RSA CONFERENCE 2013

Security in knowledge

Proxy Re-Encryption in a Stronger Security Model Extended from CT-RSA2012

Manh Ha Nguyen

Tokyo Institute of Technology

Toshiyuki Isshiki[1,2] and Keisuke Tanaka[2]

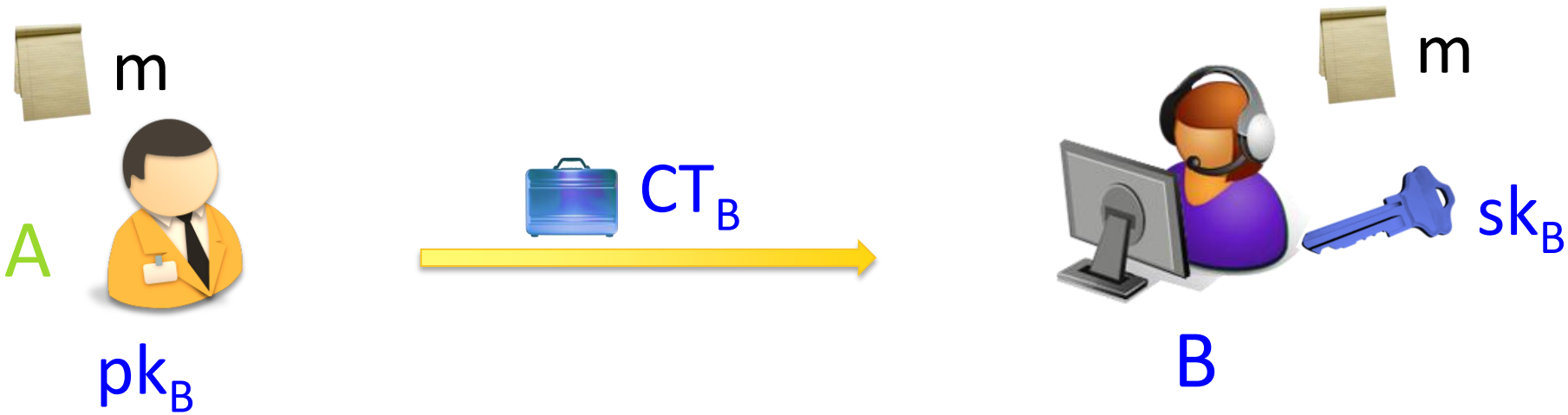
[1]NEC Corporation

[2]Tokyo Institute of Technology

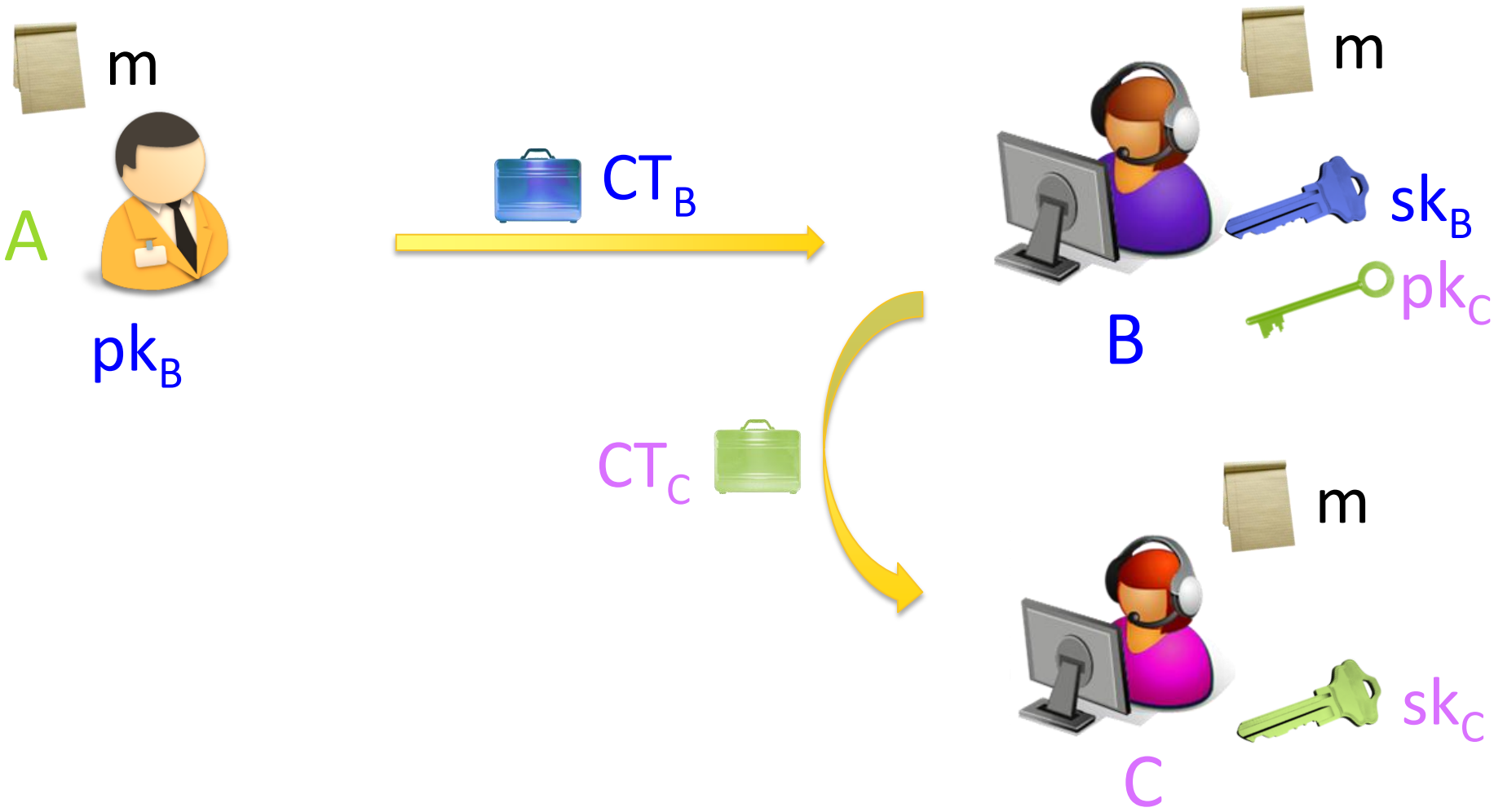
Session ID: CRYP-R35

Session Classification: Advanced

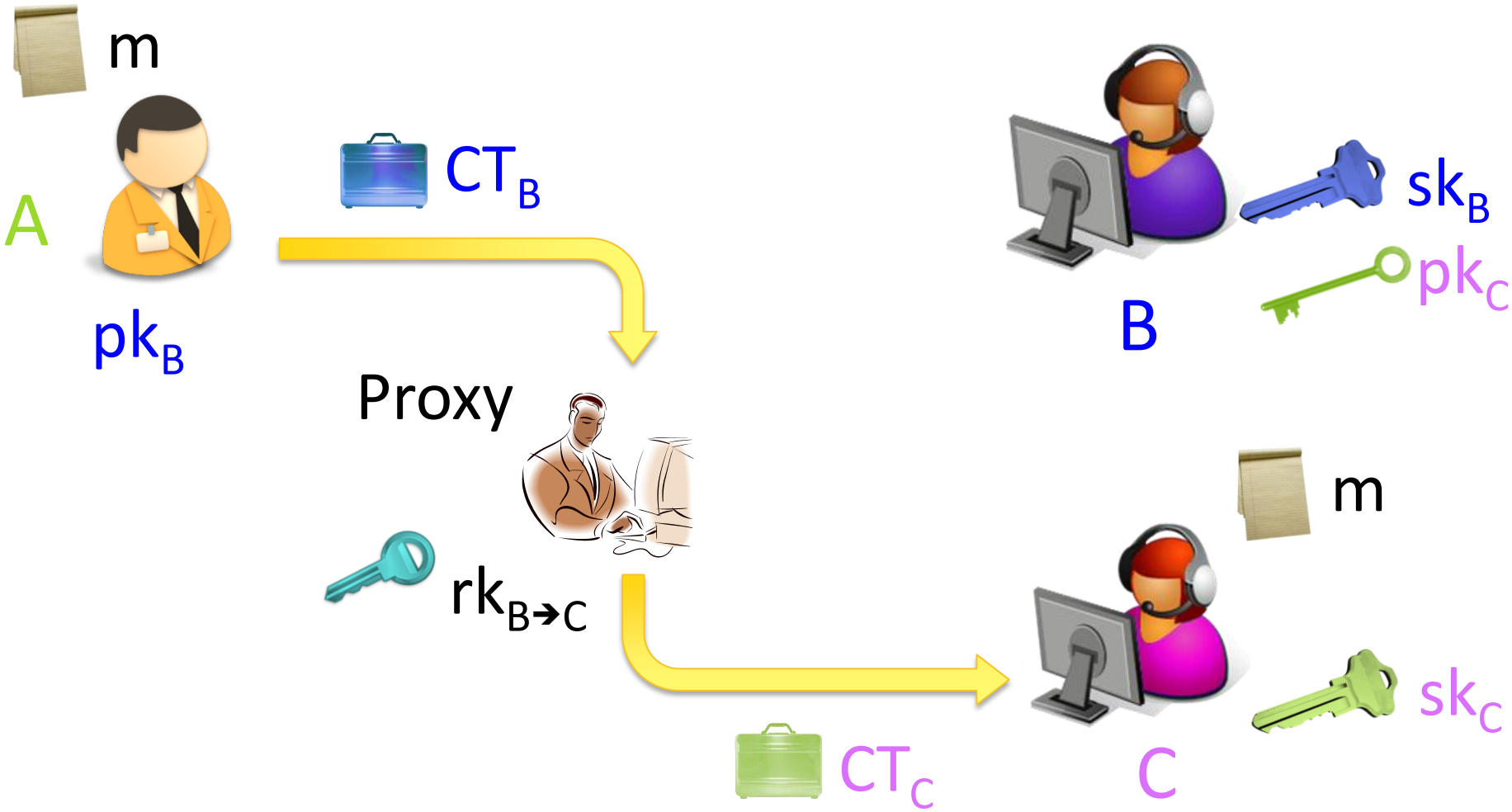
Public Key Encryption



Email Forwarding



Proxy Re-Encryption (PRE)



Proxy Re-Encryption (PRE)

Types

- Bidirectional or Unidirectional

- Multi-hop or Single-hop

This work: Unidirectional, Single-hop PRE

Our Results

- We present a definition of security against chosen-ciphertext attacks (CCA) for unidirectional, single-hop PRE
  - Extended from previous works
- New PRE scheme with this security

Agenda

- Model of PRE
- Security Notions of PRE
- Proposed PRE Scheme
- Concluding Remarks

Model of PRE

Model of PRE

- Setup$(1^k) \Rightarrow PP$
- KGen$(PP) \Rightarrow (pk,sk)$
- ReKey$(PP,sk_i,pk_j) \Rightarrow rk_{i->j}$
- Enc$_1(PP,pk_i,m) \Rightarrow 1^{st}\text{-}CT_i$ (first level)
- Enc$_2(PP,pk_i,m) \Rightarrow 2^{nd}\text{-}CT_i$ (second level)
- ReEnc$(PP, rk_{i->j}, 2^{nd}\text{-}CT_i) \Rightarrow 1^{st}\text{-}CT_j$ (first level)
- Dec$_1(PP,pk_i,1^{st}\text{-}CT_i) \Rightarrow m$
- Dec$_2(PP,pk_i,2^{nd}\text{-}CT_i) \Rightarrow m$

Model of PRE

(second level) 2nd-CT_i $\xrightarrow{\text{ReEnc}}$ (first level) 1st-CT_j

(first level) 1st-CT_i $\xrightarrow{\text{ReEnc}}$ ✗

- $\mathrm{Enc}_2(PP, pk_i, m) \Rightarrow 2^{nd}\text{-}CT_i$ (second level)
- $\mathrm{ReEnc}(PP, rk_{i\text{->}j}, 2^{nd}\text{-}CT_i) \Rightarrow 1^{st}\text{-}CT_j$ (first level)
- $\mathrm{Dec}_1(PP, pk_i, 1^{st}\text{-}CT_i) \Rightarrow m$
- $\mathrm{Dec}_2(PP, pk_i, 2^{nd}\text{-}CT_i) \Rightarrow m$

Summary of uni-PRE Schemes

Schemes	IND-CCA Security of ReEnc	RO-free	Security of Second Level Ciphertext
[AFGH06]	✗	○	CPA
[LV08]	✗	○	RCCA
[CWYD10]	✗	✗	CCA
[HMY+11]	✗	○	RCCA
[CDL11]	○	✗	CCA
[HKK+12]	✗	○	CCA
Ours	○	○	CCA

Security Notions of PRE

Security Notions of PRE

Previous works

- Security of second level ciphertext
- Security of first level ciphertext

@CT-RSA2012

(E.g. [LV08], [CWYD10], [HKK+12], …)

Security Notions of PRE

<u>This work</u>

- Security of second level ciphertext
- Security of original first level ciphertext
- Security of re-encrypted ciphertext

Security Notions of PRE

<u>This work</u>

- Security of second level ciphertext

  (2nd-level-CCA security)

- Security of original first level ciphertext

  (1st-ori-CCA security)

- Security of re-encrypted ciphertext

  (1st-re-CCA security)

2nd-level-CCA Security

$\mathcal{O}_{pk}$, $\mathcal{O}_{sk}$, $\mathcal{O}_{rk}$, $\mathcal{O}_{re}$, $\mathcal{O}_{dec1}$, $\mathcal{O}_{dec2}$

With some conditions!!!

