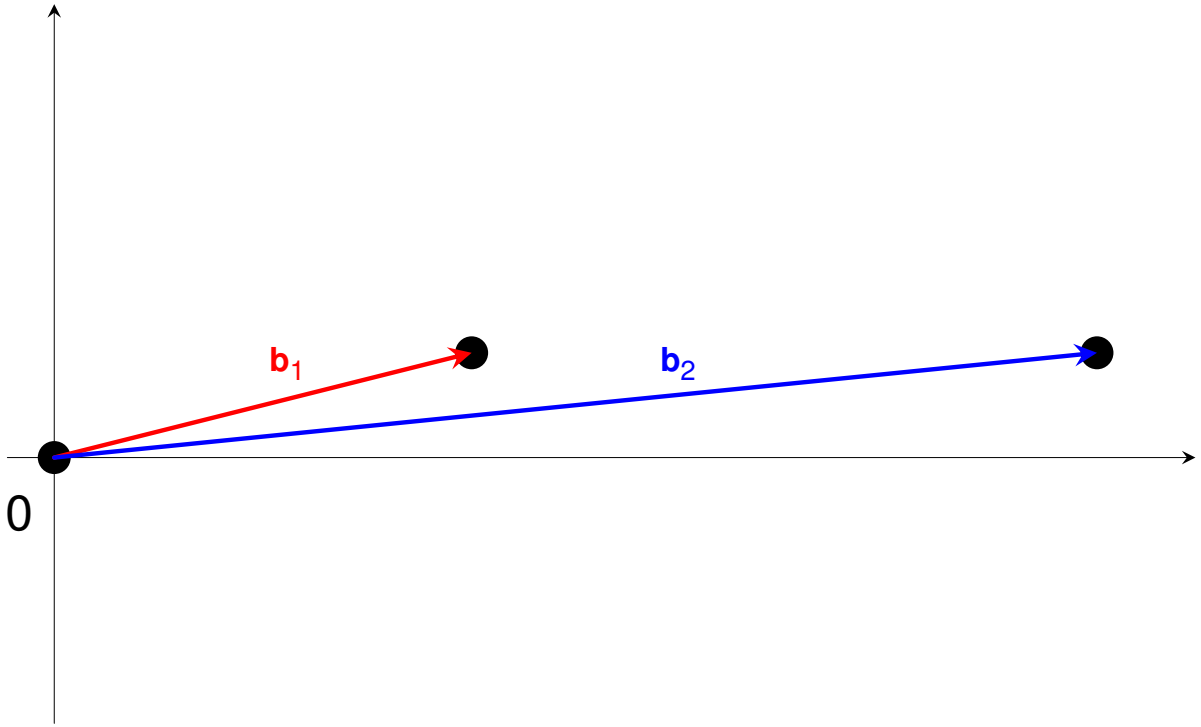
Given $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, find "small" $\mathbf{s} \in \mathbb{Z}^m$ such that $\mathbf{s}^t \mathbf{B} = \mathbf{0} \mod q$



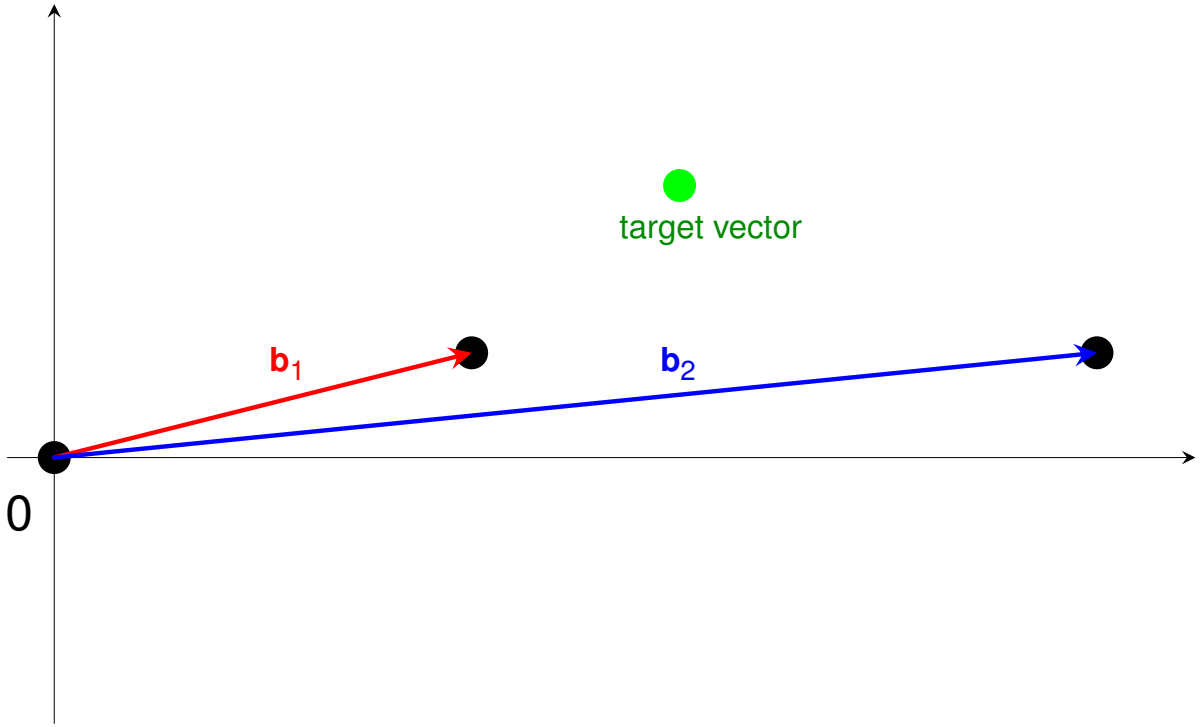
Relationship to Lattices

Solving **SIS** in random lattices **B** is "close" to solving **SVP** in $\Lambda_q^{\perp}(\mathbf{B})$

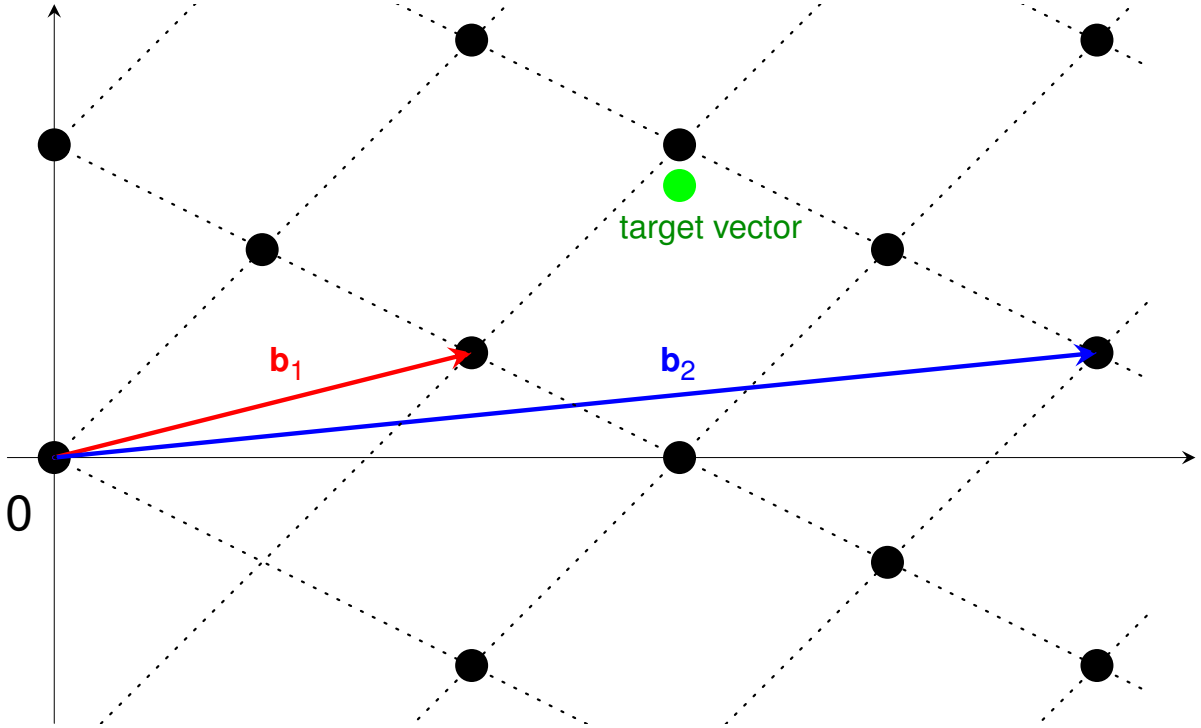
Closest Vector Problem



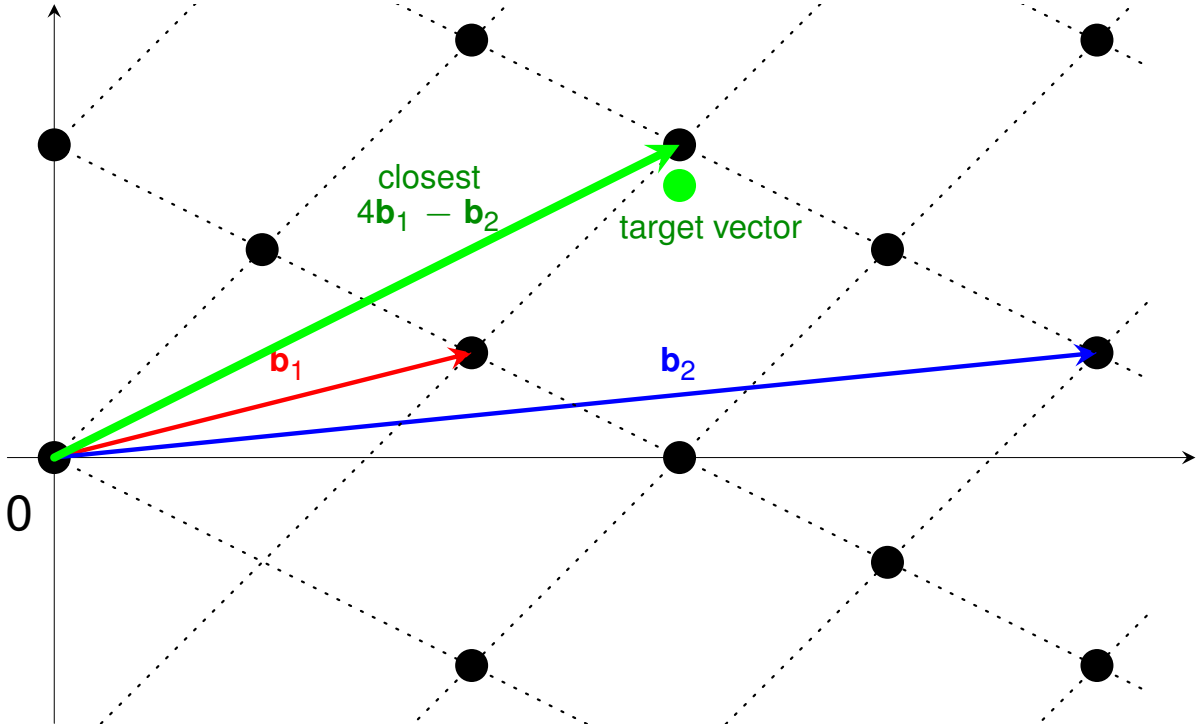
Closest Vector Problem



Closest Vector Problem



Closest Vector Problem



Security Level

- $\delta = \left(\frac{\lambda_1}{\det(\Lambda)^{1/n}}\right)^{1/n}$ [CN11]
- "Exact" bitlevel correpsondance [LP11]
- Depends on the algorithm

k	δ
80	1.00783
100	1.00696
128	1.00602

BKZ 2.0: Better Lattice Security Estimates

Yuanmi Chen and Phong Q. Nguyen

[1] ENS, Dept. Informatique, 45 rue d'Ulm, 75005 Paris, France.
http://www.eleves.ens.fr/home/ychen/
[2] INRIA and ENS, Dept. Informatique, 45 rue d'Ulm, 75005 Paris, France.
http://www.di.ens.fr/~pnguyen/

Abstract. The best lattice reduction algorithm known in practice for high dimension is Schnorr-Euchner's BKZ: all security estimates of lattice cryptosystems are based on NTL's old implementation of BKZ. However, recent progress on lattice enumeration suggests that BKZ and its NTL implementation are no longer optimal, but the precise impact on security estimates was unclear. We assess this impact thanks to extensive experiments with BKZ 2.0, the first state-of-the-art implementation of BKZ incorporating recent improvements, such as Gama-Nguyen-Regev pruning. We propose an efficient simulation algorithm to model the behaviour of BKZ in high dimension with high blocksize ≥ 50, which can predict approximately both the output quality and the running time, thereby revising lattice security estimates. For instance, our simulation suggests that the smallest NTRUSign parameter set, which was claimed to provide at least 93-bit security against key-recovery lattice attacks, actually offers at most 65-bit security.