$\mathcal{O}_{racles}$

$pk_{i*}, (m_0, m_1)$

2nd-CT^*

$\mathcal{O}_{racles}$

Output b'

$b \in \{0,1\}$

2nd-$CT^* \leftarrow \mathrm{Enc}_2(pk_{i*}, m_b)$

2nd-level-CCA Security

Conditions:

- $\mathcal{O}_{\mathrm{sk}}(i^*)$ is not allowed

- $\mathcal{O}_{\mathrm{rk}}(i^*,j)$ is allowed if pk_j is uncorrupted key

- If *Adv* issues $\mathcal{O}_{\mathrm{re}}(i,j,CT_i)$ or $\mathcal{O}_{\mathrm{dec1}}(i,CT_i)$:

(pk_i, CT_i) can't be a derivative of $(pk_{i*}, 2^{nd}\text{-}CT^*)$

Extension from The Model of [HKK+12]

- Security of second level ciphertext
  - The gap w.r.t the restriction on the first level decryption queries

The example showing this gap in the proceedings is wrong, however, we can show the gap by other examples

- Security of first level ciphertext
  - …

1st-ori-CCA Security

[CDL11]

$\mathcal{O}_{pk}$, $\mathcal{O}_{sk}$, $\mathcal{O}_{rk}$, ~~$\mathcal{O}_{re}$~~, $\mathcal{O}_{dec1}$, $\mathcal{O}_{dec2}$

$\mathcal{O}racles$

$pk_{i*}, (m_0, m_1)$

1st-CT^*

$\mathcal{O}racles$

Output b'

$b \in \{0,1\}$

1st-$CT^* \leftarrow \mathrm{Enc}_1(pk_{i*}, m_b)$

1st-re-CCA Security

[CDL11]

$\mathcal{O}_{pk}$, $\mathcal{O}_{sk}$, $\mathcal{O}_{rk}$, ~~$\mathcal{O}_{re}$~~, $\mathcal{O}_{dec1}$, $\mathcal{O}_{dec2}$

$\mathcal{O}racles$

pk_i, pk_{i*} $\boxed{(sCT_0, sCT_1)}$

1st-CT^*

$sCT_0 \leftarrow \mathrm{Enc}_2(pk_i, m_0)$

$sCT_1 \leftarrow \mathrm{Enc}_2(pk_i, m_1)$

$\mathcal{O}racles$

$b \in \{0,1\}$

Output b'

1st-$CT^* \leftarrow \mathrm{ReEnc}_1(rk_{i->i*}, sCT_b)$

CCA-Security of
first level ciphertext

VS.

1st-ori-CCA Security

1st-re-CCA Security

CCA-Security of
first level ciphertext

$1^{st}\text{-}CT^{*} = \mathrm{Enc}_1(pk_{i^*}, m_b)$ or
$\mathrm{ReEnc}(rk_{i\text{->}i^*}, \mathrm{Enc}_2(pk_i, m_b))$

CCA-Security of
first level ciphertext

$$1^{st}\text{-}CT^{*}=\mathrm{Enc}_1(pk_{i^*},m_b) \text{ or } \mathrm{ReEnc}(rk_{i\text{->}i^*},\mathrm{Enc}_2(pk_i,m_b))$$

1st-ori-CCA
Security

1st-re-CCA
Security

CCA-Security of
first level ciphertext

CCA-Security of
first level ciphertext

$1^{st}\text{-}CT^* = \mathrm{Enc}_1(pk_{i^*}, m_b)$ or
$\mathrm{ReEnc}(rk_{i\text{->}i^*}, \mathrm{Enc}_2(pk_i, m_b))$

If *Adv both compromised*
delegator (sk_i) and proxy ($rk_{i\text{->}i^*}$)

then *Adv* may have obtained the original ciphertext
$2^{nd}\text{-}CT_b \leftarrow \mathrm{Enc}_2(pk_i, m_b)$ and use sk_i to decrypt trivially

CCA-Security of first level ciphertext ⇏ Secure

1st-re-CCA Security ⇒ Secure

If *Adv both compromised*
delegator (sk_i) and proxy ($rk_{i\text{->}i*}$)

then *Adv* may have obtained the <u>original ciphertext</u>
$2^{nd}\text{-}CT_b \leftarrow \mathrm{Enc}_2(pk_i, m_b)$ and use sk_i to decrypt trivially

CCA-Security of
first level ciphertext

1st-ori-CCA
Security

1st-re-CCA
Security

Proposed PRE Scheme

Introduction of Our Scheme

- Construction based on
  - IND-CCA secure PKE scheme from CT-RSA'10 (Lai, Deng, Liu, and Kou)
  - IND-CCA secure PKE scheme from PKC'07 (Kiltz)
- Properties
  - 2nd-level-CCA
  - 1st-ori-CCA
  - 1st-re-CCA
  - w/o Random Oracle
  - Using bilinear maps

The Proposed Scheme (1/2)

- Setup(1^λ): PP=(p,G,G_T,g,g_1,h,u,v,d,u_1,v_1,d_1,e,H, TCR, TCR').

- KGen: $pk = (g^x, g_1^{x^2}, g^y), sk = (x, y)$

- ReKey(sk_i,pk_j): $rk_{i \to j} = g_1^{x_j^2 / x_i}$

The Proposed Scheme (2/2)

- $\mathrm{Enc}_2(pk_i, m)$: $CT_i = (C_1, C_2, C_3, C_4, C_5), where$

$$C_1 = pk_{i1}^r, C_2 = h^r, C_3 = e(g, g_1)^r \times m, t = H(C_2, C_3),$$

$$C_4 = (u^t v^s d), C_5 = s.$$

- $\mathrm{ReEnc}(rk_{i\to j}, pk_i, CT_i)$:
  - Check the validity of CT_i
  - Compute $CT_j = (A, B, C), where$

$$C_6 = C_1^R, C_7 = pk_{i1}^R, C_8 = rk_{i\to j}^{1/R}, CT' = C_2 \| C_3 \| \ldots \| C_8$$

$$A = g^{r'}, t' = TCR(A), B = (pk_{i,3}^{t'} \cdot h)^{r'},$$

$$C \leftarrow SYM.Enc(H(pk_{i,3}^{r'}), CT')$$

Security of Our Scheme

Theorem 1

Our scheme meets the 2nd-level-CCA security, assuming the hash function H is target collision resistant, the 6-AmDBDH assumption holds in groups (G, G_T), *and the 2-AmCDH problem is hard.*

Theorem 2

Our scheme meets the 1st-ori-CCA and the 1st-re-CCA security, assuming the hash function TCR' is target collision resistant, the GHDH problem is hard, and **SYM** *is CCA-secure.*

Concluding Remarks

Concluding Remarks

- We have present a CCA security definition for unidirectional, single-hop PRE
  - Extended from previous works
- New PRE scheme with this security

Solving BDD by Enumeration: An Update

Mingjie Liu[1], Phong Q.Nguyen[2]

[1]BICMR, Peking University and Tsinghua University

[2]INRIA and Tsinghua University

CT-RSA 2013

Outline

Motivation

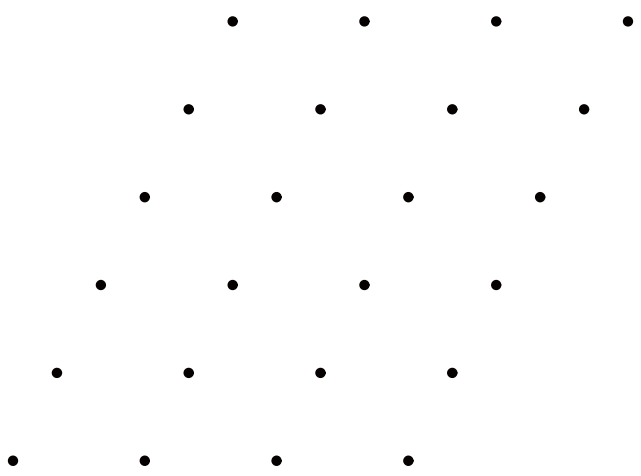
Algorithm for BDD
- BDD Problem
- Babai NearestPlane Algorithm
- Lindner-Peikert's NearestPlanes Algorithm Revisited

Solving BDD by (GNR) Pruned Enumeration
- Randomizing the NearestPlanes Algorithm
- GNR Pruned Enumeration to BDD

Conclusion

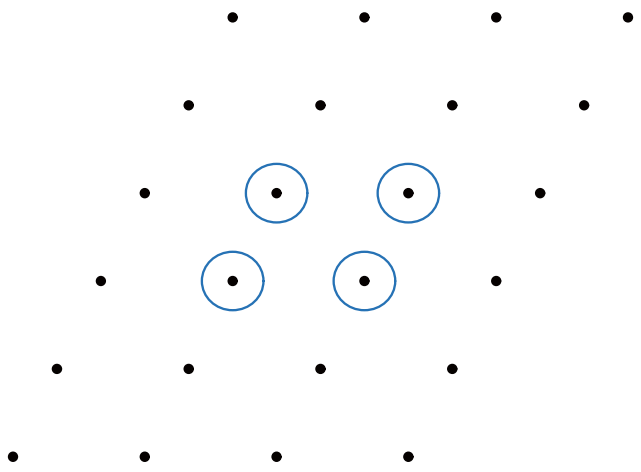
BDD: Why Is Important

- Lattice is a discrete additive subgroup of $\mathbb{R}^m$
- BDD: Bounded Distance Decoding, a basic lattice problem, a special case of CVP (closest vector problem), the target vector very close to lattice



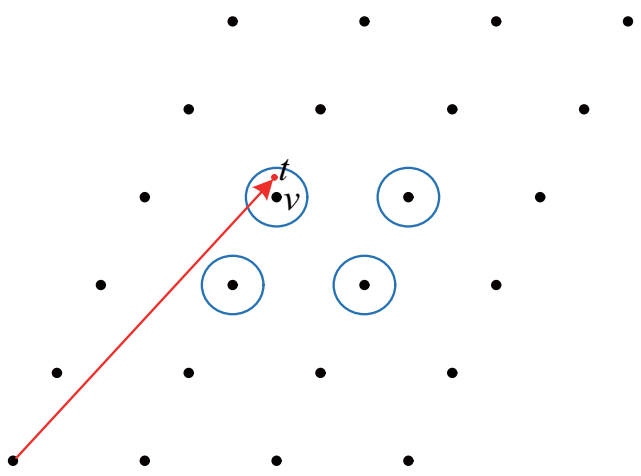
BDD: Why Is Important

- Lattice is a discrete additive subgroup of $\mathbb{R}^m$
- BDD: Bounded Distance Decoding, a basic lattice problem, a special case of CVP (closest vector problem), the target vector very close to lattice



BDD: Why Is Important

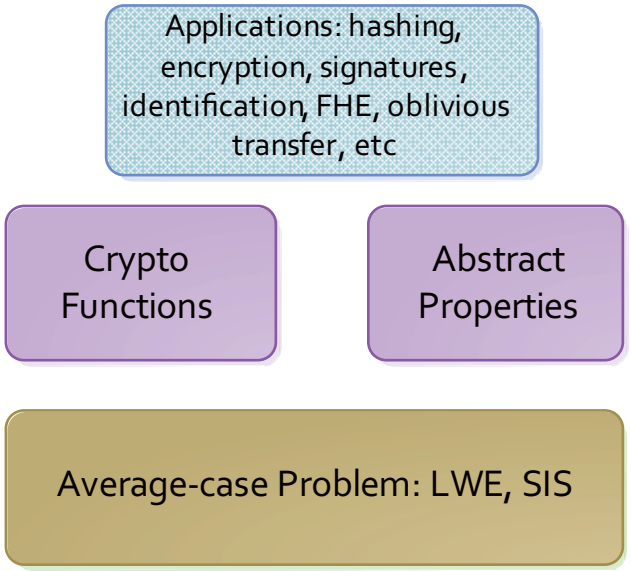
- Lattice is a discrete additive subgroup of $\mathbb{R}^m$
- BDD: Bounded Distance Decoding, a basic lattice problem, a special case of CVP (closest vector problem), the target vector very close to lattice



Lattice in Crypto

- Applications to cryptanalysis
  - Lattice based schemes:NTRU, GGH etc
  - Non-lattice based schemes: knapsack, RSA, DSA, etc

- Cryptography



BDD: Why Is Important

- The security of most lattice-based encryption schemes relies on the hardness of BDD, such as
  - LWE (Learning with error): one of the two average-case problems most of the provably-secure lattice-based constructions are based on
  - Hidden number problem(HNP)– DSA with partially known nonces, chosen ciphertext attack on RSA-PKCS#1 encryption version 1.5,

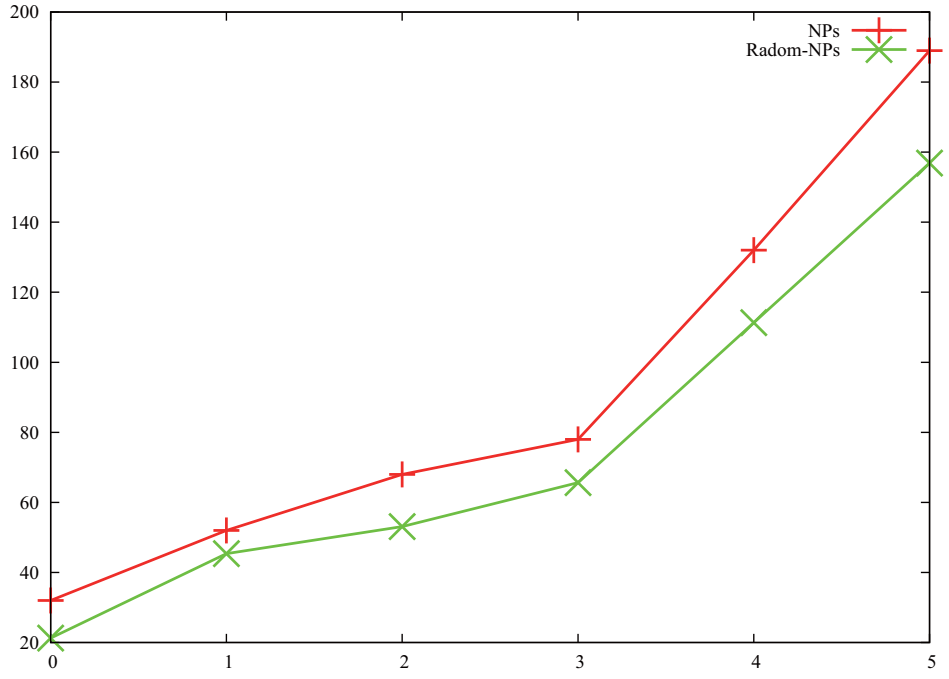
  What is the best algorithm for BDD ?

Previous Work

- Babai's NearestPlane algorithm, Babai, 1985
- NearestPlanes algorithm, Lindner and Peikert, 2011
  - Claimed that this was the best attack known on Search-LWE
- Embedding method, Kannan 1987
  - Heuristically reduces BDD to the unique-shortest vector problem
  - The largest BDD cryptanalytic instances ever solved in practice using this method

Our Work

- Rephrasing NearestPlanes algorithm in the pruned-enumeration framework of Gama-Nguyen-Regev (GNR)
  - Present a simple randomized variant.
  - In the case of LWE, significantly better than the Lindner-Peikert attack the speedup can be as big as 2^{32}



Our Work

- Consider GNR pruned-enumeration algorithms to solve BDD:
  - Provides even better attacks on Search-LWE
  - The method of choice in practice for the general BDD case.
    - Recover the DSA secret key from $2 \leftarrow 3$ least significant bits with 100 instances
    - Lattice version of Bleichenbacher's celebrated chosen ciphertext attack [Ble98] on RSA-PKCS#1.5 encryption version 1.5, 1024bit, 80 dimension (Babai algorithm)$\longrightarrow$ 65 dimesion (Enumeration with pruning)
    - GGH: the first partial secret-key recovery in dimensions 200-300 and re-solved the 350-dimensional message-recovery challenge using much weaker lattice reduction

Outline

Motivation

Algorithm for BDD
- BDD Problem
- Babai NearestPlane Algorithm
- Lindner-Peikert's NearestPlanes Algorithm Revisited

Solving BDD by (GNR) Pruned Enumeration
- Randomizing the NearestPlanes Algorithm
- GNR Pruned Enumeration to BDD

Conclusion

BDD Problem

- Bounded Distance Decoding (BDD): Given a lattice and a target vector unusually close to the lattice, to find the closest lattice vector to the target
- What is the best algorithm for solving BDD in practice?
- Several parameters impact the answer, *e.g.* the dimension, the size and shape of the error.

LWE problem

- Search: find $\mathbf{s} \in \mathbb{Z}_q^n$, given 'noisy random inner product'

$$\begin{aligned} a_1 \quad &, \quad b_1 = \langle a_1, s \rangle + e_1 \mod q \\ a_2 \quad &, \quad b_2 = \langle a_2, s \rangle + e_2 \mod q \\ &\vdots \end{aligned}$$

- Uniform $\mathbf{a}_i \in \mathbb{Z}_q^n$, Gaussian error $\mathbf{e}_i$
- LWE is a BDD instance for the lattice $\Lambda_q(A) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} = \mathbf{s}A \mod q \text{ for } \mathbf{s} \in \mathbb{Z}_q^n\}$

LWE problem

- Search: find $\mathbf{s} \in \mathbb{Z}_q^n$, given 'noisy random inner product'

$$\begin{array}{ccc} a_1 & , & b_1 \\ a_2 & , & b_2 \\ a_3 & , & b_3 \end{array}$$

- Uniform $\mathbf{a}_i \in \mathbb{Z}_q^n$, Gaussian error $\mathbf{e}_i$
- LWE is a BDD instance for the lattice $\Lambda_q(A) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} = \mathbf{s}A \mod q \text{ for } \mathbf{s} \in \mathbb{Z}_q^n\}$

LWE problem

- Search: find $\mathbf{s} \in \mathbb{Z}_q^n$, given 'noisy random inner product'

$$m\left\{\begin{pmatrix} \vdots \\ A^t \\ \vdots \end{pmatrix}, \begin{pmatrix} \vdots \\ b \\ \vdots \end{pmatrix}\right. = A^t s + e$$

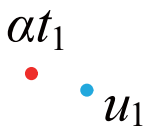
- Uniform $\mathbf{a}_i \in \mathbb{Z}_q^n$, Gaussian error $\mathbf{e}_i$
- LWE is a BDD instance for the lattice $\Lambda_q(A) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} = \mathbf{s}A \mod q \text{ for } \mathbf{s} \in \mathbb{Z}_q^n\}$

HNP Problem: Hidden Number Problem

- Recover $\alpha \in \mathbb{Z}_q$, given $u_i = APP_{\ell,q}(\alpha t_i)$, $1 \leq i \leq d$ where $APP_{\ell,q}(n)$ denotes any rational number r satisfying $|n - r|_q \leq \frac{q}{2^{\ell+1}}$, $|z|_q = \min_{b \in \mathbb{Z}} |z - bq|$
- Example: DSA with partially known nonces, chosen ciphertext attack on RSA-PKCS#1 encryption version 1.5

HNP Problem: Hidden Number Problem

- Recover $\alpha \in \mathbb{Z}_q$, given $u_i = APP_{\ell,q}(\alpha t_i)$, $1 \leq i \leq d$ where $APP_{\ell,q}(n)$ denotes any rational number r satisfying $|n - r|_q \leq \frac{q}{2^{\ell+1}}$, $|z|_q = \min_{b \in \mathbb{Z}} |z - bq|$
- Example: DSA with partially known nonces, chosen ciphertext attack on RSA-PKCS#1 encryption version 1.5



HNP Problem: Hidden Number Problem

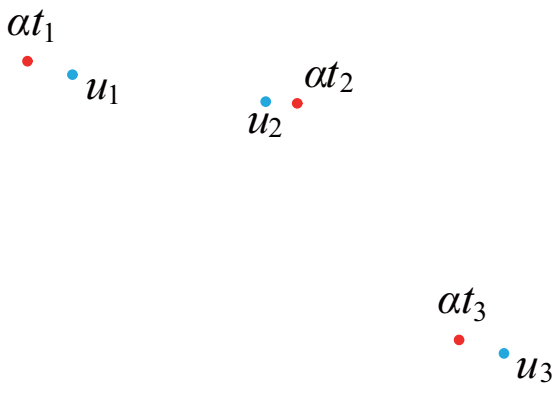
- Recover $\alpha \in \mathbb{Z}_q$, given $u_i = APP_{\ell,q}(\alpha t_i)$, $1 \leq i \leq d$ where $APP_{\ell,q}(n)$ denotes any rational number r satisfying $|n - r|_q \leq \frac{q}{2^{\ell+1}}$, $|z|_q = \min_{b \in \mathbb{Z}} |z - bq|$
- Example: DSA with partially known nonces, chosen ciphertext attack on RSA-PKCS#1 encryption version 1.5

HNP Problem: Hidden Number Problem

- Recover $\alpha \in \mathbb{Z}_q$, given $u_i = APP_{\ell,q}(\alpha t_i)$, $1 \leq i \leq d$ where $APP_{\ell,q}(n)$ denotes any rational number r satisfying $|n - r|_q \leq \frac{q}{2^{\ell+1}}$, $|z|_q = \min_{b \in \mathbb{Z}} |z - bq|$
- Example: DSA with partially known nonces, chosen ciphertext attack on RSA-PKCS#1 encryption version 1.5

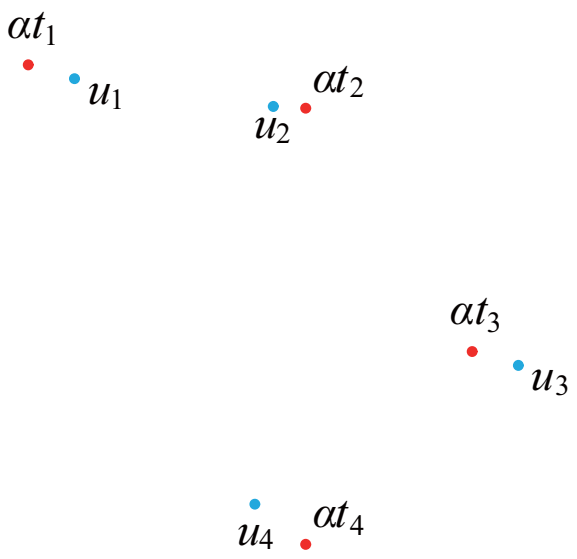
HNP Problem: Hidden Number Problem

- Recover $\alpha \in \mathbb{Z}_q$, given $u_i = APP_{\ell,q}(\alpha t_i)$, $1 \leq i \leq d$ where $APP_{\ell,q}(n)$ denotes any rational number r satisfying $|n - r|_q \leq \frac{q}{2^{\ell+1}}$, $|z|_q = \min_{b \in \mathbb{Z}} |z - bq|$
- Example: DSA with partially known nonces, chosen ciphertext attack on RSA-PKCS#1 encryption version 1.5



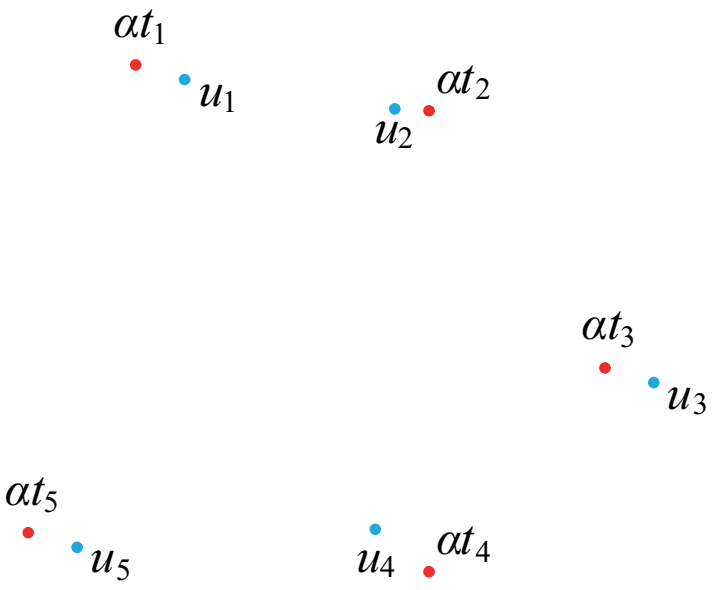
HNP Problem: Hidden Number Problem

- Recover $\alpha \in \mathbb{Z}_q$, given $u_i = APP_{\ell,q}(\alpha t_i)$, $1 \leq i \leq d$ where $APP_{\ell,q}(n)$ denotes any rational number r satisfying $|n - r|_q \leq \frac{q}{2^{\ell+1}}$, $|z|_q = \min_{b \in \mathbb{Z}} |z - bq|$
- Example: DSA with partially known nonces, chosen ciphertext attack on RSA-PKCS#1 encryption version 1.5



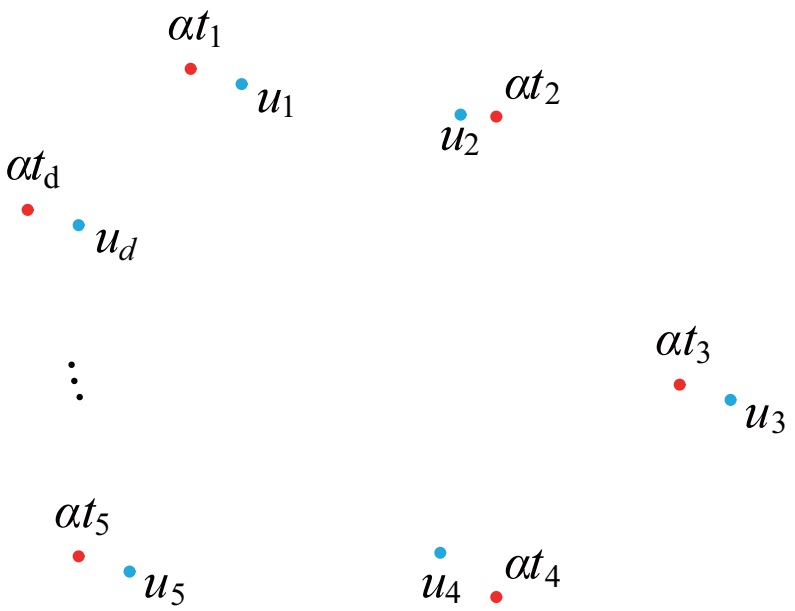
HNP Problem: Hidden Number Problem

- Recover $\alpha \in \mathbb{Z}_q$, given $u_i = APP_{\ell,q}(\alpha t_i)$, $1 \leq i \leq d$ where $APP_{\ell,q}(n)$ denotes any rational number r satisfying $|n - r|_q \leq \frac{q}{2^{\ell+1}}$, $|z|_q = \min_{b \in \mathbb{Z}} |z - bq|$
- Example: DSA with partially known nonces, chosen ciphertext attack on RSA-PKCS#1 encryption version 1.5



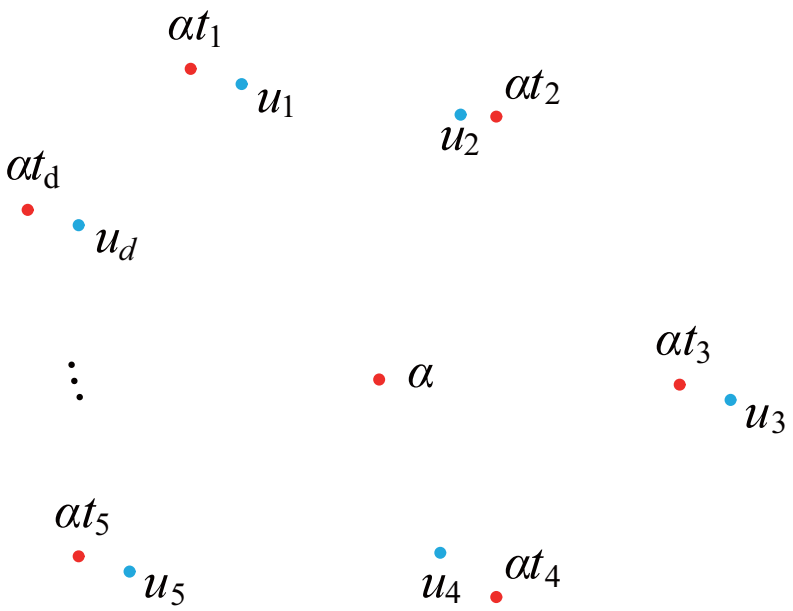
HNP Problem: Hidden Number Problem

- Recover $\alpha \in \mathbb{Z}_q$, given $u_i = APP_{\ell,q}(\alpha t_i)$, $1 \leq i \leq d$ where $APP_{\ell,q}(n)$ denotes any rational number r satisfying $|n - r|_q \leq \frac{q}{2^{\ell+1}}$, $|z|_q = \min_{b \in \mathbb{Z}} |z - bq|$
- Example: DSA with partially known nonces, chosen ciphertext attack on RSA-PKCS#1 encryption version 1.5