Security Level

- $\delta = \left(\frac{\lambda_1}{\det(\Lambda)^{1/n}}\right)^{1/n}$ [CN11]
- "Exact" bitlevel correpsondance [LP11]
- Depends on the algorithm

k	δ
80	1.00783
100	1.00696
128	1.00602

$$\log_2(\delta) := \frac{1.8}{\log_2(\frac{T_{BKZ}(\delta)}{2^{30}}) + 110} = \frac{1.8}{k - 30 + 110} = \frac{1.8}{k + 80}$$

Better Key Sizes (and Attacks) for
LWE-Based Encryption

Richard Lindner* Chris Peikert†

November 30, 2010

Abstract

We analyze the concrete security and key sizes of theoretically sound lattice-based encryption schemes based on the "learning with errors" (LWE) problem. Our main contributions are: (1) a new lattice attack on LWE that combines basis reduction with an enumeration algorithm admitting a time/success tradeoff, which performs better than the simple distinguishing attack considered in prior analyses; (2) concrete parameters and security estimates for an LWE-based cryptosystem that is more compact and efficient than the well-known schemes from the literature. Our new key sizes are up to 10 times smaller than prior examples, while providing even stronger concrete security levels.

Security Level

- $\delta = \left(\frac{\lambda_1}{\det(\Lambda)^{1/n}}\right)^{1/n}$ [CN11]
- "Exact" bitlevel correpsondance [LP11]
- Depends on the algorithm

k	δ
80	1.00783
100	1.00696
128	1.00602

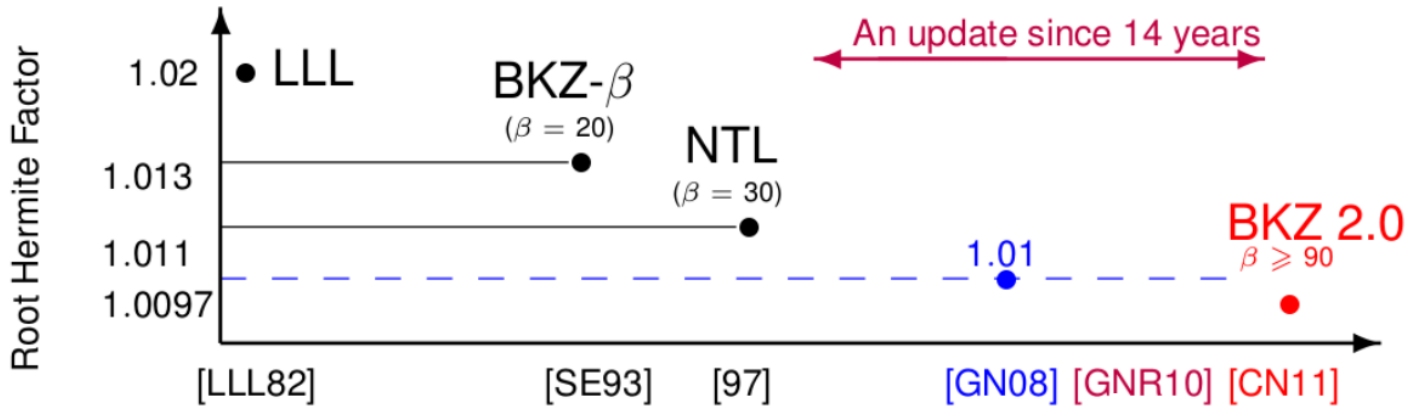


Figure : Borrowed from Yuanmi [CN11]

Outline

NTRUSign

History

- Originally NSS [HPS01]
- NTRUSign [HPSW02]

$$\mathbf{f}, \mathbf{g} = \begin{cases} d \text{ coefficients } +1 \\ N-d \text{ coefficients } 0 \end{cases}$$

$$\mathbf{F}, \mathbf{G} \text{ st. } \mathbf{f} * \mathbf{G} - \mathbf{F} * \mathbf{g} = q$$

$$\mathbf{h} = \mathbf{g} * \mathbf{f}^{-1} \overset{\$}{\leftsquigarrow} \mathcal{R}_q = \mathbb{Z}_q[X]/\langle X^N + 1 \rangle$$

NSS: An NTRU Lattice-Based Signature Scheme

Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman

NTRU Cryptosystems, Inc., 5 Burlington Woods,
Burlington, MA 01803 USA,
jhoff@ntru.com, jpipher@ntru.com, jhs@ntru.com

Abstract. A new authentication and digital signature scheme called the NTRU Signature Scheme (NSS) is introduced. NSS provides an authentication/signature method complementary to the NTRU public key cryptosystem. The hard lattice problem underlying NSS is similar to the hard problem underlying NTRU, and NSS similarly features high speed, low footprint, and easy key creation.

NTRUSign

History

- Originally NSS [HPS01]
  Quickly broken [GS02]
- NTRUSign [HPSW02]

$$\mathbf{f}, \mathbf{g} = \begin{cases} d \text{ coefficients } +1 \\ N-d \text{ coefficients } 0 \end{cases}$$

$$\mathbf{F}, \mathbf{G} \text{ st. } \mathbf{f} * \mathbf{G} - \mathbf{F} * \mathbf{g} = q$$

$$\mathbf{h} = \mathbf{g} * \mathbf{f}^{-1} \overset{\$}{\leftarrow} \mathcal{R}_q = \mathbb{Z}_q[X]/\langle X^N + 1 \rangle$$

Cryptanalysis of the Revised NTRU Signature Scheme

Craig Gentry[1] and Mike Szydlo[2]

[1] DoCoMo USA Labs, San Jose, CA, USA,
cgentry@docomolabs-usa.com
[2] RSA Laboratories, Bedford, MA, USA,
mszydlo@rsasecurity.com

Abstract. In this paper, we describe a three-stage attack against Revised NSS, an NTRU-based signature scheme proposed at the Eurocrypt 2001 conference as an enhancement of the (broken) proceedings version of the scheme. The first stage, which typically uses a transcript of only 4 signatures, effectively cuts the key length in half while completely avoiding the intended hard lattice problem. After an empirically fast second stage, the third stage of the attack combines lattice-based and congruence-based methods in a novel way to recover the private key in polynomial time. This cryptanalysis shows that a passive adversary observing only a few valid signatures can recover the signer's entire private key. We also briefly address the security of NTRUSign, another NTRU-based signature scheme that was recently proposed at the rump session of Asiacrypt 2001. As we explain, some of our attacks on Revised NSS may be extended to NTRUSign, but a *much* longer transcript is necessary. We also indicate how the security of NTRUSign is based on the hardness of several problems, not solely on the hardness of the usual NTRU lattice problem.

NTRUSign

History

- Originally NSS [HPS01]
  Quickly broken [GS02]
- NTRUSign [HPSW02]

$$\mathbf{f}, \mathbf{g} = \begin{cases} d \text{ coefficients } +1 \\ N-d \text{ coefficients } 0 \end{cases}$$

$$\mathbf{F}, \mathbf{G} \text{ st. } \mathbf{f} * \mathbf{G} - \mathbf{F} * \mathbf{g} = q$$

$$\mathbf{h} = \mathbf{g} * \mathbf{f}^{-1} \overset{\$}{\leftsquigarrow} \mathcal{R}_q = \mathbb{Z}_q[X]/\langle X^N + 1 \rangle$$

NTRUSign

History

- Originally NSS [HPS01]
  Quickly broken [GS02]
- NTRUSign [HPSW02]

$$\mathbf{f}, \mathbf{g} = \begin{cases} d \text{ coefficients } +1 \\ N-d \text{ coefficients } 0 \end{cases}$$

$$\mathbf{F}, \mathbf{G} \text{ st. } \mathbf{f} * \mathbf{G} - \mathbf{F} * \mathbf{g} = q$$

$$\mathbf{h} = \mathbf{g} * \mathbf{f}^{-1} \overset{\$}{\leftarrow} \mathcal{R}_q = \mathbb{Z}_q[X]/\langle X^N + 1 \rangle$$

$$\mathbf{P} = \left(\begin{array}{c|c} \mathbf{1} \circlearrowleft & \mathbf{h} \circlearrowleft \\ \hline \mathbf{0} \circlearrowleft & \mathbf{q} \circlearrowleft \end{array} \right) \qquad \mathbf{S} = \left(\begin{array}{c|c} \mathbf{f} \circlearrowleft & \mathbf{g} \circlearrowleft \\ \hline \mathbf{F} \circlearrowleft & \mathbf{G} \circlearrowleft \end{array} \right)$$

NTRUSign

History

- Originally NSS [HPS01]
  Quickly broken [GS02]
- NTRUSign [HPSW02]

$$\mathbf{f}, \mathbf{g} = \begin{cases} d \text{ coefficients } +1 \\ N-d \text{ coefficients } 0 \end{cases}$$

$$\mathbf{F}, \mathbf{G} \text{ st. } \mathbf{f} * \mathbf{G} - \mathbf{F} * \mathbf{g} = q$$

$$\mathbf{h} = \mathbf{g} * \mathbf{f}^{-1} \overset{\$}{\leftsquigarrow} \mathcal{R}_q = \mathbb{Z}_q[X]/\langle X^N + 1 \rangle$$

$$\mathbf{P} = \left(\begin{array}{c|c} \mathbf{1} \circlearrowleft & \mathbf{h} \circlearrowleft \\ \hline \mathbf{0} \circlearrowleft & \mathbf{q} \circlearrowleft \end{array} \right) \qquad \mathbf{S} = \left(\begin{array}{c|c} \mathbf{f} \circlearrowleft & \mathbf{g} \circlearrowleft \\ \hline \mathbf{F} \circlearrowleft & \mathbf{G} \circlearrowleft \end{array} \right)$$

NTRUSign

History

- Originally NSS [HPS01]
  Quickly broken [GS02]
- NTRUSign [HPSW02]

$$\mathbf{f}, \mathbf{g} = \begin{cases} d \text{ coefficients } +1 \\ N-d \text{ coefficients } 0 \end{cases}$$

$$\mathbf{F}, \mathbf{G} \text{ st. } \mathbf{f} * \mathbf{G} - \mathbf{F} * \mathbf{g} = q$$

$$\mathbf{h} = \mathbf{g} * \mathbf{f}^{-1} \overset{\$}{\leftsquigarrow} \mathcal{R}_q = \mathbb{Z}_q[X]/\langle X^N + 1 \rangle$$

$$\mathbf{P} = \left(\begin{array}{c|c} \mathbf{1} \circlearrowleft & \mathbf{h} \circlearrowleft \\ \hline \mathbf{0} \circlearrowleft & \mathbf{q} \circlearrowleft \end{array} \right) \qquad \mathbf{S} = \left(\begin{array}{c|c} \mathbf{f} \circlearrowleft & \mathbf{g} \circlearrowleft \\ \hline \mathbf{F} \circlearrowleft & \mathbf{G} \circlearrowleft \end{array} \right)$$

NTRU lattice: $\Lambda_{\mathbf{h},q} = \{(\mathbf{u}, \mathbf{u} * \mathbf{h} \mod q), \mathbf{u} \in \mathcal{R}_q\}$

NTRUSign

Sign

Given $\mu \in \{0,1\}^*$ to sign:

- Define $\mathbf{m} = \mathcal{H}(\mu)$
- Solve CVP with target $(\mathbf{0}, \mathbf{m})$ and good basis $\mathbf{S}$

Verify

Given the signature $\mathbf{s}$, check:

- It's a lattice point (using bad basis $\mathbf{P}$)
- Not far from $(\mathbf{0}, \mathbf{m})$

NTRUSign

Sign

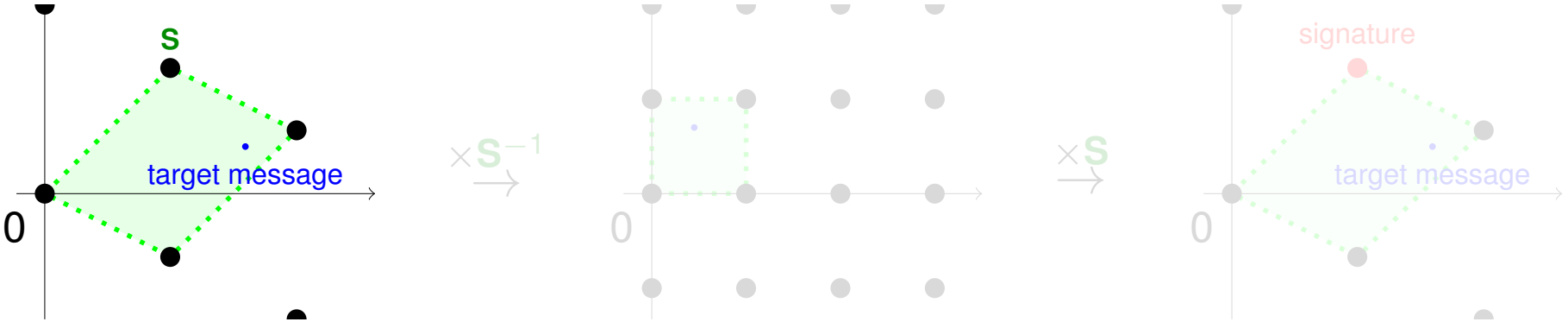
Given $\mu \in \{0,1\}^*$ to sign:

- Define $\mathbf{m} = \mathcal{H}(\mu)$
- Solve CVP with target $(\mathbf{0}, \mathbf{m})$ and good basis $\mathbf{S}$

Verify

Given the signature $\mathbf{s}$, check:

- It's a lattice point (using bad basis $\mathbf{P}$)
- Not far from $(\mathbf{0}, \mathbf{m})$



NTRUSign

Sign

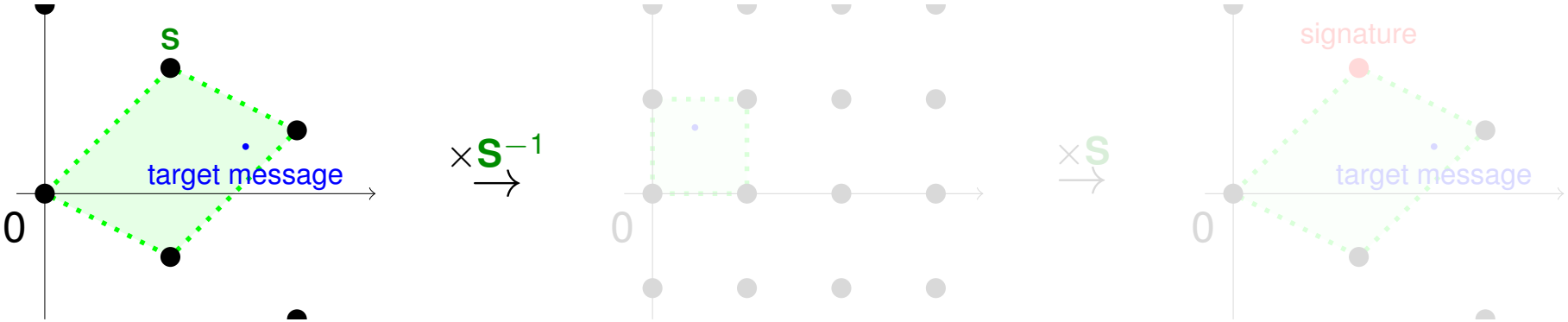
Given $\mu \in \{0,1\}^*$ to sign:

- Define $\mathbf{m} = \mathcal{H}(\mu)$
- Solve CVP with target $(\mathbf{0}, \mathbf{m})$ and good basis $\mathbf{S}$

Verify

Given the signature $\mathbf{s}$, check:

- It's a lattice point (using bad basis $\mathbf{P}$)
- Not far from $(\mathbf{0}, \mathbf{m})$



NTRUSign

Sign

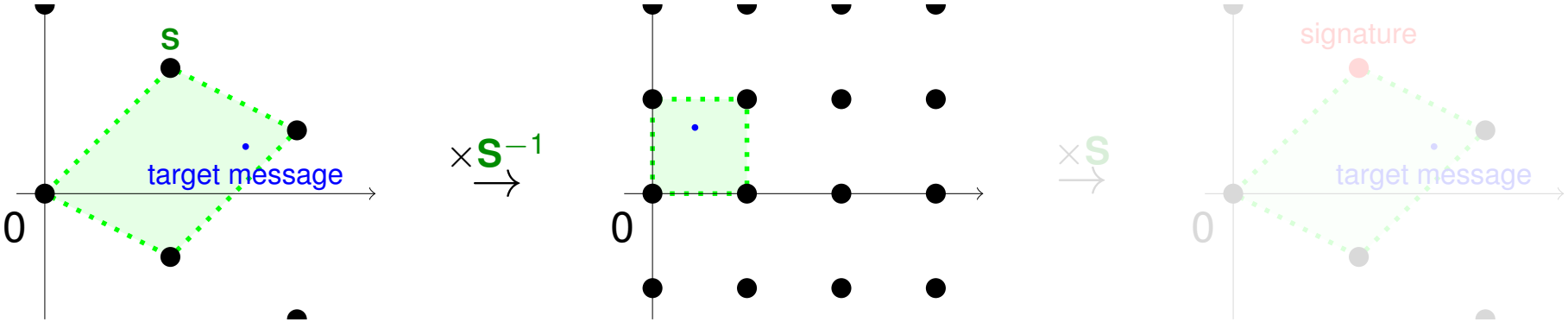
Given $\mu \in \{0,1\}^*$ to sign:

- Define $\mathbf{m} = \mathcal{H}(\mu)$
- Solve CVP with target $(\mathbf{0}, \mathbf{m})$ and good basis $\mathbf{S}$

Verify

Given the signature $\mathbf{s}$, check:

- It's a lattice point (using bad basis $\mathbf{P}$)
- Not far from $(\mathbf{0}, \mathbf{m})$



NTRUSign

Sign

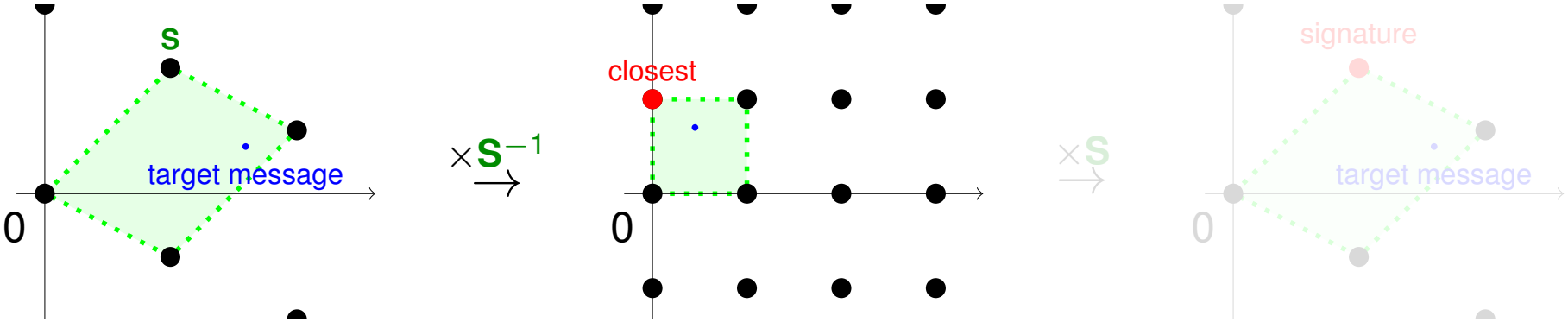
Given $\mu \in \{0,1\}^*$ to sign:

- Define $\mathbf{m} = \mathcal{H}(\mu)$
- Solve CVP with target $(\mathbf{0}, \mathbf{m})$ and good basis $\mathbf{S}$

Verify

Given the signature $\mathbf{s}$, check:

- It's a lattice point (using bad basis $\mathbf{P}$)
- Not far from $(\mathbf{0}, \mathbf{m})$



NTRUSign

Sign

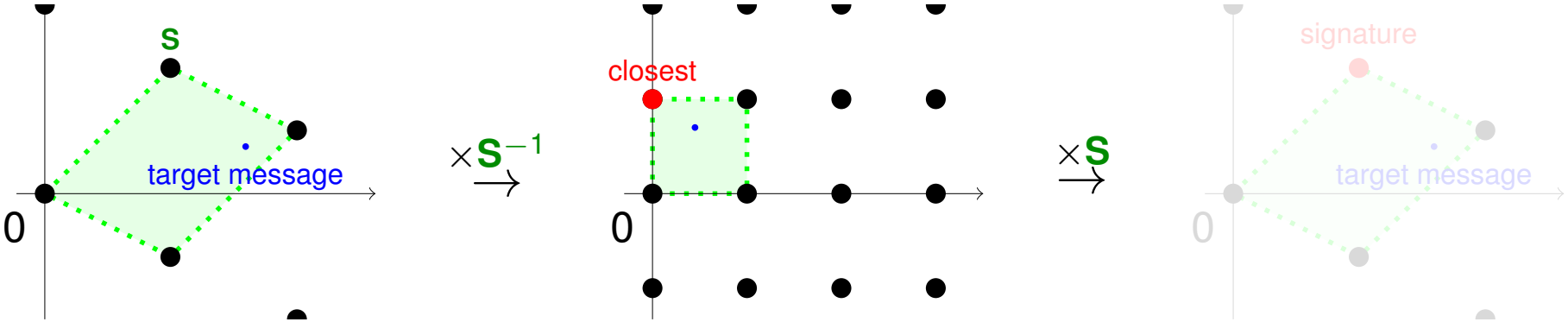
Given $\mu \in \{0,1\}^*$ to sign:

- Define $\mathbf{m} = \mathcal{H}(\mu)$
- Solve CVP with target $(\mathbf{0}, \mathbf{m})$ and good basis $\mathbf{S}$

Verify

Given the signature $\mathbf{s}$, check:

- It's a lattice point (using bad basis $\mathbf{P}$)
- Not far from $(\mathbf{0}, \mathbf{m})$



NTRUSign

Sign

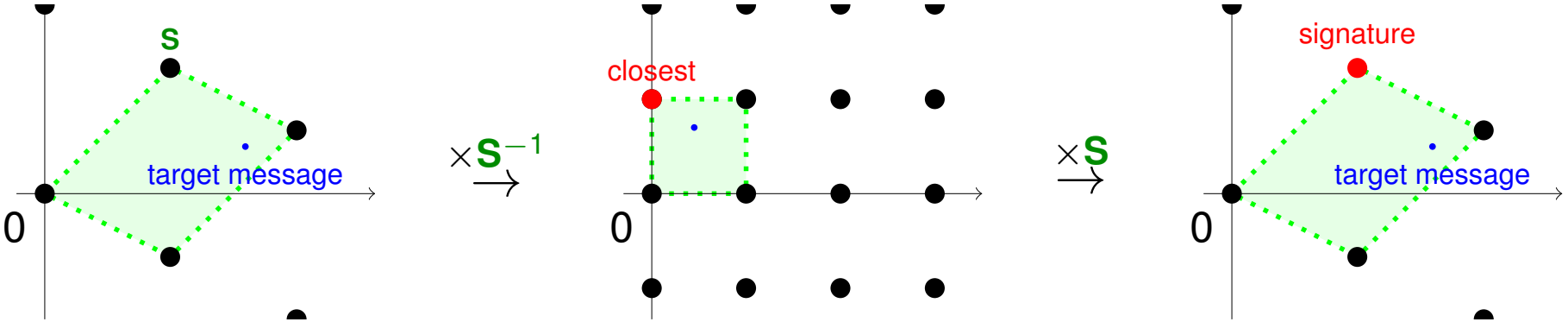
Given $\mu \in \{0,1\}^*$ to sign:

- Define $\mathbf{m} = \mathcal{H}(\mu)$
- Solve CVP with target $(\mathbf{0}, \mathbf{m})$ and good basis $\mathbf{S}$

Verify

Given the signature $\mathbf{s}$, check:

- It's a lattice point (using bad basis $\mathbf{P}$)
- Not far from $(\mathbf{0}, \mathbf{m})$



NTRUSign

Sign

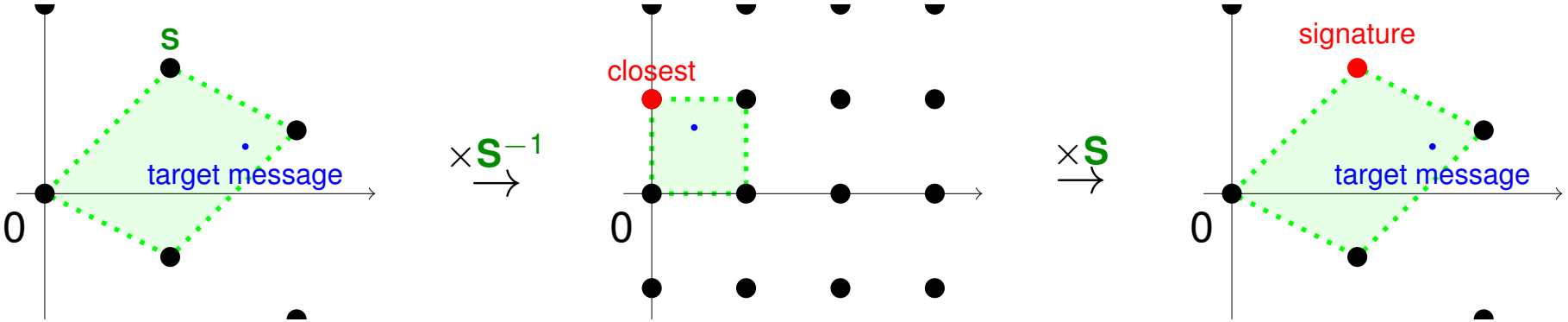
Given $\mu \in \{0,1\}^*$ to sign:

- Define $\mathbf{m} = \mathcal{H}(\mu)$
- Solve CVP with target $(\mathbf{0}, \mathbf{m})$ and good basis **S**

Verify

Given the signature **s**, check:

- It's a lattice point (using bad basis **P**)
- Not far from $(\mathbf{0}, \mathbf{m})$



NTRUSign

Sign

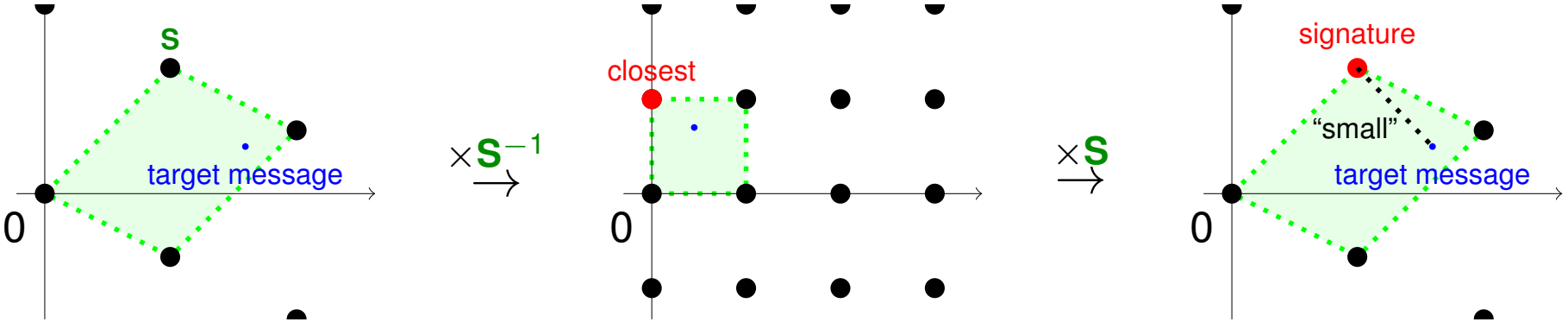
Given $\mu \in \{0,1\}^*$ to sign:

- Define $\mathbf{m} = \mathcal{H}(\mu)$
- Solve CVP with target $(\mathbf{0}, \mathbf{m})$ and good basis **S**

Verify

Given the signature **s**, check:

- It's a lattice point (using bad basis **P**)
- Not far from $(\mathbf{0}, \mathbf{m})$



NTRUSign

SIGNATURE SIZE (IN BITS)

security	80	112	128	160
NTRUSign	1256	1576	1784	2367
ECDSA_{sign}	320	448	512	640
RSA	1024	2048	3072	4096

NTRUSign runs faster !

NTRUSign

SIGNATURE SIZE (IN BITS)

security	80	112	128	160
NTRUSign	1256	1576	1784	2367
ECDSA_{sign}	320	448	512	640
RSA	1024	2048	3072	4096

NTRUSign runs faster !

NTRUSign

SIGNATURE SIZE (IN BITS)

security	80	112	128	160
NTRUSign	1256	1576	1784	2367
ECDSA_{sign}	320	448	512	640
RSA	1024	2048	3072	4096

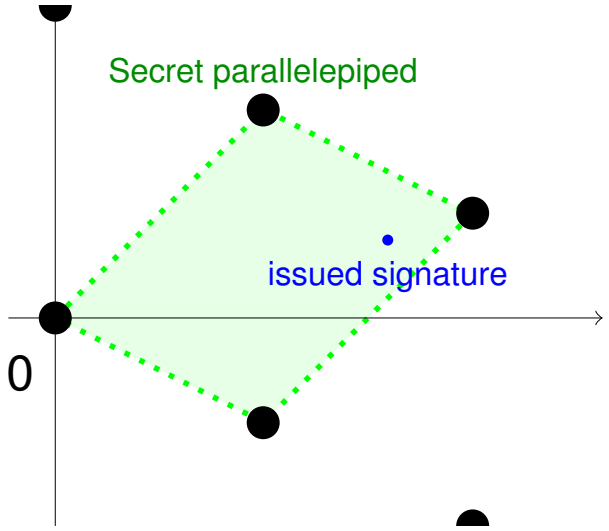
NTRUSign runs faster !

But...