HNP Problem: Hidden Number Problem

- Recover $\alpha \in \mathbb{Z}_q$, given $u_i = APP_{\ell,q}(\alpha t_i)$, $1 \leq i \leq d$ where $APP_{\ell,q}(n)$ denotes any rational number r satisfying $|n - r|_q \leq \frac{q}{2^{\ell+1}}$, $|z|_q = \min_{b \in \mathbb{Z}} |z - bq|$
- Example: DSA with partially known nonces, chosen ciphertext attack on RSA-PKCS#1 encryption version 1.5



HNP Problem

- Can be reduced to :BDD problem in a $d+1$-dimensional lattice， target vector： $\mathbf{u} = (u_1, u_2, \ldots, u_d, 0)$

$$\begin{pmatrix} q & 0 & \cdots & 0 & 0 \\ 0 & q & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & q & 0 \\ t_1 & \cdots & \cdots & t_d & \frac{1}{2^{l+1}} \end{pmatrix} \tag{1}$$

- There exists a lattice vector $\mathbf{h} = (\alpha t_1 + qh_1, \ldots, \alpha t_d + qh_d, \frac{\alpha}{2^{\ell+1}})$, such that $\| \mathbf{h} - \mathbf{u} \| \leq \sqrt{d+1}\frac{q}{2^{\ell+1}}$

Outline

Motivation

Algorithm for BDD
- BDD Problem
- Babai NearestPlane Algorithm
- Lindner-Peikert's NearestPlanes Algorithm Revisited

Solving BDD by (GNR) Pruned Enumeration
- Randomizing the NearestPlanes Algorithm
- GNR Pruned Enumeration to BDD

Conclusion

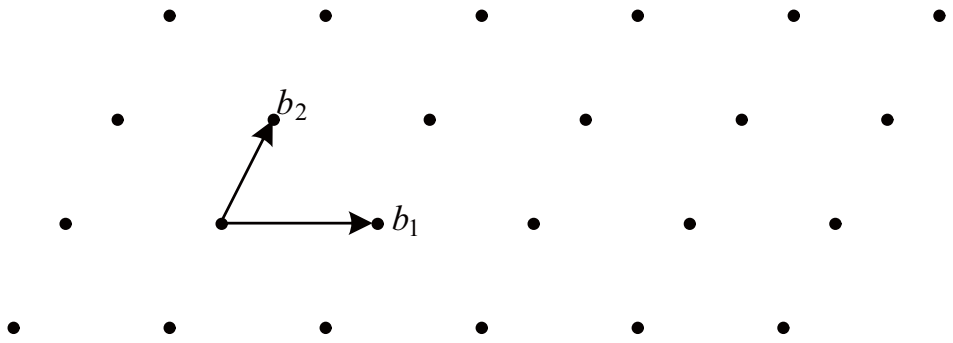
Babai's NearestPlane Algorithm

- Gram-schmidt orthogonalization: Give n linearly independent vetors, GS constructs n orthogonal vectors span the same subspaces. $\mathbf{b}_i^* = \pi_i(\mathbf{b}_i)$, where π_i: orthogonal projection over span $(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_{i-1})^\perp$

$$
\begin{array}{c}
\left(\begin{array}{ccccc}
\|b_1^*\| & 0 & 0 & \cdots & 0 \\
\mu_{21}\|b_1^*\| & \|b_2^*\| & 0 & \cdots & 0 \\
\mu_{31}\|b_1^*\| & \mu_{32}\|b_2^*\| & \|b_{m-k+1}^*\| & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mu_{m1}\|b_1^*\| & \mu_{m2}\|b_2^*\| & \mu_{m\ m-k+1}\|b_{m-k+1}^*\| & \cdots & \|b_m^*\|
\end{array}\right) \\
\begin{array}{ccccc}
\pi_1(B) & \pi_2(B) & \pi_{m-k+1}(B) & & \pi_m(B)
\end{array}
\end{array}
$$

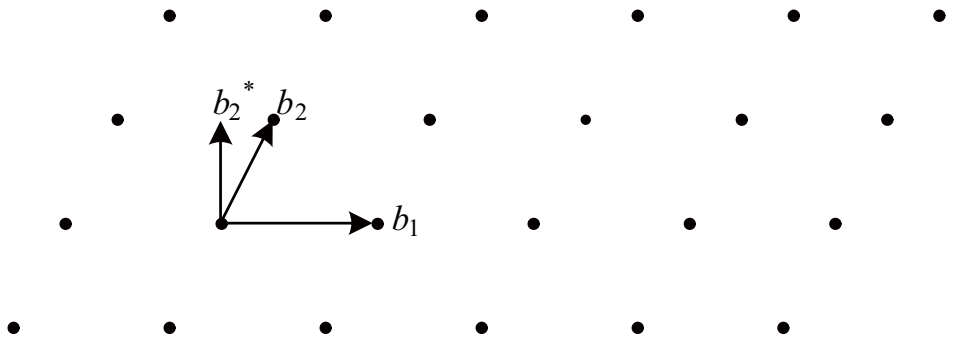
Babai's NearestPlane Algorithm

- Babai's algorithm outputs a lattice vector $\mathbf{v}$ relatively close to the input target vector $\mathbf{t}$. $\mathbf{v}$ is the unique lattice vector such that $\mathbf{v}-\mathbf{t} \in \mathcal{P}_{1/2}(B^*) = \{\sum_{i=1}^{m} x_i b_i^* : -\frac{1}{2} < x_i \leq \frac{1}{2}\} \Longleftrightarrow \|\xi_i(\mathbf{v}-\mathbf{t})\| \leq \|\mathbf{b}_i^*\|/2$ where $\xi_i(\mathbf{x})$ denotes the i-th coordinate of x in the normalized Gram-Schmidt basis $(\mathbf{b}_1^*/\|\mathbf{b}_1^*\|, \ldots, \mathbf{b}_m^*/\|\mathbf{b}_m^*\|)$.



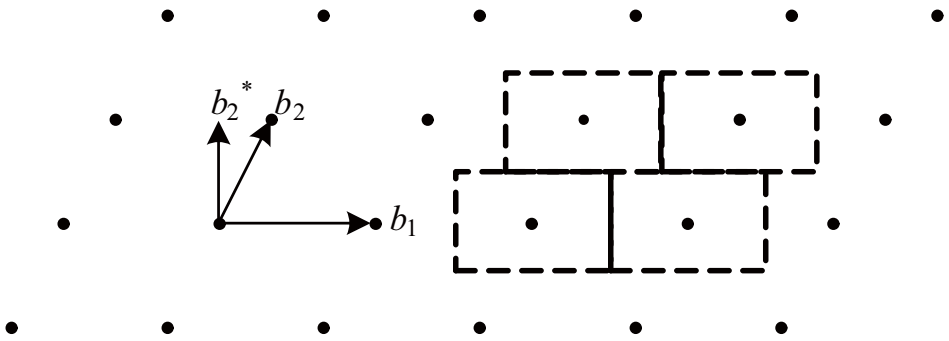
Babai's NearestPlane Algorithm

- Babai's algorithm outputs a lattice vector $\mathbf{v}$ relatively close to the input target vector $\mathbf{t}$. $\mathbf{v}$ is the unique lattice vector such that $\mathbf{v}-\mathbf{t} \in \mathcal{P}_{1/2}(B^*) = \{\sum_{i=1}^{m} x_i b_i^* : -\frac{1}{2} < x_i \leq \frac{1}{2}\} \Longleftrightarrow \|\xi_i(\mathbf{v}-\mathbf{t})\| \leq \|\mathbf{b}_i^*\|/2$ where $\xi_i(\mathbf{x})$ denotes the i-th coordinate of x in the normalized Gram-Schmidt basis $(\mathbf{b}_1^*/\|\mathbf{b}_1^*\|, \ldots, \mathbf{b}_m^*/\|\mathbf{b}_m^*\|)$.



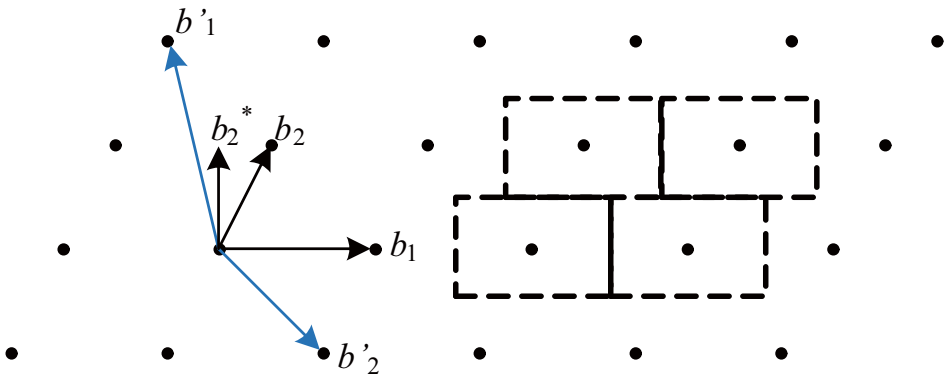
Babai's NearestPlane Algorithm

- Babai's algorithm outputs a lattice vector $\mathbf{v}$ relatively close to the input target vector $\mathbf{t}$. $\mathbf{v}$ is the unique lattice vector such that $\mathbf{v}-\mathbf{t} \in \mathcal{P}_{1/2}(B^*) = \{\sum_{i=1}^{m} x_i b_i^* : -\frac{1}{2} < x_i \leq \frac{1}{2}\} \Longleftrightarrow \|\xi_i(\mathbf{v}-\mathbf{t})\| \leq \|\mathbf{b}_i^*\|/2$ where $\xi_i(\mathbf{x})$ denotes the i-th coordinate of x in the normalized Gram-Schmidt basis $(\mathbf{b}_1^*/\|\mathbf{b}_1^*\|, \ldots, \mathbf{b}_m^*/\|\mathbf{b}_m^*\|)$.



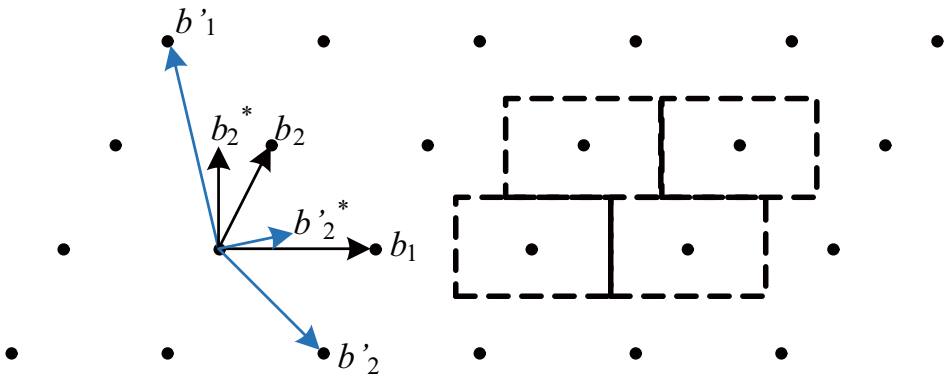
Babai's NearestPlane Algorithm

- Babai's algorithm outputs a lattice vector $\mathbf{v}$ relatively close to the input target vector $\mathbf{t}$. $\mathbf{v}$ is the unique lattice vector such that $\mathbf{v}-\mathbf{t} \in \mathcal{P}_{1/2}(B^*) = \{\sum_{i=1}^{m} x_i b_i^* : -\frac{1}{2} < x_i \leq \frac{1}{2}\} \Longleftrightarrow \|\xi_i(\mathbf{v}-\mathbf{t})\| \leq \|\mathbf{b}_i^*\|/2$ where $\xi_i(\mathbf{x})$ denotes the i-th coordinate of x in the normalized Gram-Schmidt basis $(\mathbf{b}_1^*/\|\mathbf{b}_1^*\|, \ldots, \mathbf{b}_m^*/\|\mathbf{b}_m^*\|)$.



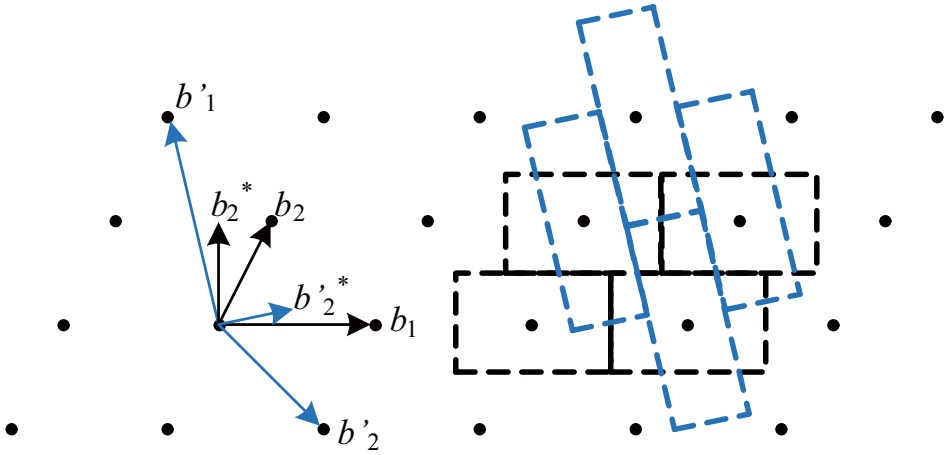
Babai's NearestPlane Algorithm

- Babai's algorithm outputs a lattice vector $\mathbf{v}$ relatively close to the input target vector $\mathbf{t}$. $\mathbf{v}$ is the unique lattice vector such that $\mathbf{v}-\mathbf{t} \in \mathcal{P}_{1/2}(B^*) = \{\sum_{i=1}^{m} x_i b_i^* : -\frac{1}{2} < x_i \leq \frac{1}{2}\} \Longleftrightarrow \|\xi_i(\mathbf{v}-\mathbf{t})\| \leq \|\mathbf{b}_i^*\|/2$ where $\xi_i(\mathbf{x})$ denotes the i-th coordinate of x in the normalized Gram-Schmidt basis $(\mathbf{b}_1^*/\|\mathbf{b}_1^*\|, \ldots, \mathbf{b}_m^*/\|\mathbf{b}_m^*\|)$.



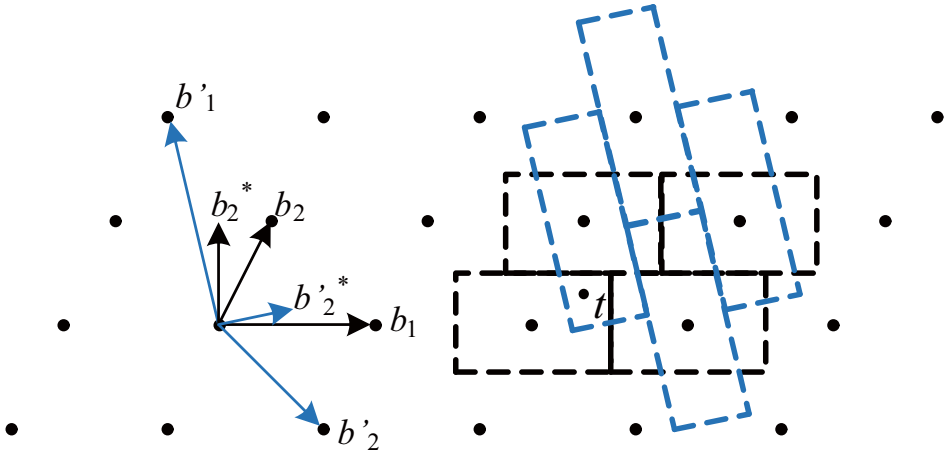
Babai's NearestPlane Algorithm

- Babai's algorithm outputs a lattice vector $\mathbf{v}$ relatively close to the input target vector $\mathbf{t}$. $\mathbf{v}$ is the unique lattice vector such that $\mathbf{v}-\mathbf{t} \in \mathcal{P}_{1/2}(B^*) = \{\sum_{i=1}^{m} x_i b_i^* : -\frac{1}{2} < x_i \leq \frac{1}{2}\} \Longleftrightarrow \|\xi_i(\mathbf{v}-\mathbf{t})\| \leq \|\mathbf{b}_i^*\|/2$ where $\xi_i(\mathbf{x})$ denotes the i-th coordinate of x in the normalized Gram-Schmidt basis $(\mathbf{b}_1^*/\|\mathbf{b}_1^*\|, \ldots, \mathbf{b}_m^*/\|\mathbf{b}_m^*\|)$.



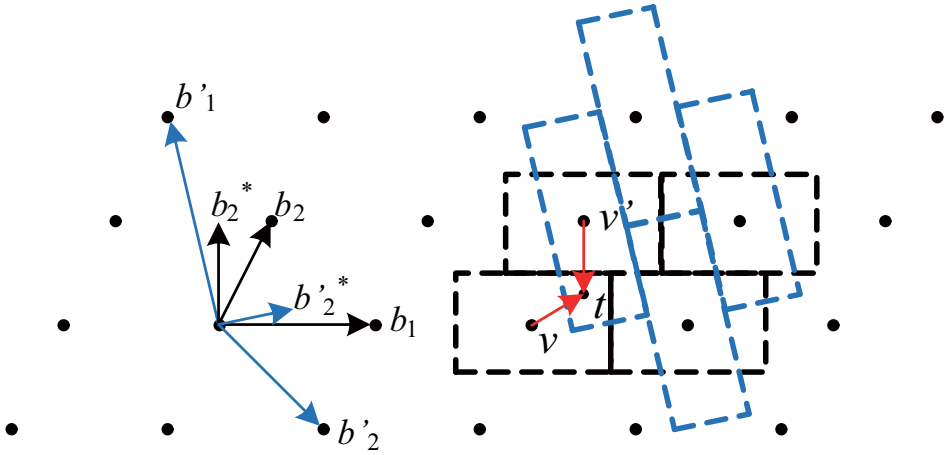
Babai's NearestPlane Algorithm

- Babai's algorithm outputs a lattice vector $\mathbf{v}$ relatively close to the input target vector $\mathbf{t}$. $\mathbf{v}$ is the unique lattice vector such that $\mathbf{v}-\mathbf{t} \in \mathcal{P}_{1/2}(B^*) = \{\sum_{i=1}^{m} x_i b_i^* : -\frac{1}{2} < x_i \leq \frac{1}{2}\} \Longleftrightarrow \|\xi_i(\mathbf{v}-\mathbf{t})\| \leq \|\mathbf{b}_i^*\|/2$ where $\xi_i(\mathbf{x})$ denotes the i-th coordinate of x in the normalized Gram-Schmidt basis $(\mathbf{b}_1^*/\|\mathbf{b}_1^*\|, \ldots, \mathbf{b}_m^*/\|\mathbf{b}_m^*\|)$.



Babai's NearestPlane Algorithm

- Babai's algorithm outputs a lattice vector $\mathbf{v}$ relatively close to the input target vector $\mathbf{t}$. $\mathbf{v}$ is the unique lattice vector such that $\mathbf{v}-\mathbf{t} \in \mathcal{P}_{1/2}(B^*) = \{\sum_{i=1}^{m} x_i b_i^* : -\frac{1}{2} < x_i \leq \frac{1}{2}\} \Longleftrightarrow \|\xi_i(\mathbf{v}-\mathbf{t})\| \leq \|\mathbf{b}_i^*\|/2$ where $\xi_i(\mathbf{x})$ denotes the i-th coordinate of x in the normalized Gram-Schmidt basis $(\mathbf{b}_1^*/\|\mathbf{b}_1^*\|, \ldots, \mathbf{b}_m^*/\|\mathbf{b}_m^*\|)$.

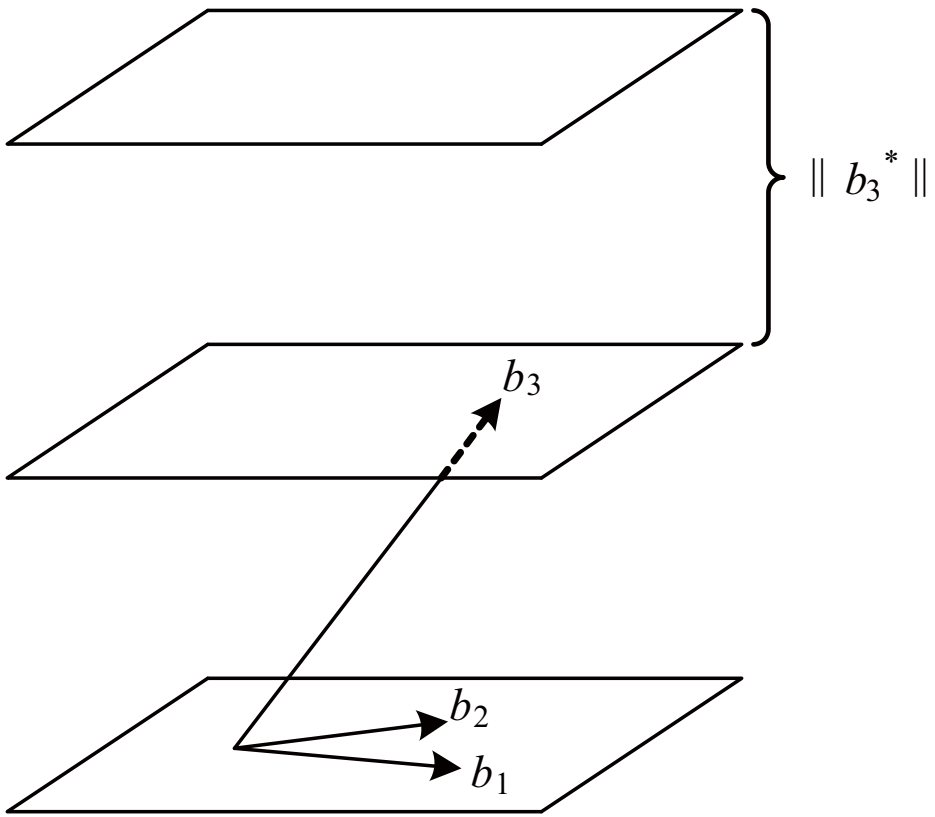


Babai's Nearest Plane Algorithm

- The algorithm success $\Longleftrightarrow$ error vector in $\mathcal{P}_{1/2}(B^*)$
- When the error distribution is Gaussian with standard deviation $s/\sqrt{2\pi}$, the success probability $\prod_{i=1}^{m} erf\left(\frac{\|\mathbf{b}_i^*\|\sqrt{\pi}}{2s}\right)$

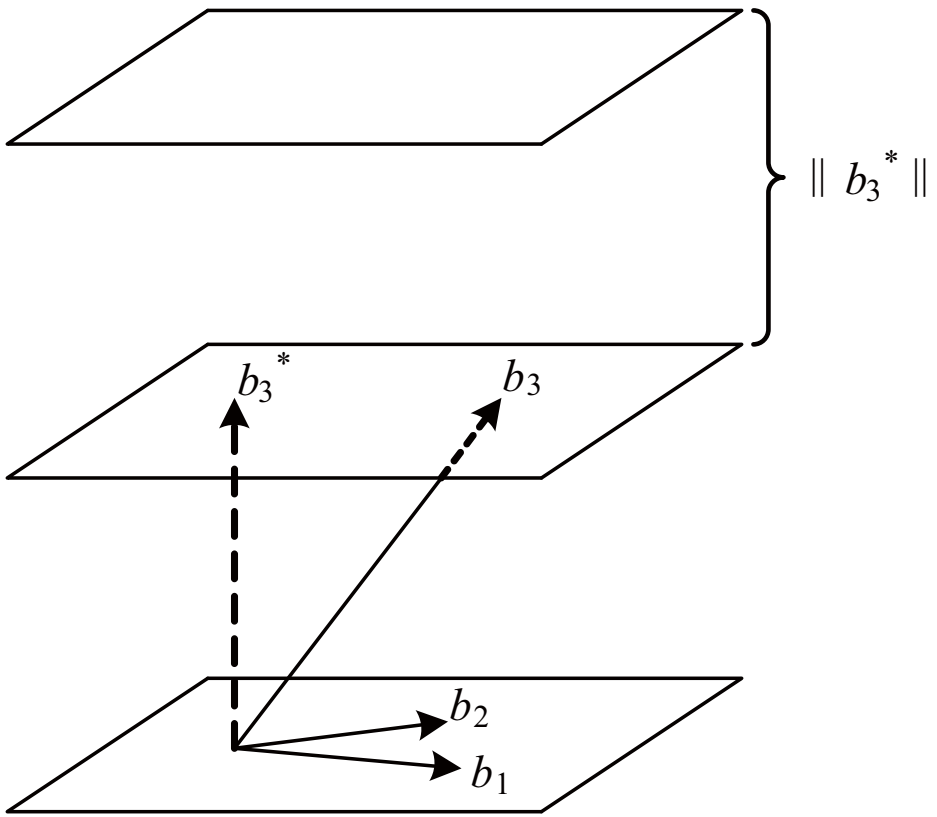
Babai's NearestPlane Algorithm

- Split the m-dimensional lattice as $m-1$-dimensional hyperplane $\Sigma_{i=1}^{m-1} x_i \mathbf{b}_i + k\mathbf{b}_m$, project the target vector to the nearest one. Then solve the $(m-1)$-dimensional case.



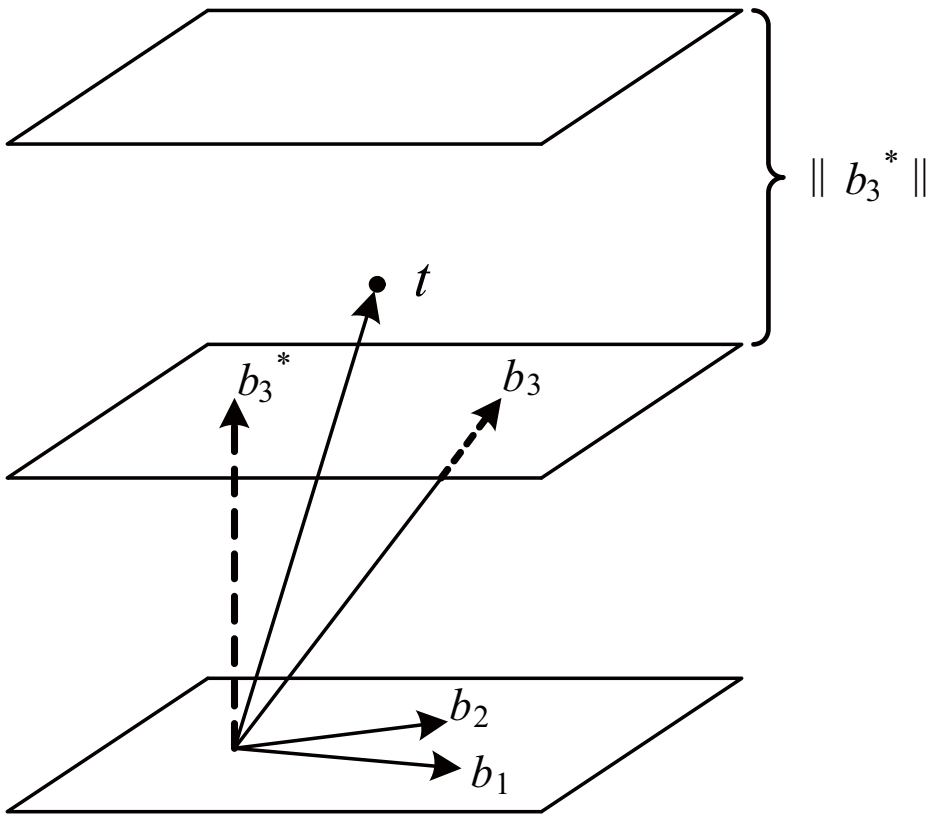
Babai's NearestPlane Algorithm

- Split the m-dimensional lattice as $m-1$-dimensional hyperplane $\Sigma_{i=1}^{m-1} x_i \mathbf{b}_i + k\mathbf{b}_m$, project the target vector to the nearest one. Then solve the $(m-1)$-dimensional case.