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

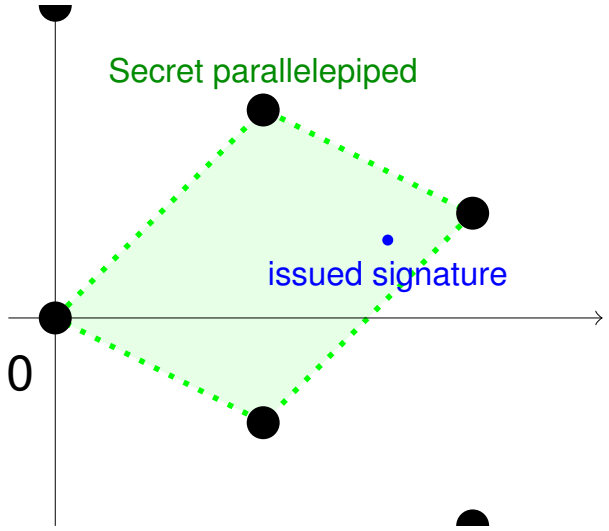


Number of signature issued : 1

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

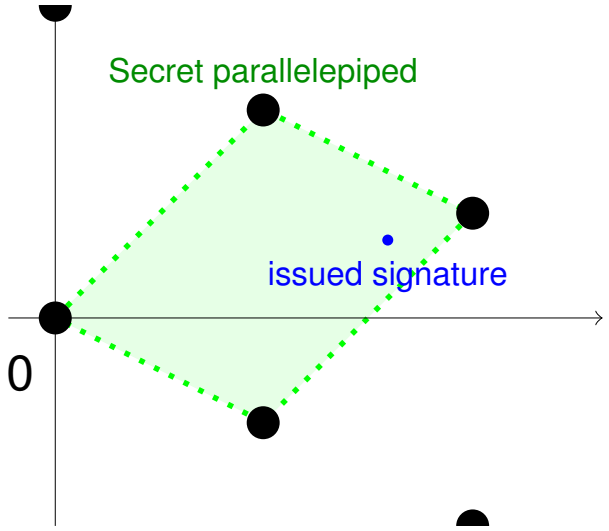


Number of signature issued : 1

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

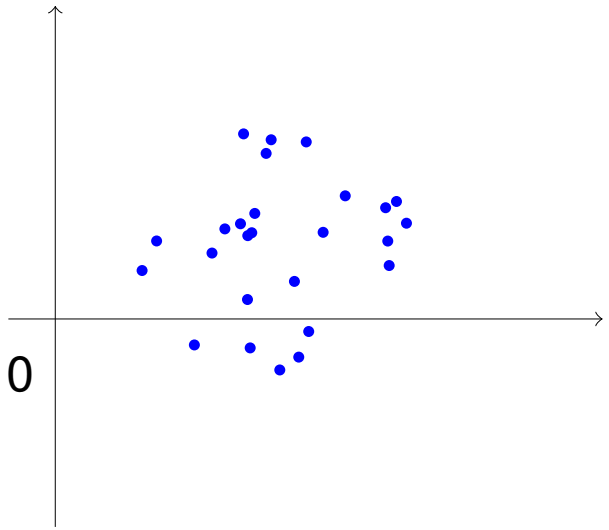


Number of signature issued : 1

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

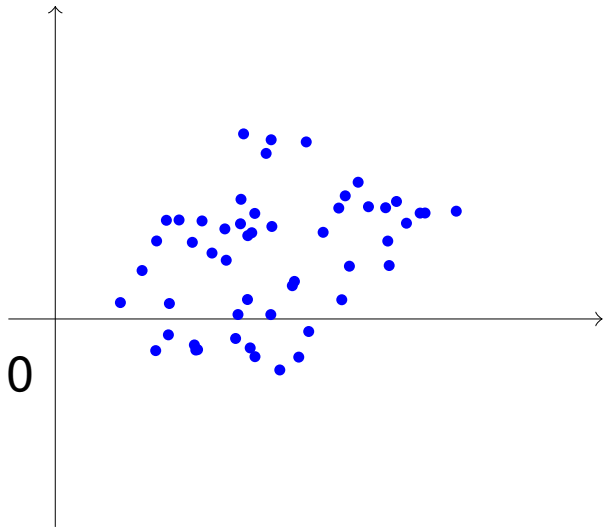


Number of signatures issued : 25

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

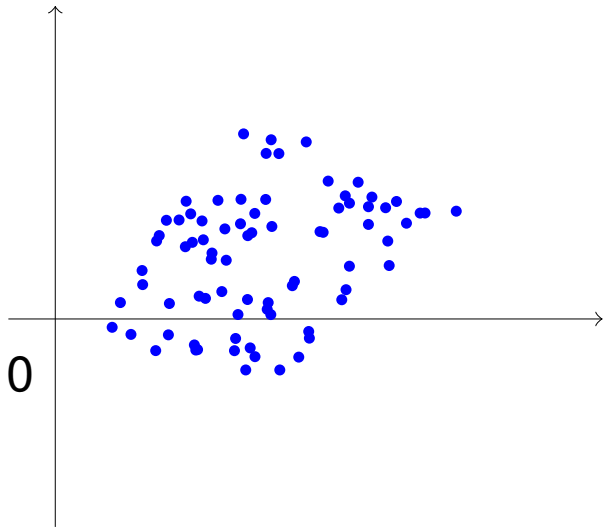


Number of signatures issued : 50

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

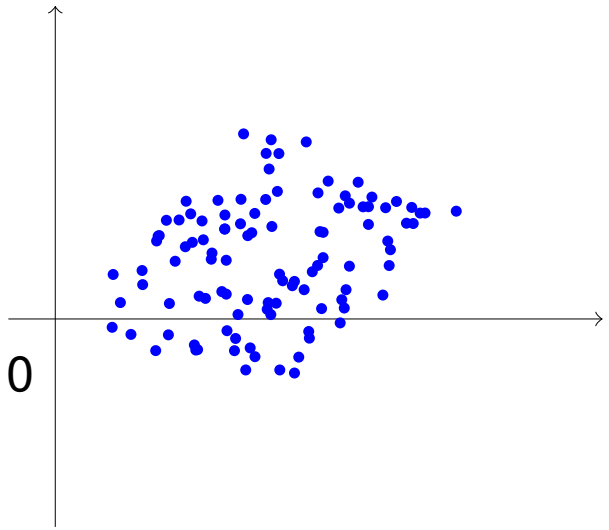


Number of signatures issued : 75

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

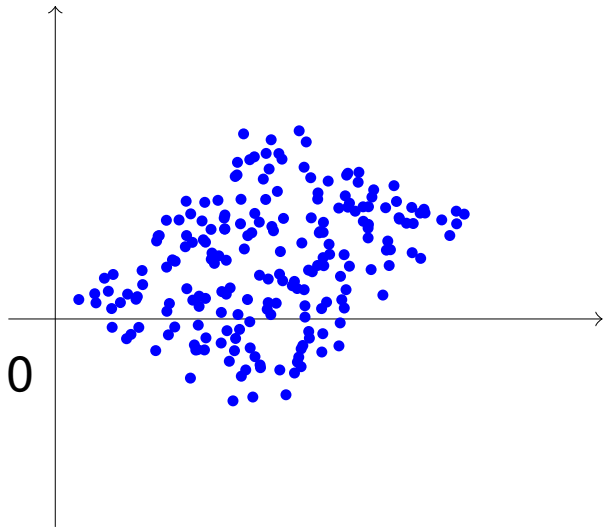


Number of signatures issued : 100

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

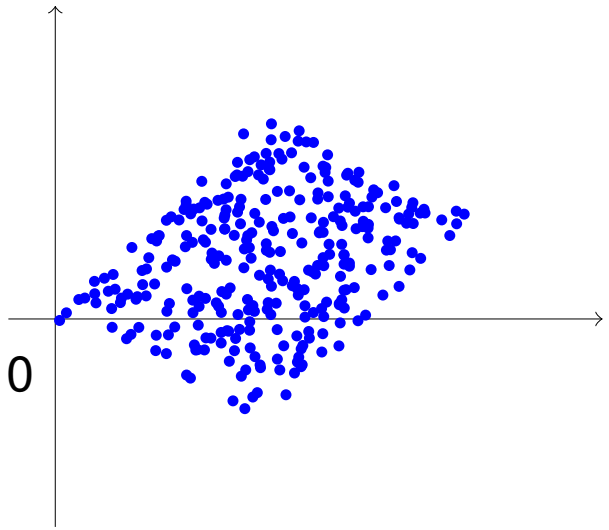


Number of signatures issued : 200

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

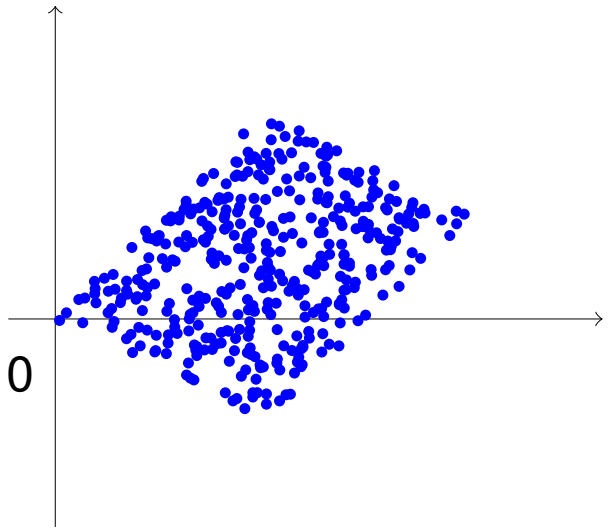


Number of signatures issued : 300

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

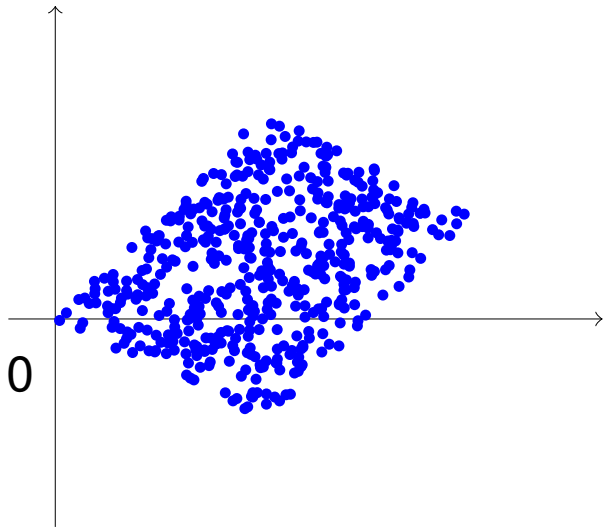


Number of signatures issued : 400

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

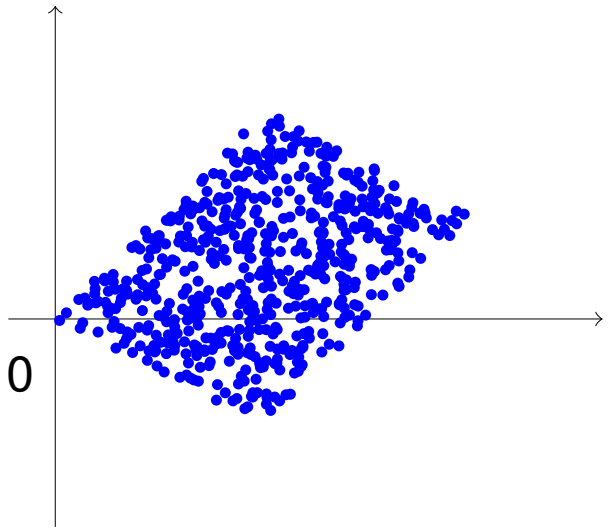


Number of signatures issued : 500

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

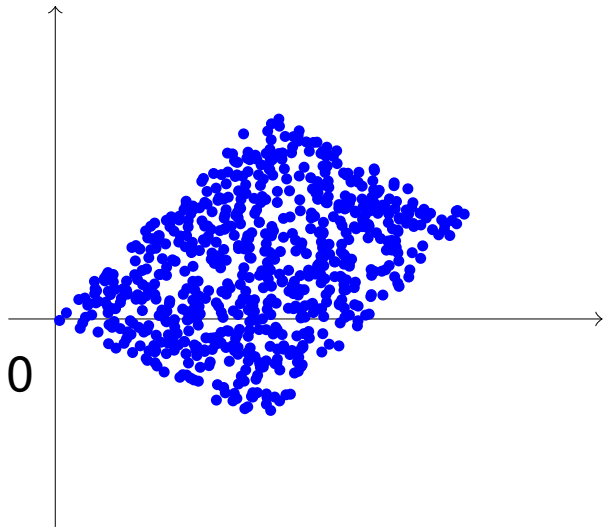


Number of signatures issued : 600

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

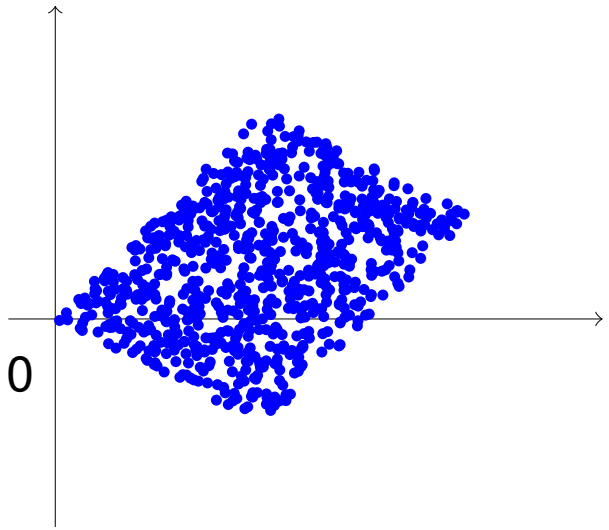


Number of signatures issued : 700

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

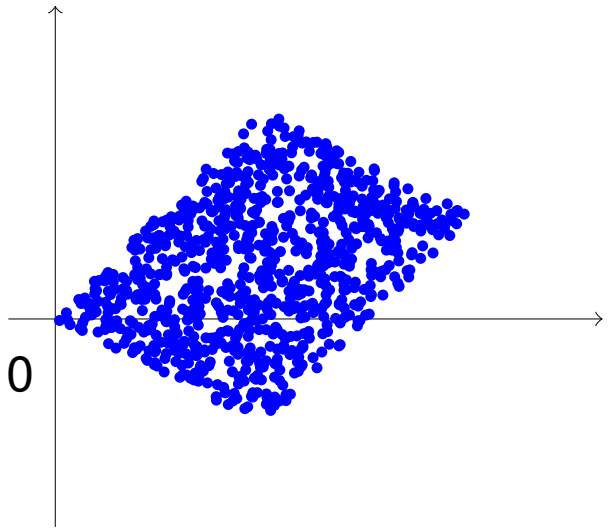


Number of signatures issued : 800

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]

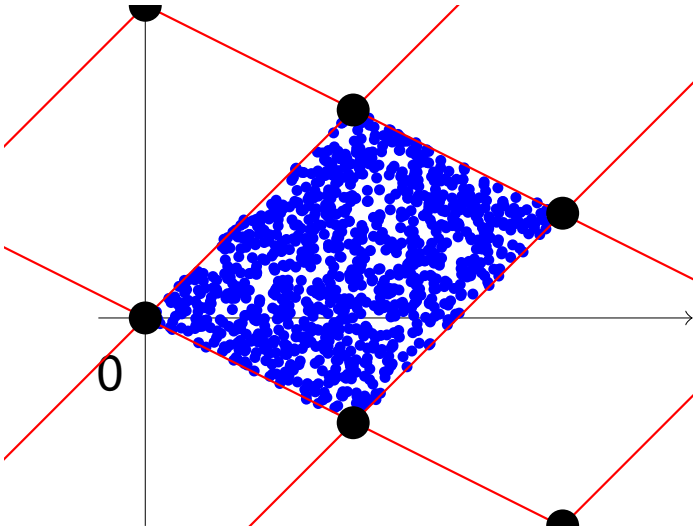


Number of signatures issued : 900

Problem : Not Zero-Knowledge

Key-recovery attacks

- Only a few signatures for original scheme [NR06]
- And a little more to break coutermeasures [DN12]



Number of signatures issued : 1000

Outline

Overview

KeyGen

- Secret key : $\mathbf{S} \xleftarrow{\$} \{-d, \ldots, 0, \ldots, d\}^{m \times k}$
- Public key : $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $\mathbf{T} = \mathbf{A} \cdot \mathbf{S} \in \mathbb{Z}_q^{n \times k}$

Sign

First stage [Finding pre-image]

- map μ to a space element $\mathbf{c}$
- $\mathbf{Sc}$ is a short pre-image of $\mathbf{Tc}$

Second stage [Hiding pre-image]

- Add gaussian noise $\mathbf{y}$ to $\mathbf{Sc}$
- Apply rejection sampling to avoid leakage

Overview

KeyGen

- Secret key : $\mathbf{S} \xleftarrow{\$} \{-d, \ldots, 0, \ldots, d\}^{m \times k}$
- Public key : $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $\mathbf{T} = \mathbf{A} \cdot \mathbf{S} \in \mathbb{Z}_q^{n \times k}$

Sign

First stage [Finding pre-image]

- map μ to a space element $\mathbf{c}$
- $\mathbf{Sc}$ is a short pre-image of $\mathbf{Tc}$

Second stage [Hiding pre-image]

- Add gaussian noise $\mathbf{y}$ to $\mathbf{Sc}$
- Apply rejection sampling to avoid leakage

Overview

KeyGen

- Secret key : $\mathbf{S} \xleftarrow{\$} \{-d, \ldots, 0, \ldots, d\}^{m \times k}$
- Public key : $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $\mathbf{T} = \mathbf{A} \cdot \mathbf{S} \in \mathbb{Z}_q^{n \times k}$