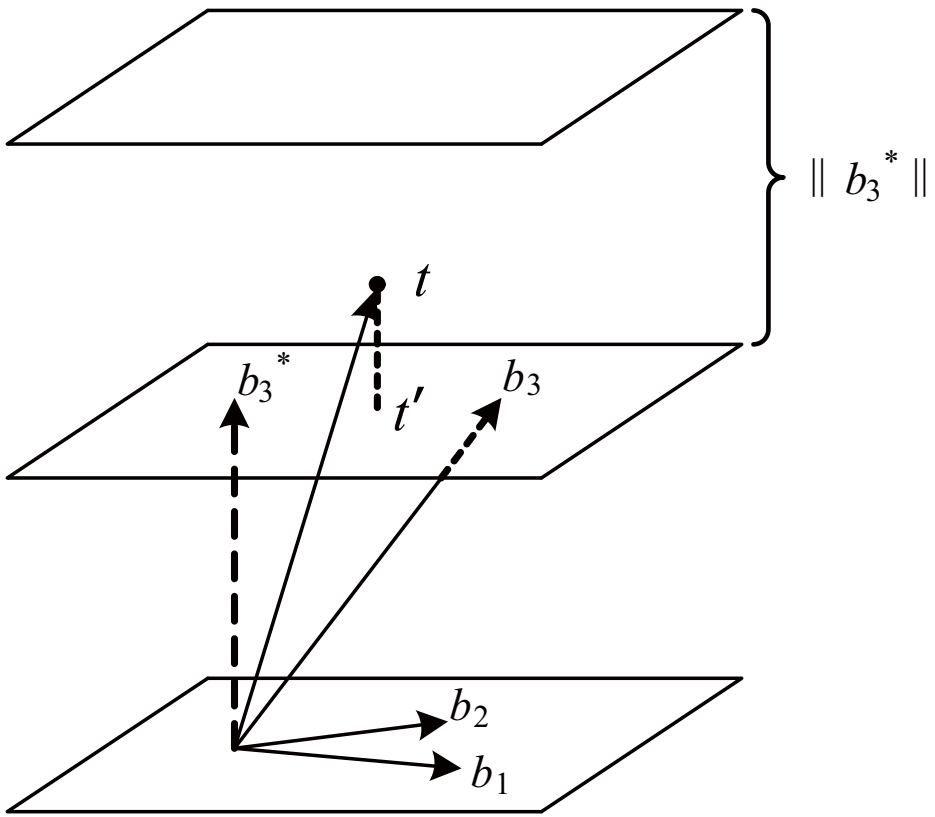
Babai's NearestPlane Algorithm

- Split the m-dimensional lattice as $m-1$-dimensional hyperplane $\Sigma_{i=1}^{m-1} x_i \mathbf{b}_i + k\mathbf{b}_m$, project the target vector to the nearest one. Then solve the $(m-1)$-dimensional case.



Babai's NearestPlane Algorithm

- Split the m-dimensional lattice as $m-1$-dimensional hyperplane $\Sigma_{i=1}^{m-1} x_i \mathbf{b}_i + k\mathbf{b}_m$, project the target vector to the nearest one. Then solve the $(m-1)$-dimensional case.



Outline

Motivation

Algorithm for BDD
- BDD Problem
- Babai NearestPlane Algorithm
- Lindner-Peikert's NearestPlanes Algorithm Revisited

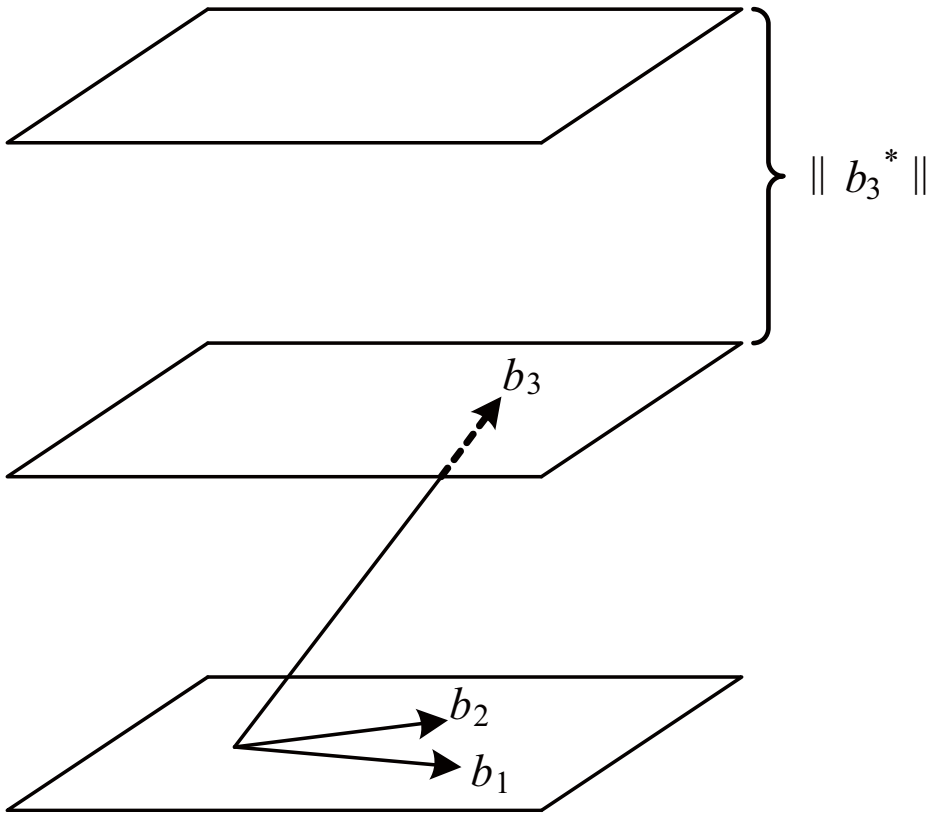
Solving BDD by (GNR) Pruned Enumeration
- Randomizing the NearestPlanes Algorithm
- GNR Pruned Enumeration to BDD

Conclusion

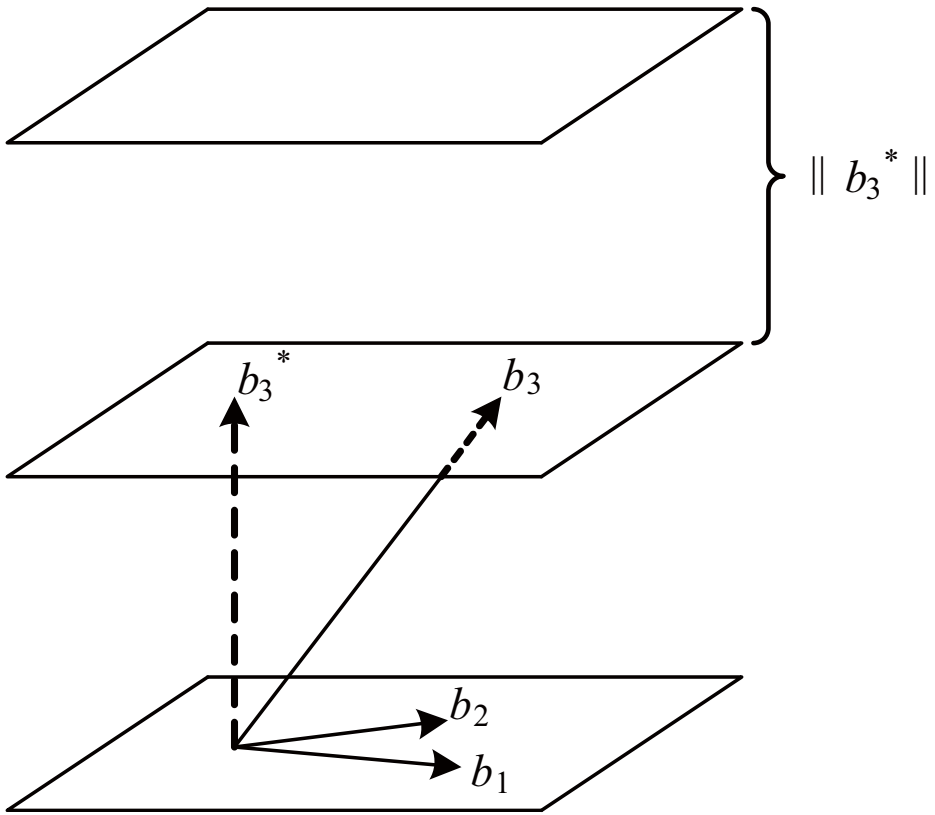
The NearestPlanes Algorithm

- Choose $d_i \geq 1$ distinct planes in the i-th level
- These generalizations make $\mathcal{P}_{1/2}(B^*)$ wider in the direction of $\mathbf{b}^*{}_i$ by a factor of d_i
- Output a set of Πd_i distinct lattice vectors in $\mathcal{L}(\mathbf{B})$, $\mathbf{v} - \mathbf{t} \in \mathcal{P}_{1/2}(DB^*)$ where D is diagonal matrix with $(d_1, d_2, ..., d_m)$. $\|\xi_i(\mathbf{v} - \mathbf{t})\| \leq \|d_i \mathbf{b}_i^*\|/2$
- The success probability is $\prod_{i=1}^{m} erf\left(\frac{d_i \|\mathbf{b}_i^*\| \sqrt{\pi}}{2s}\right)$.
- Lindner and Peikert[LP11]: their algorithm is better for most parameters and success probability than the distinguisher of Micciancio and Regev[MR08] for LWE.