Sign

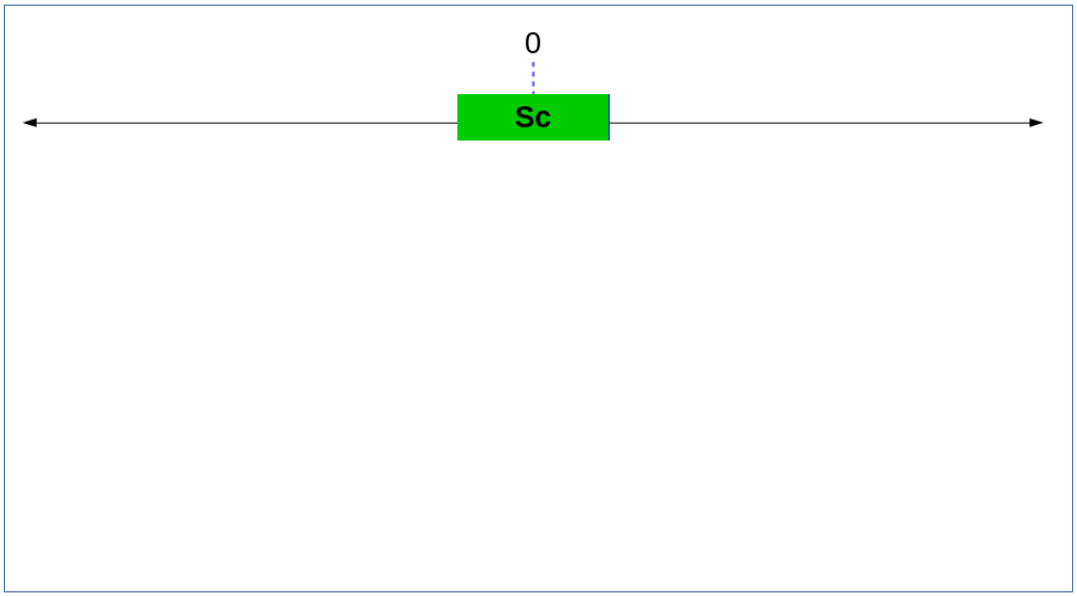
First stage [Finding pre-image]

- map μ to a space element $\mathbf{c}$
- $\mathbf{Sc}$ is a short pre-image of $\mathbf{Tc}$

Second stage [Hiding pre-image]

- Add gaussian noise $\mathbf{y}$ to $\mathbf{Sc}$
- Apply rejection sampling to avoid leakage

Overview

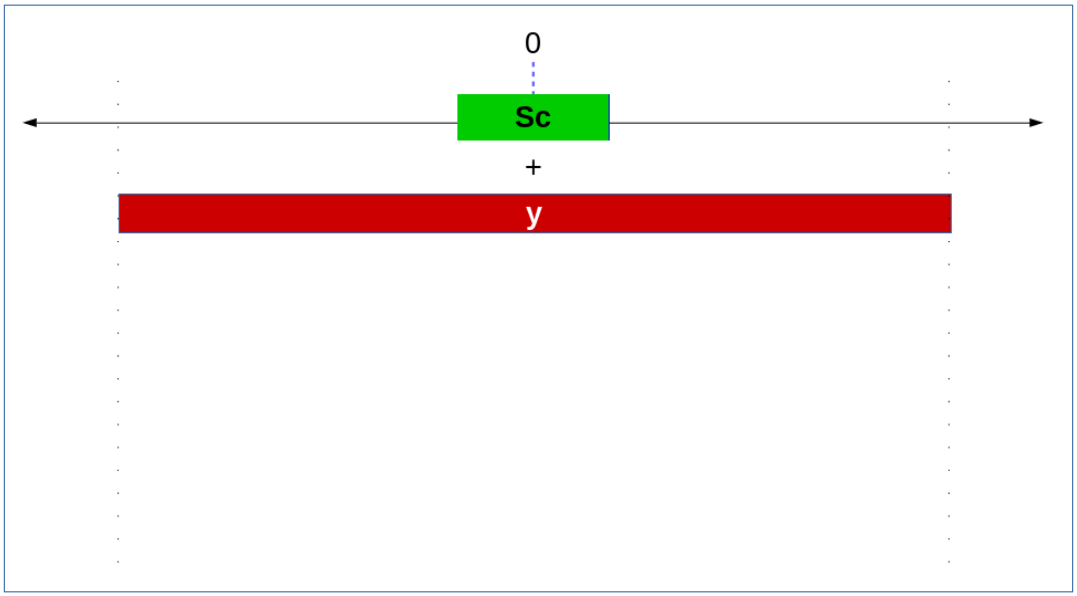


Verify

Given $(\mathbf{z}, \mathbf{c})$, check that :

- $H(\underbrace{\mathbf{Az} - \mathbf{Tc}}_{\mathbf{A}(\mathbf{Sc}+\mathbf{y})-\mathbf{ASc}}, \mu) = \mathbf{c}$ $\rightarrow$ it is a lattice vector
- $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ $\rightarrow$ it has reasonnable norm

Overview

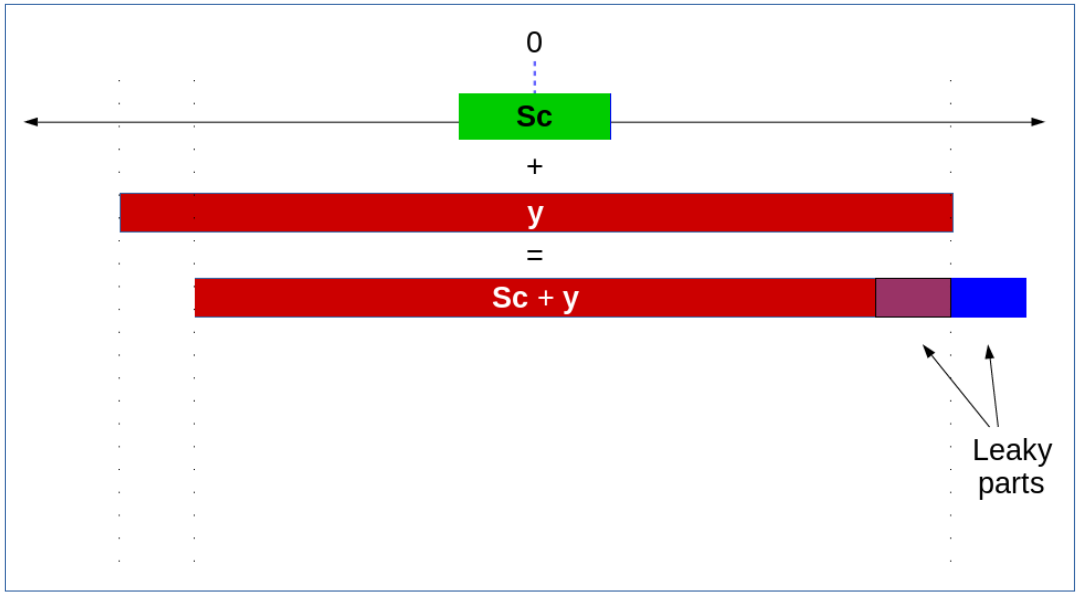


Verify

Given $(\mathbf{z}, \mathbf{c})$, check that :

- $H(\underbrace{\mathbf{Az} - \mathbf{Tc}}_{\mathbf{A}(\mathbf{Sc}+\mathbf{y})-\mathbf{ASc}}, \mu) = \mathbf{c}$ $\rightarrow$ it is a lattice vector
- $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ $\rightarrow$ it has reasonnable norm

Overview

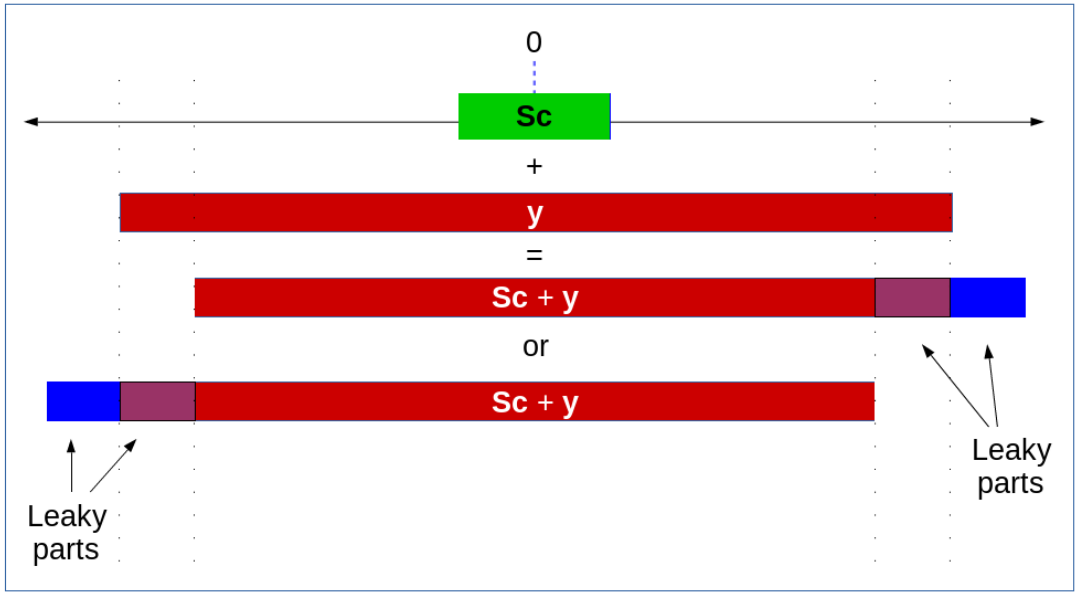


Verify

Given $(\mathbf{z}, \mathbf{c})$, check that :

- $H(\underbrace{\mathbf{Az} - \mathbf{Tc}}_{\mathbf{A}(\mathbf{Sc}+\mathbf{y})-\mathbf{ASc}}, \mu) = \mathbf{c}$ $\rightarrow$ it is a lattice vector
- $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ $\rightarrow$ it has reasonnable norm

Overview

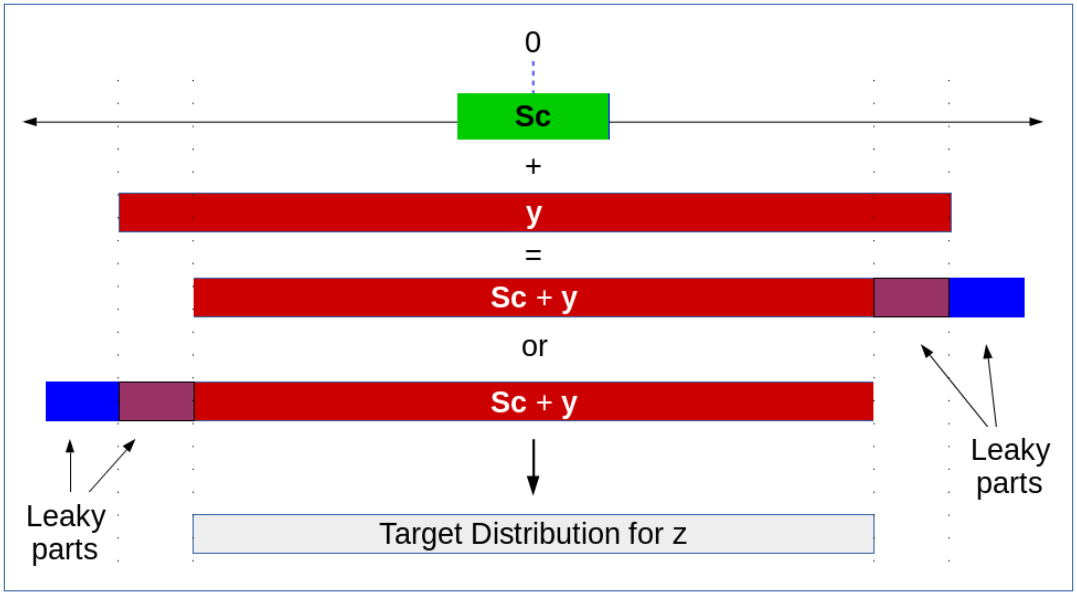


Verify

Given $(\mathbf{z}, \mathbf{c})$, check that :

- $H(\underbrace{\mathbf{Az} - \mathbf{Tc}}_{\mathbf{A}(\mathbf{Sc}+\mathbf{y})-\mathbf{ASc}}, \mu) = \mathbf{c}$ $\rightarrow$ it is a lattice vector
- $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ $\rightarrow$ it has reasonnable norm

Overview

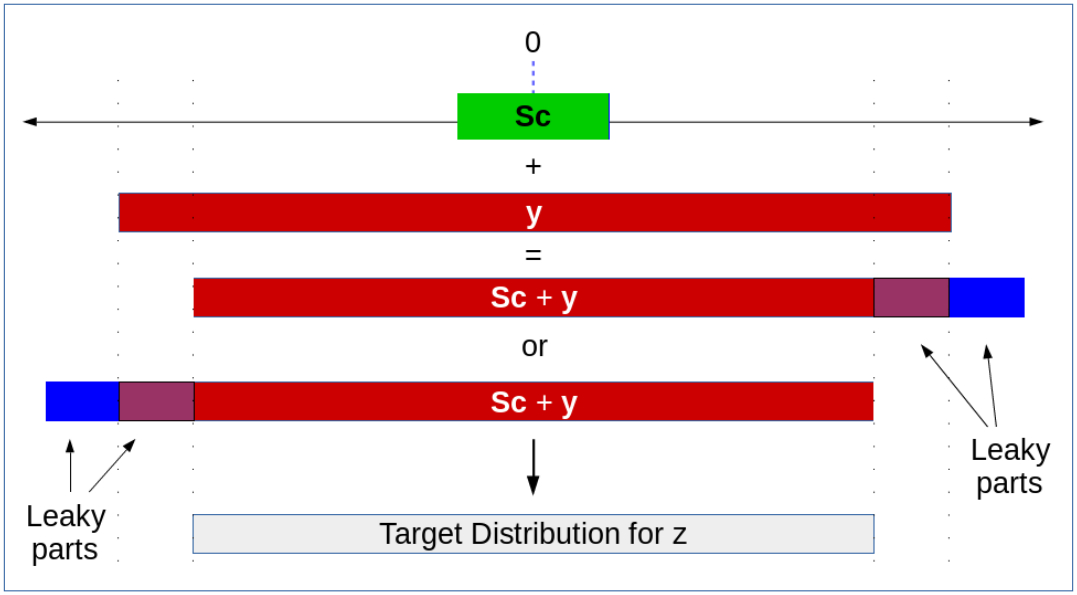


Verify

Given $(\mathbf{z}, \mathbf{c})$, check that :

- $H(\underbrace{\mathbf{Az} - \mathbf{Tc}}_{\mathbf{A}(\mathbf{Sc}+\mathbf{y})-\mathbf{ASc}}, \mu) = \mathbf{c}$ $\rightarrow$ it is a lattice vector
- $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ $\rightarrow$ it has reasonnable norm

Overview



Verify

Given $(\mathbf{z}, \mathbf{c})$, check that :

- $H(\underbrace{\mathbf{Az} - \mathbf{Tc}}_{\mathbf{A}(\mathbf{Sc}+\mathbf{y})-\mathbf{ASc}}, \mu) = \mathbf{c}$ $\rightarrow$ it is a lattice vector
- $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ $\rightarrow$ it has reasonnable norm

Sets of parameters

100 bits of security

n	512	512	512	512	512
m	8,786	8,139	3,253	1,024	1,024
k	80	512	512	512	512
$\log_2(q)$	27	25	33	18	26
d	1	1	31	1	31
M (retries)	2.72	2.72	2.72	7.4	7.4
$\approx$ sign size	163,000	142,300	73,000	14,500	19,500
$\approx$ pk size	2^{20}	$2^{22.5}$	2^{23}	$2^{19.5}$	$2^{21.5}$
$\approx$ sk size	2^{20}	$2^{22.5}$	2^{23}	$2^{22.1}$	$2^{22.7}$

Outline

Presentation

KeyGen

$\mathbf{S} = (\mathbf{f}, \mathbf{g}) \in \mathcal{R}_q$ and $\mathbf{P} = (\mathbf{1}, \mathbf{h} = \mathbf{g} * \mathbf{f}^{-1})$ as in NTRUSign

Sign

Given $\mu \in \{0,1\}^*$ to sign :

- $\mathbf{y} \xleftarrow{\$} D_\sigma^{2N}$
- $\mathbf{e} = \mathcal{H}(\mathbf{Py}, \mu)$
- $\mathbf{s} = \text{NTRUSign}_{\mathbf{S}}(\mathbf{e})$

Output $\mathbf{x} = \mathbf{e} - \mathbf{s} + \mathbf{y}$ with some proba

Presentation

KeyGen

$$\mathbf{S} = (\mathbf{f}, \mathbf{g}) \in \mathcal{R}_q \text{ and } \mathbf{P} = (\mathbf{1}, \mathbf{h} = \mathbf{g} * \mathbf{f}^{-1}) \text{ as in NTRUSign}$$

Sign

Given $\mu \in \{0,1\}^*$ to sign :

- $\mathbf{y} \xleftarrow{\$} D_\sigma^{2N}$
- $\mathbf{e} = \mathcal{H}(\mathbf{Py}, \mu)$
- $\mathbf{s} = \text{NTRUSign}_{\mathbf{S}}(\mathbf{e})$

Output $\mathbf{x} = \mathbf{e} - \mathbf{s} + \mathbf{y}$ with some proba

Presentation

KeyGen

$$\mathbf{S} = (\mathbf{f}, \mathbf{g}) \in \mathcal{R}_q \text{ and } \mathbf{P} = (\mathbf{1}, \mathbf{h} = \mathbf{g} * \mathbf{f}^{-1}) \text{ as in NTRUSign}$$

Sign

Given $\mu \in \{0,1\}^*$ to sign :

- $\mathbf{y} \xleftarrow{\$} D_\sigma^{2N}$
- $\mathbf{e} = \mathcal{H}(\mathbf{Py}, \mu)$
- $\mathbf{s} = \text{NTRUSign}_{\mathbf{S}}(\mathbf{e})$

Output $\mathbf{x} = \mathbf{e} - \mathbf{s} + \mathbf{y}$ with some proba

Presentation

KeyGen

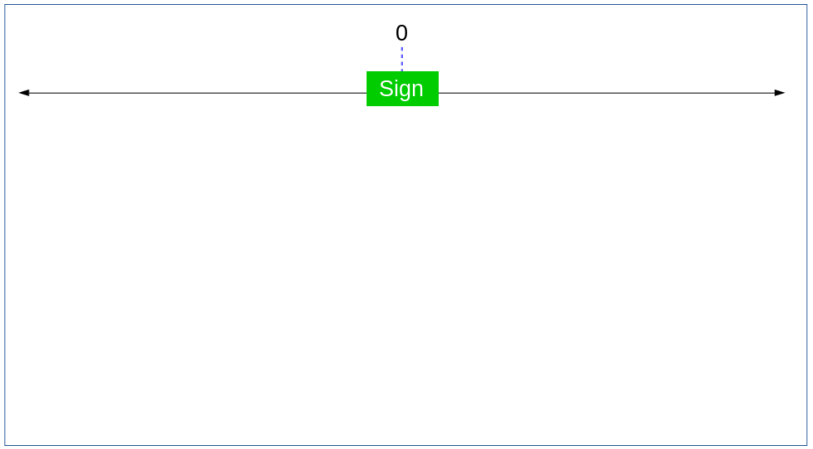
$$\mathbf{S} = (\mathbf{f}, \mathbf{g}) \in \mathcal{R}_q \text{ and } \mathbf{P} = (\mathbf{1}, \mathbf{h} = \mathbf{g} * \mathbf{f}^{-1}) \text{ as in NTRUSign}$$

Sign

Given $\mu \in \{0,1\}^*$ to sign :

- $\mathbf{y} \xleftarrow{\$} D_\sigma^{2N}$
- $\mathbf{e} = \mathcal{H}(\mathbf{Py}, \mu)$
- $\mathbf{s} = \text{NTRUSign}_{\mathbf{S}}(\mathbf{e})$

Output $\mathbf{x} = \mathbf{e} - \mathbf{s} + \mathbf{y}$ with some proba



Presentation

KeyGen

$$\mathbf{S} = (\mathbf{f}, \mathbf{g}) \in \mathcal{R}_q \text{ and } \mathbf{P} = (\mathbf{1}, \mathbf{h} = \mathbf{g} * \mathbf{f}^{-1}) \text{ as in NTRUSign}$$

Sign

Given $\mu \in \{0,1\}^*$ to sign :

- $\mathbf{y} \xleftarrow{\$} D_\sigma^{2N}$
- $\mathbf{e} = \mathcal{H}(\mathbf{Py}, \mu)$
- $\mathbf{s} = \text{NTRUSign}_{\mathbf{S}}(\mathbf{e})$

Output $\mathbf{x} = \mathbf{e} - \mathbf{s} + \mathbf{y}$ with some proba

Presentation

KeyGen

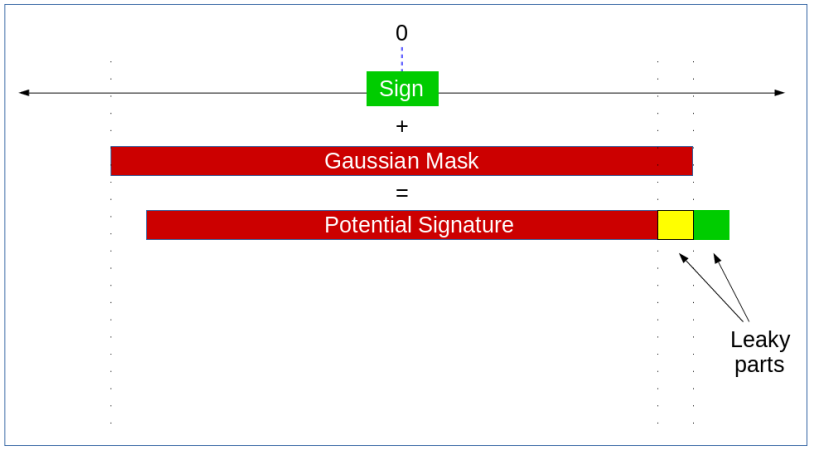
$$\mathbf{S} = (\mathbf{f}, \mathbf{g}) \in \mathcal{R}_q \text{ and } \mathbf{P} = (\mathbf{1}, \mathbf{h} = \mathbf{g} * \mathbf{f}^{-1}) \text{ as in NTRUSign}$$

Sign

Given $\mu \in \{0,1\}^*$ to sign :

- $\mathbf{y} \xleftarrow{\$} D_\sigma^{2N}$
- $\mathbf{e} = \mathcal{H}(\mathbf{Py}, \mu)$
- $\mathbf{s} = \text{NTRUSign}_{\mathbf{S}}(\mathbf{e})$

Output $\mathbf{x} = \mathbf{e} - \mathbf{s} + \mathbf{y}$ with some proba



Presentation

KeyGen

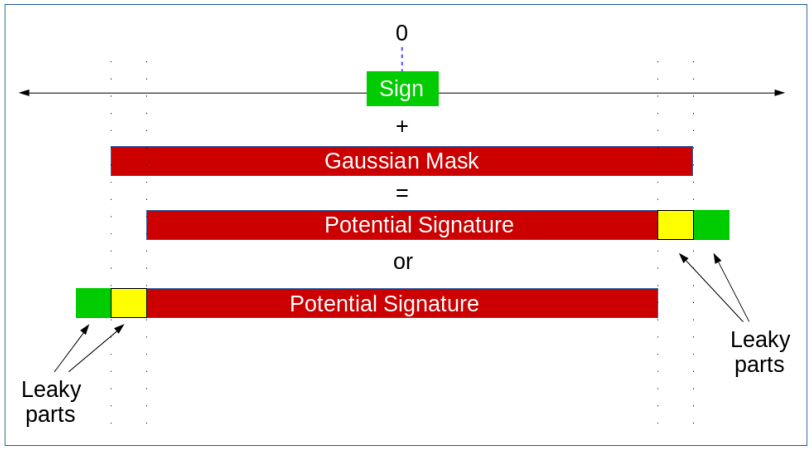
$$\mathbf{S} = (\mathbf{f}, \mathbf{g}) \in \mathcal{R}_q \text{ and } \mathbf{P} = (\mathbf{1}, \mathbf{h} = \mathbf{g} * \mathbf{f}^{-1}) \text{ as in NTRUSign}$$