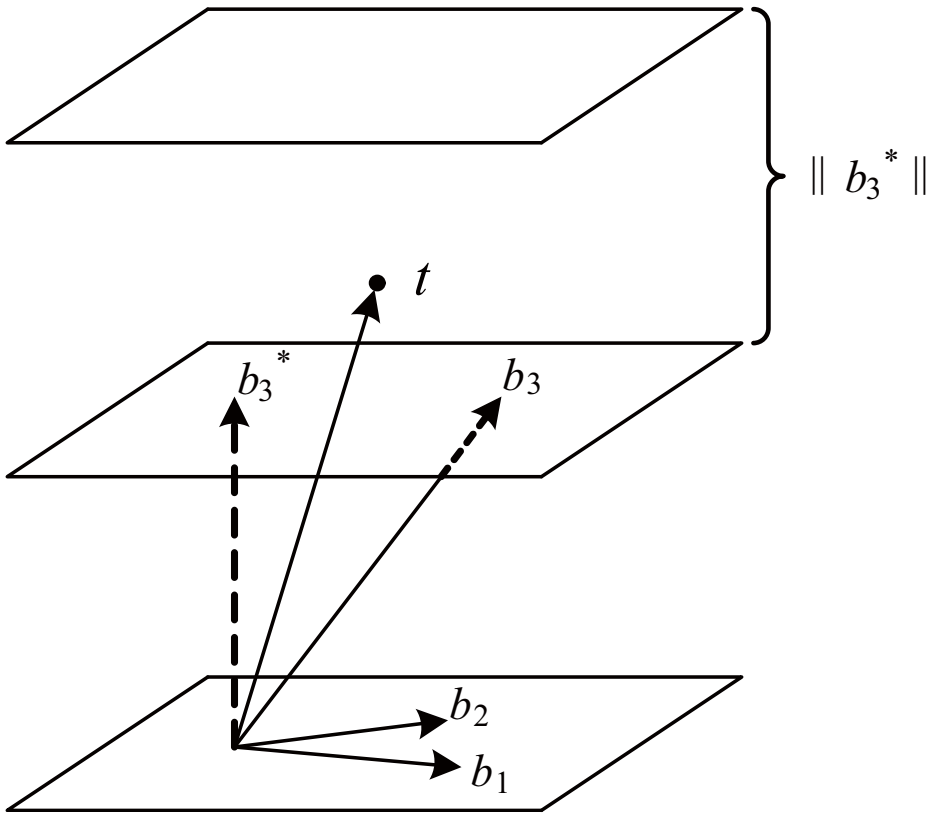
The NearestPlanes Algorithm



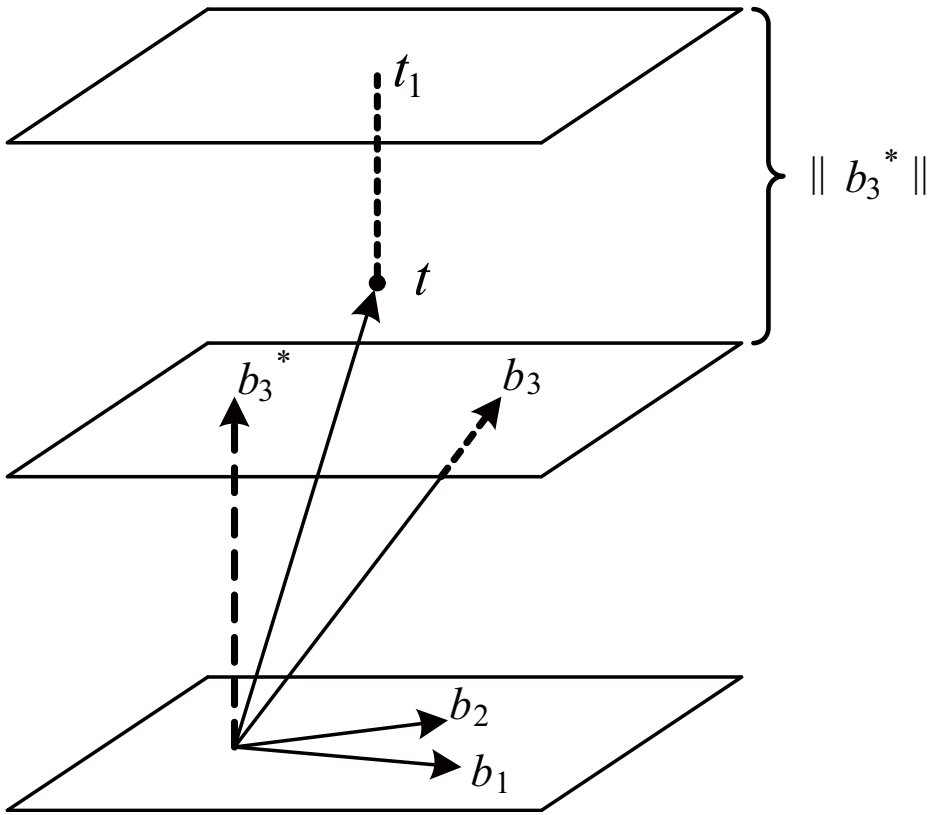
The NearestPlanes Algorithm



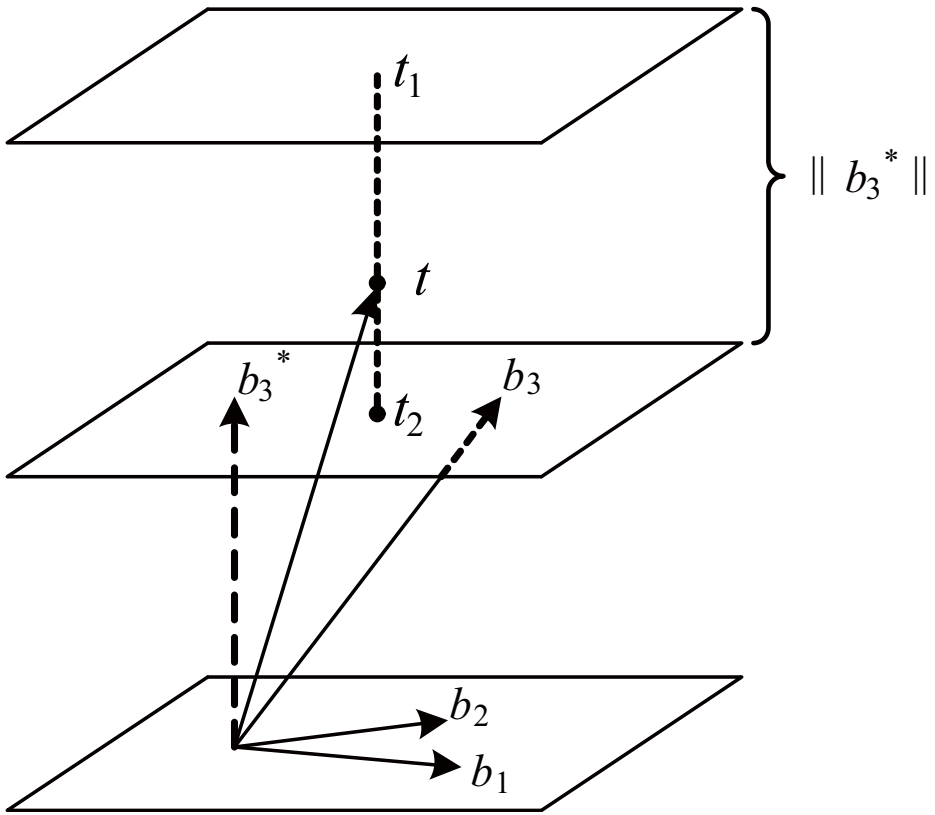
The NearestPlanes Algorithm



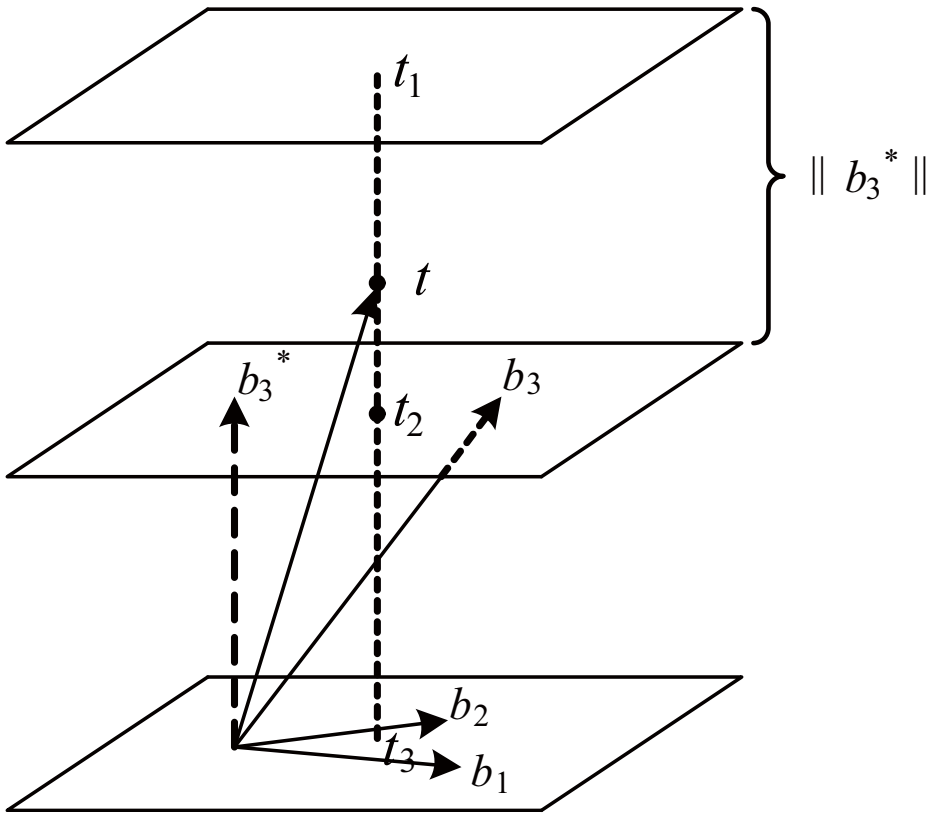
The NearestPlanes Algorithm



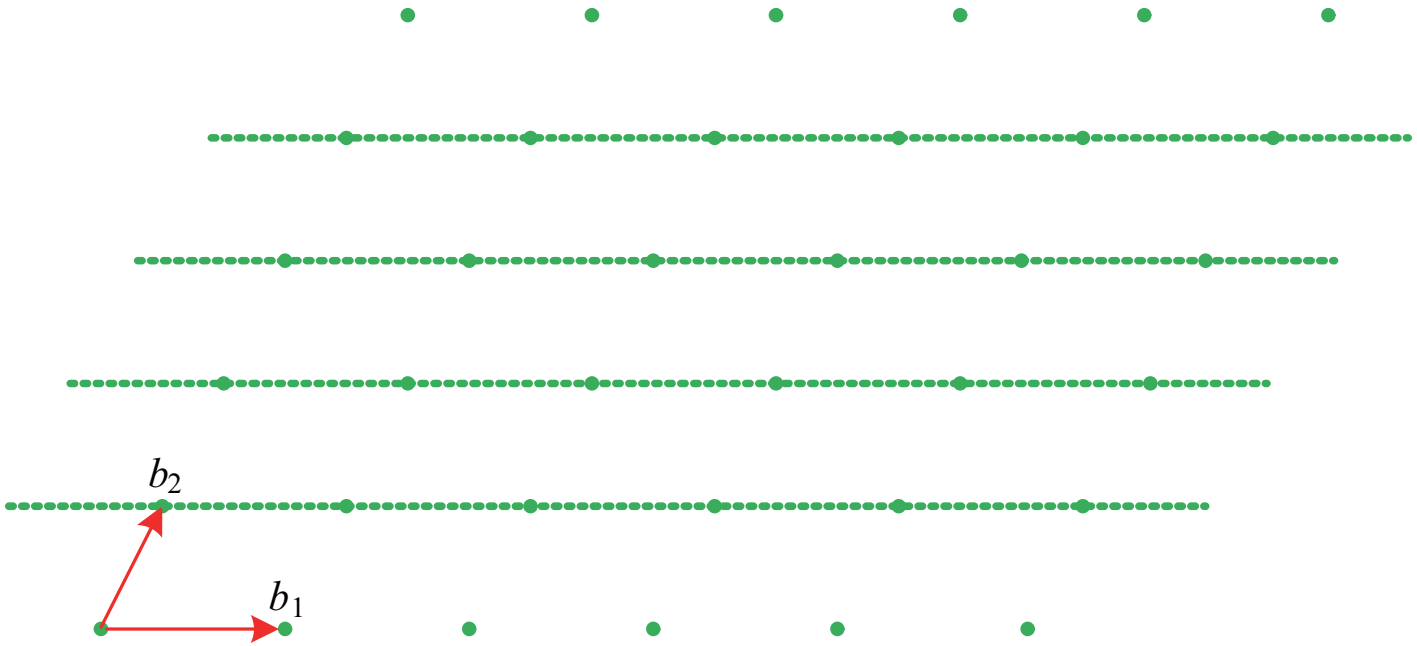
The NearestPlanes Algorithm



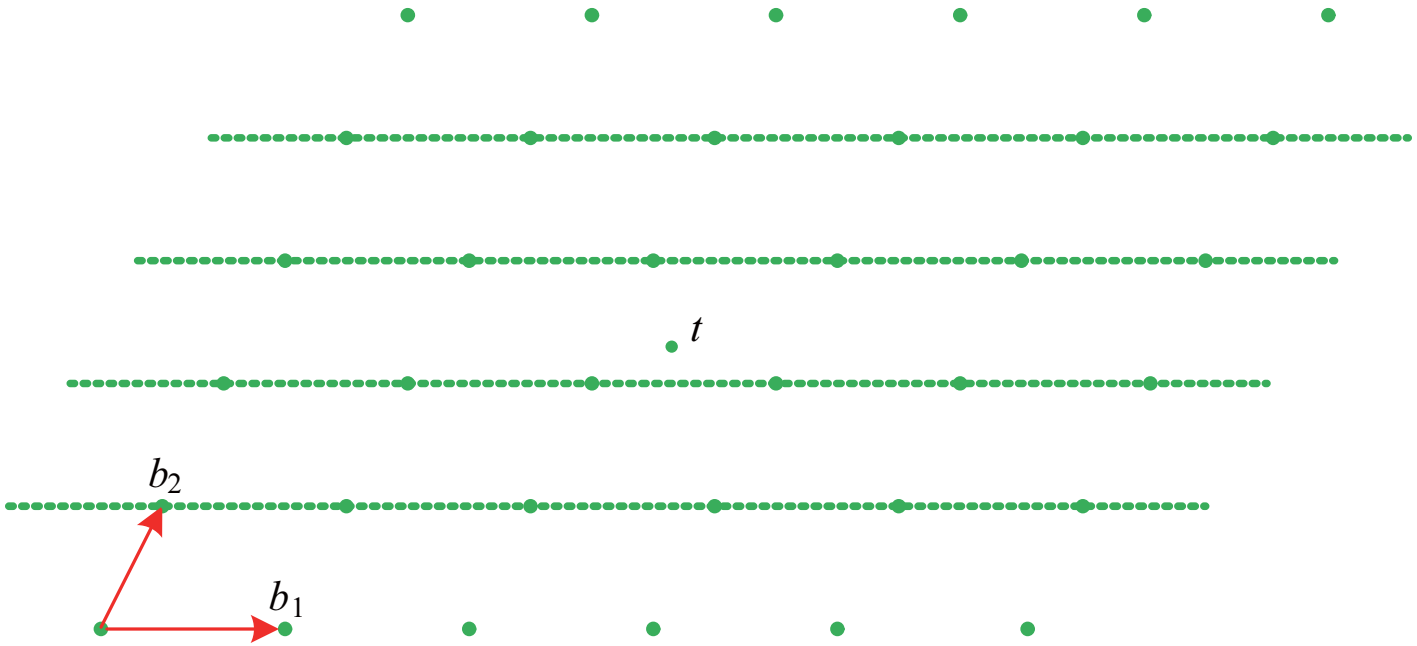
The NearestPlanes Algorithm



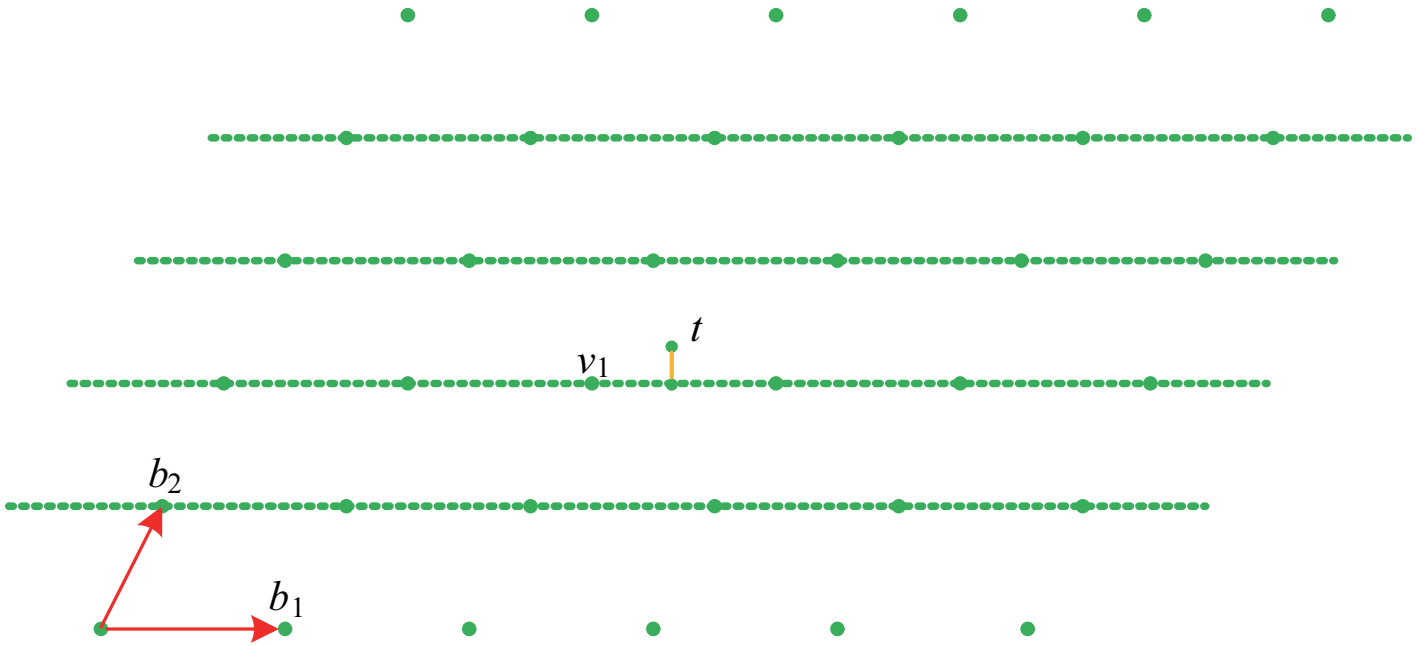
The NearestPlanes Algorithm



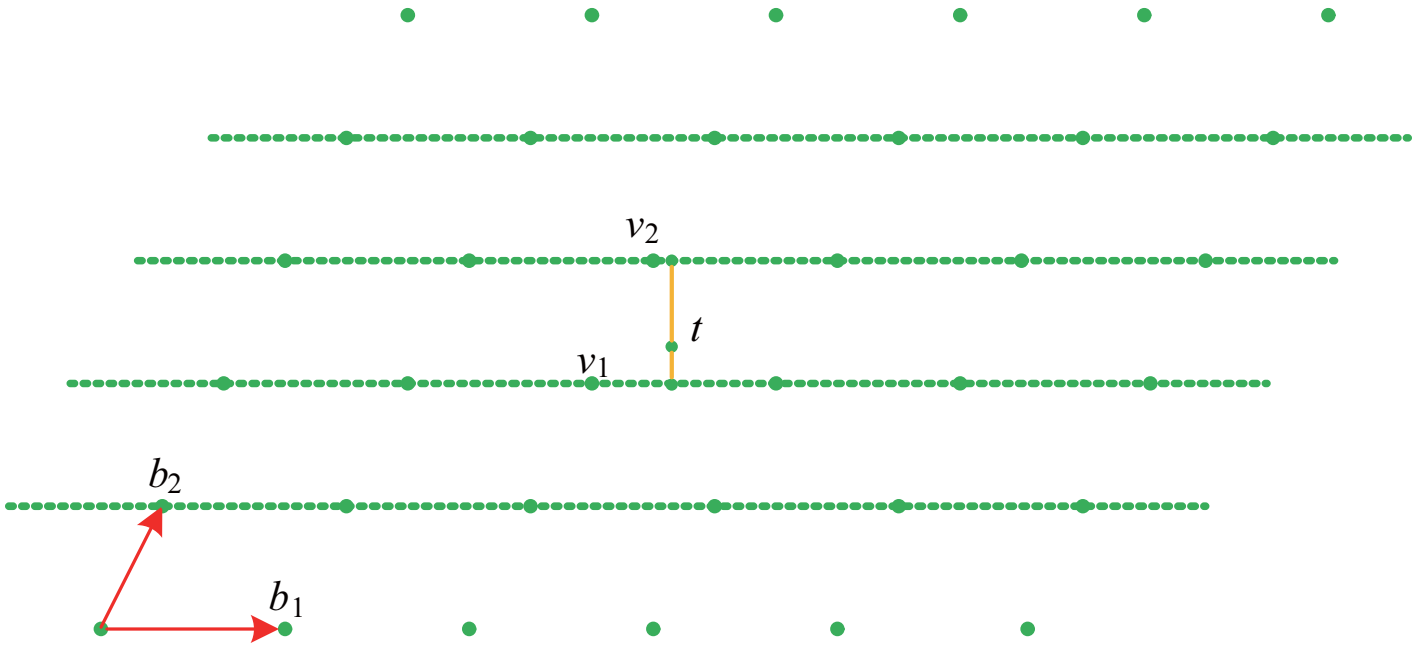
The NearestPlanes Algorithm



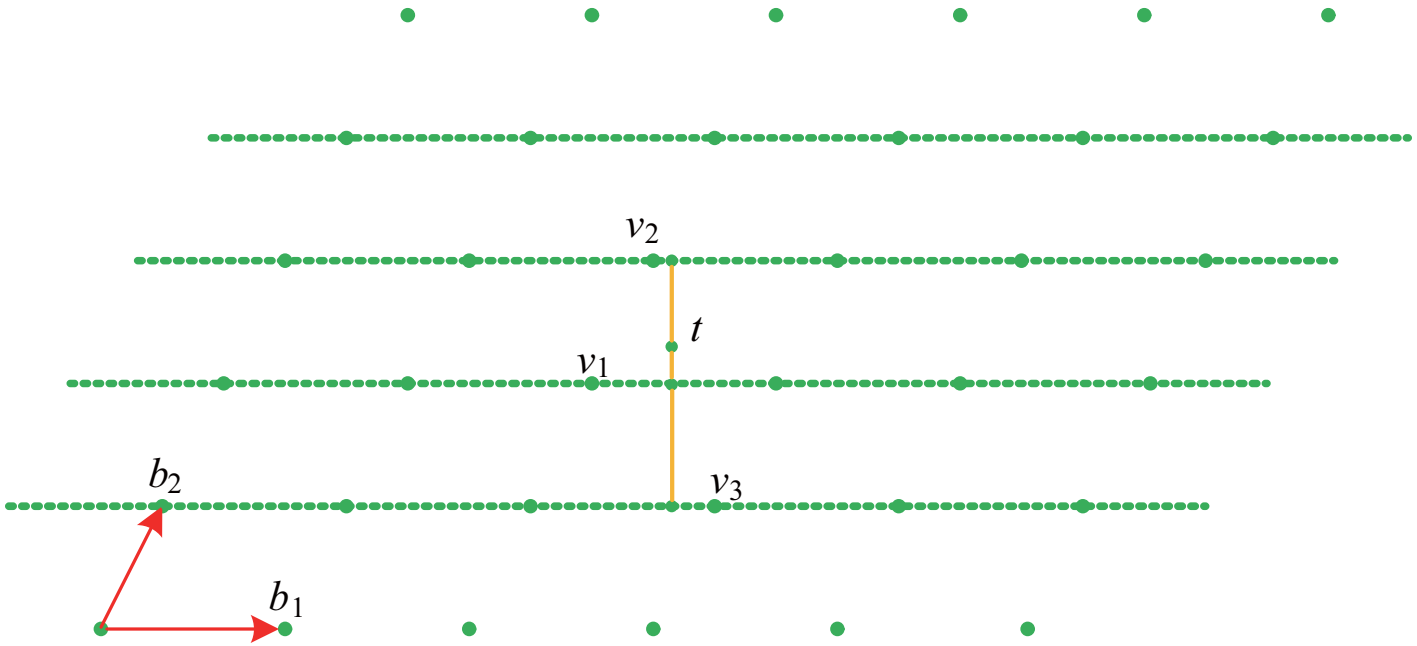
The NearestPlanes Algorithm



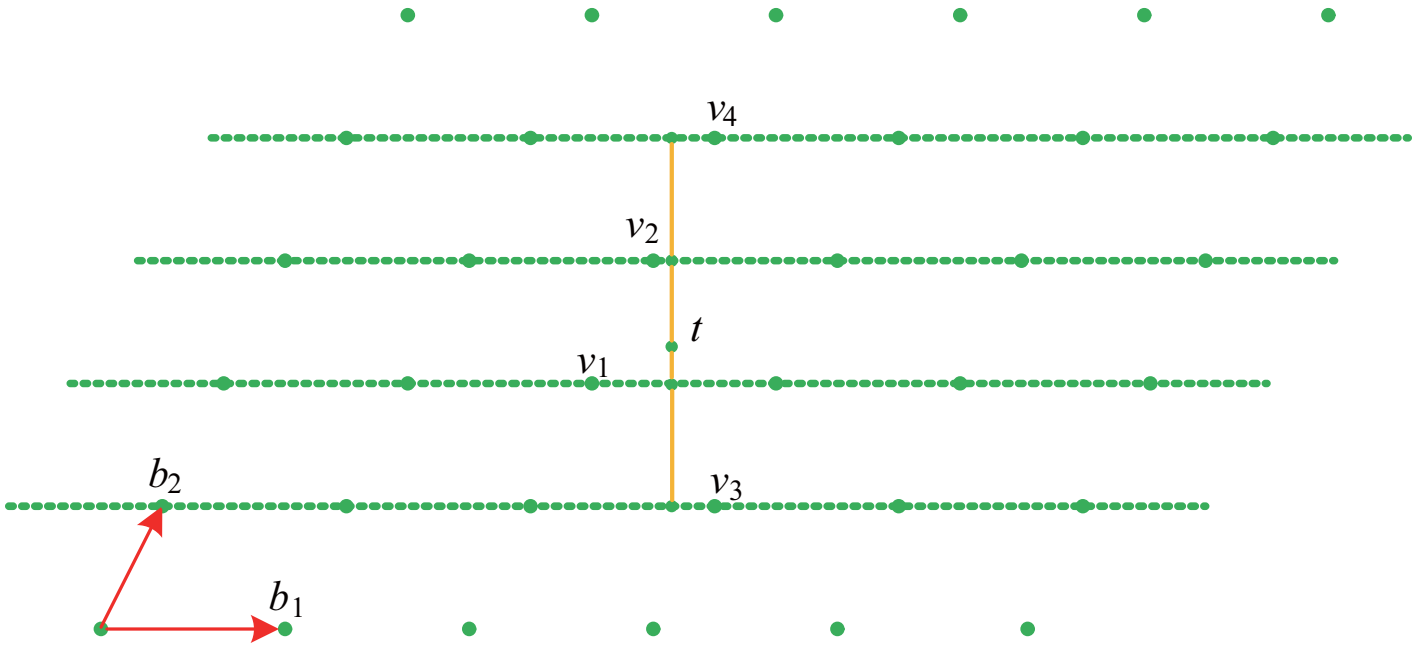
The NearestPlanes Algorithm



The NearestPlanes Algorithm



The NearestPlanes Algorithm



Outline

Motivation

Algorithm for BDD
- BDD Problem
- Babai NearestPlane Algorithm
- Lindner-Peikert's NearestPlanes Algorithm Revisited

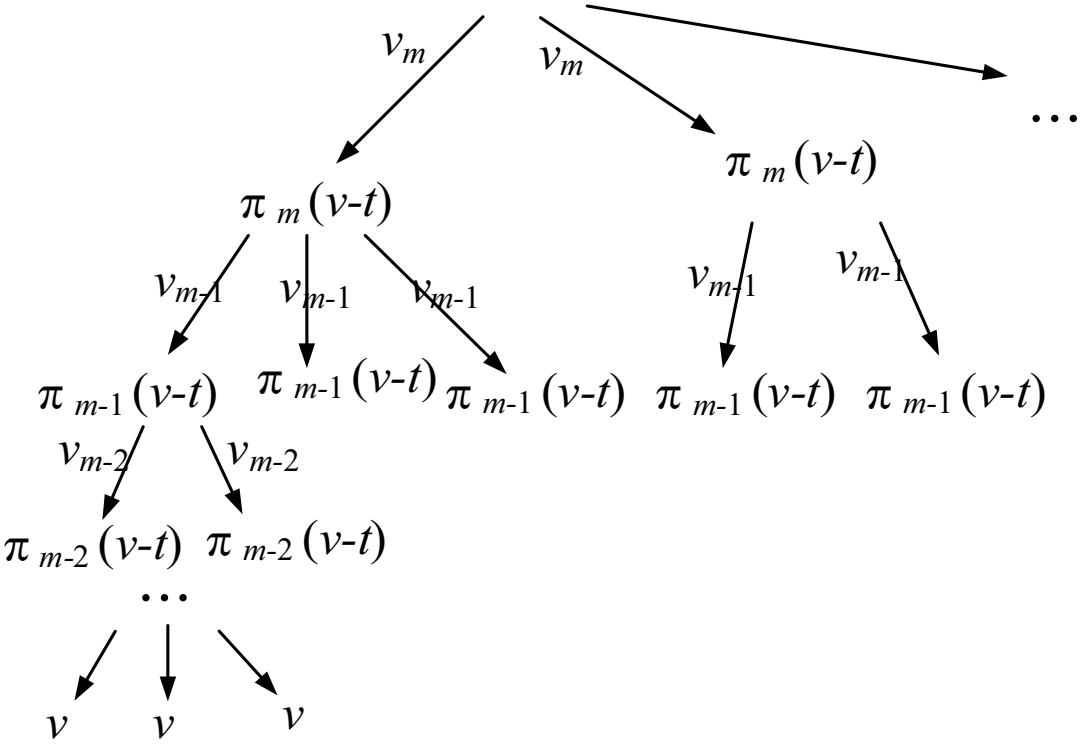
Solving BDD by (GNR) Pruned Enumeration
- Randomizing the NearestPlanes Algorithm
- GNR Pruned Enumeration to BDD

Conclusion

Lattice Enumeration for CVP

- Both Babai's algorithm and NearestPlanes algorithm can be viewed as a pruned enumeration:
  $\|\xi_i(\mathbf{v}-\mathbf{t})\| \leq \|\mathbf{b}_i^*\|/2$ Vs $\|\xi_i(\mathbf{v}-\mathbf{t})\| \leq \|d_i\mathbf{b}_i^*\|/2$
- Outline of enumeration
  - Step 1: Basis reduction