Sign

Given $\mu \in \{0,1\}^*$ to sign :

- $\mathbf{y} \xleftarrow{\$} D_\sigma^{2N}$
- $\mathbf{e} = \mathcal{H}(\mathbf{Py}, \mu)$
- $\mathbf{s} = \text{NTRUSign}_{\mathbf{S}}(\mathbf{e})$

Output $\mathbf{x} = \mathbf{e} - \mathbf{s} + \mathbf{y}$ with some proba



Presentation

KeyGen

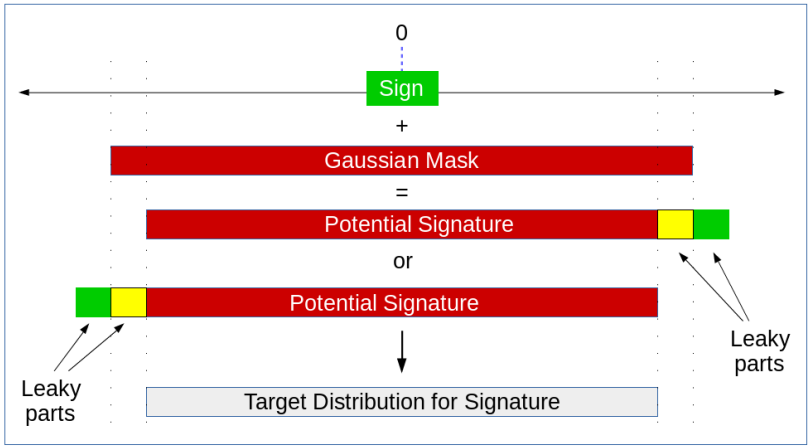
$$\mathbf{S} = (\mathbf{f}, \mathbf{g}) \in \mathcal{R}_q \text{ and } \mathbf{P} = (\mathbf{1}, \mathbf{h} = \mathbf{g} * \mathbf{f}^{-1}) \text{ as in NTRUSign}$$

Sign

Given $\mu \in \{0,1\}^*$ to sign :

- $\mathbf{y} \xleftarrow{\$} D_\sigma^{2N}$
- $\mathbf{e} = \mathcal{H}(\mathbf{Py}, \mu)$
- $\mathbf{s} = \text{NTRUSign}_{\mathbf{S}}(\mathbf{e})$

Output $\mathbf{x} = \mathbf{e} - \mathbf{s} + \mathbf{y}$ with some proba



Presentation

KeyGen

$$\mathbf{S} = (\mathbf{f}, \mathbf{g}) \in \mathcal{R}_q \text{ and } \mathbf{P} = (\mathbf{1}, \mathbf{h} = \mathbf{g} * \mathbf{f}^{-1}) \text{ as in NTRUSign}$$

Sign

Given $\mu \in \{0,1\}^*$ to sign :

- $\mathbf{y} \xleftarrow{\$} D_\sigma^{2N}$
- $\mathbf{e} = \mathcal{H}(\mathbf{P}\mathbf{y}, \mu)$
- $\mathbf{s} = \text{NTRUSign}_{\mathbf{S}}(\mathbf{e})$

Output $\mathbf{x} = \mathbf{e} - \mathbf{s} + \mathbf{y}$ with some proba

Verify

Given $\mathbf{x}$, check that :

- $\mathcal{H}(\mathbf{P}\mathbf{x} - \mathbf{e}, \mu) = \mathbf{e}$
- $\|\mathbf{x}\| \leq \eta\sigma\sqrt{2N}$

Outline

Security Sketch

Guess the secret key ?

i.e. finding $\mathbf{S} = (\mathbf{f}, \mathbf{g})$ from $\mathbf{P} = (\mathbf{1}, \mathbf{h})$

Then you have an algorithm with unexpectedly low Hermite factor...

Find back the leaky part ?

Unlikely due to the distribution we picked $\mathbf{y}$ from

Forge a signature ?

Then we can solve SIS on all NTRU-like lattices.

Security Sketch

Guess the secret key ?

i.e. finding $\mathbf{S} = (\mathbf{f}, \mathbf{g})$ from $\mathbf{P} = (\mathbf{1}, \mathbf{h})$

Then you have an algorithm with unexpectedly low Hermite factor...

Find back the leaky part ?

Unlikely due to the distribution we picked $\mathbf{y}$ from

Forge a signature ?

Then we can solve SIS on all NTRU-like lattices.

Security Sketch

Guess the secret key ?

i.e. finding $\mathbf{S} = (\mathbf{f}, \mathbf{g})$ from $\mathbf{P} = (\mathbf{1}, \mathbf{h})$
Then you have an algorithm with unexpectedly low Hermite factor...

Find back the leaky part ?

Unlikely due to the distribution we picked $\mathbf{y}$ from

Forge a signature ?

Then we can solve SIS on all NTRU-like lattices.

Security Sketch

Guess the secret key ?

i.e. finding $\mathbf{S} = (\mathbf{f}, \mathbf{g})$ from $\mathbf{P} = (\mathbf{1}, \mathbf{h})$
Then you have an algorithm with unexpectedly low Hermite factor... Lucky you !

Find back the leaky part ?

Unlikely due to the distribution we picked $\mathbf{y}$ from

Forge a signature ?

Then we can solve SIS on all NTRU-like lattices.

Security Sketch

Guess the secret key ?

i.e. finding $\mathbf{S} = (\mathbf{f}, \mathbf{g})$ from $\mathbf{P} = (\mathbf{1}, \mathbf{h})$
Then you have an algorithm with unexpectedly low Hermite factor... Lucky you !

Find back the leaky part ?

Unlikely due to the distribution we picked $\mathbf{y}$ from

Forge a signature ?

Then we can solve SIS on all NTRU-like lattices.

Security Sketch

Guess the secret key ?

i.e. finding $\mathbf{S} = (\mathbf{f}, \mathbf{g})$ from $\mathbf{P} = (\mathbf{1}, \mathbf{h})$
Then you have an algorithm with unexpectedly low Hermite factor... Lucky you !

Find back the leaky part ?

Unlikely due to the distribution we picked $\mathbf{y}$ from

Forge a signature ?

Then we can solve SIS on all NTRU-like lattices.

Security Sketch

Guess the secret key ?

i.e. finding $\mathbf{S} = (\mathbf{f}, \mathbf{g})$ from $\mathbf{P} = (\mathbf{1}, \mathbf{h})$
Then you have an algorithm with unexpectedly low Hermite factor... Lucky you !

Find back the leaky part ?

Unlikely due to the distribution we picked $\mathbf{y}$ from

Forge a signature ?

Then we can solve SIS on all NTRU-like lattices.

Security Sketch

Guess the secret key ?

i.e. finding $\mathbf{S} = (\mathbf{f}, \mathbf{g})$ from $\mathbf{P} = (\mathbf{1}, \mathbf{h})$
Then you have an algorithm with unexpectedly low Hermite factor... Lucky you !

Find back the leaky part ?

Unlikely due to the distribution we picked $\mathbf{y}$ from

Forge a signature ?

Then we can solve SIS on all NTRU-like lattices.

Outline

security	100-size	100-speed	128-size	128-speed
M (retries)	7.492	2.728	7.465	2.725
$\approx$ sign size	10,700	12,700	14,500	17,100
$\approx$ *pk* size	6,900	6900	8,700	8,700
$\approx$ *sk* size	1,400	1,400	1,750	1,750

100 bits of security	[Lyu12]	Our scheme
$\approx$ sign size	14,500	10,700
$\approx$ *pk* size	$2^{19.5}$	6,900
$\approx$ *sk* size	$2^{22.1}$	1,400

security	100-size	100-speed	128-size	128-speed
M (retries)	7.492	2.728	7.465	2.725
$\approx$ sign size	10,700	12,700	14,500	17,100
$\approx$ *pk* size	6,900	6900	8,700	8,700
$\approx$ *sk* size	1,400	1,400	1,750	1,750

100 bits of security	[Lyu12]	Our scheme
$\approx$ sign size	14,500	10,700
$\approx$ *pk* size	$2^{19.5}$	6,900
$\approx$ *sk* size	$2^{22.1}$	1,400

security	100-size	100-speed	128-size	128-speed
M (retries)	7.492	2.728	7.465	2.725
$\approx$ sign size	10,700	12,700	14,500	17,100
$\approx$ *pk* size	6,900	6900	8,700	8,700
$\approx$ *sk* size	1,400	1,400	1,750	1,750

100 bits of security	[Ring-Lyu12]	Our scheme
$\approx$ sign size	14,500	10,700
$\approx$ *pk* size	1,500	6,900
$\approx$ *sk* size	8,800	1,400

Conclusion and Perspectives

What we did

- NTRUSign EU-CMA under SIS hardness
- extends to GGH
- does a little better than [Lyu12]

Left to do

- Generalize to LWE
- Use improved rejection sampling [DDLL13]
- Benchmark and compare to GPV techniques [SS13]

Conclusion and Perspectives

What we did

- NTRUSign EU-CMA under SIS hardness
- extends to GGH
- does a little better than [Lyu12]

Left to do

- Generalize to LWE
- Use improved rejection sampling [DDLL13]
- Benchmark and compare to GPV techniques [SS13]

Conclusion and Perspectives

What we did

- NTRUSign EU-CMA under SIS hardness
- extends to GGH
- does a little better than [Lyu12]

Left to do

- Generalize to LWE
- Use improved rejection sampling [DDLL13]
- Benchmark and compare to GPV techniques [SS13]

Conclusion and Perspectives

What we did

- NTRUSign EU-CMA under SIS hardness
- extends to GGH
- does a little better than [Lyu12]

Left to do

- Generalize to LWE
- Use improved rejection sampling [DDLL13]
- Benchmark and compare to GPV techniques [SS13]

Conclusion and Perspectives

What we did

- NTRUSign EU-CMA under SIS hardness
- extends to GGH
- does a little better than [Lyu12]

Left to do

- Generalize to LWE
- Use improved rejection sampling [DDLL13]
- Benchmark and compare to GPV techniques [SS13]

Conclusion and Perspectives

What we did

- NTRUSign EU-CMA under SIS hardness
- extends to GGH
- does a little better than [Lyu12]

Left to do

- Generalize to LWE
- Use improved rejection sampling [DDLL13]
- Benchmark and compare to GPV techniques [SS13]

Conclusion and Perspectives

What we did

- NTRUSign EU-CMA under SIS hardness
- extends to GGH
- does a little better than [Lyu12]

Left to do

- Generalize to LWE
- Use improved rejection sampling [DDLL13]
- Benchmark and compare to GPV techniques [SS13]

Conclusion and Perspectives

What we did

- NTRUSign EU-CMA under SIS hardness
- extends to GGH
- does a little better than [Lyu12]

Left to do

- Generalize to LWE
- Use improved rejection sampling [DDLL13]
- Benchmark and compare to GPV techniques [SS13]

Thank you !

[CN11] Chen, Y., Nguyen, P. Q. *BKZ 2.0: Better Lattice Security Estimates*. AsiaCrypt'11

[DDLL13] Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V. *Lattice Signatures and Bimodal Gaussians*. Crypto'13

[DN12] Ducas, L., Nguyen, Phong Q. *Learning a Zonotope and More: Cryptanalysis of NTRUSign Countermeasures*. AsiaCrypt'12

[HPS01] Hoffstein, J., Pipher, J., Silverman, J. H. *NSS: An NTRU Lattice-Based Signature Scheme*. EuroCrypt'01

[HHPSW03] Hoffstein, J., Howgrave-graham, N., Pipher, J., Silverman, J.H., Whyte, W. *NTRUSign: Digital Signatures Using the NTRU Lattice*. CT-RSA'03

[LP11] Lindner, R., Peikert, C. *Better Key Sizes (and Attacks) for LWE-Based Encryption*. CT-RSA'11

[Lyu12] Lyubashevsky, V. *Lattice Signatures Without Trapdoors*. EuroCrypt'12

[NR06] Nguyen, Phong Q., Regev, O. *Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures*. EuroCrypt'06