  - Step 2: Enumerate all lattice vector $\mathbf{v}$, $\|\mathbf{v}-\mathbf{t}\| \leq R$ by project lattice. R is the expected length of error vector.
- Target vector $\mathbf{t} = \Sigma_{i=1}^{m} t_i\mathbf{b}_i$, $\mathbf{v} = \Sigma_{i=1}^{m} v_i\mathbf{b}_i$. $\|\pi_i(\mathbf{v}-\mathbf{t})\| \leq R$, $i = 1, ..., m$
- $\|\pi_m(\mathbf{v}-\mathbf{t})\| \leq R \Longrightarrow v_m \in (t_m - \frac{R}{\|\mathbf{b}_m^*\|}, t_m + \frac{R}{\|\mathbf{b}_m^*\|})$
- For each value of v_m, $\|\pi_{m-1}(\mathbf{v}-\mathbf{t})\| \leq R$ implies that the integer v_{m-1} belongs to an interval of "small" length

Lattice Enumeration for CVP



Prunned Technique

- First proposed by Schnorr and Euchner 1994
- First rigorous analysis: Gama, Nguyen, Regev, 2010
- Cutting down branches to decrease the complexity at the cost of losing the solution. The success probability should be considered.

Prunned Technique

- Replace each inequality $\|\pi_{m-k+1}(\mathbf{v}-\mathbf{t})\| \leq R$ by $\|\pi_{m-k+1}(\mathbf{v}-\mathbf{t})\| \leq R_k$ for each index k , where $0 < R_1 \leq R_2 \leq \cdots R_m \leq R$.
- Linear pruning $R_k = \sqrt{\frac{k}{m}}R$.
- Choosing the optimal bounding function is very helpful for finding the solution.
- Take dimension 3 as a example, the search range from a ball to a cylinder $\begin{cases} x_1^2 \leq R_3^2 \\ x_1^2 + x_2^2 \leq R_2^2 \\ x_1^2 + x_2^2 + x_3^2 \leq R_1^2 \end{cases}$

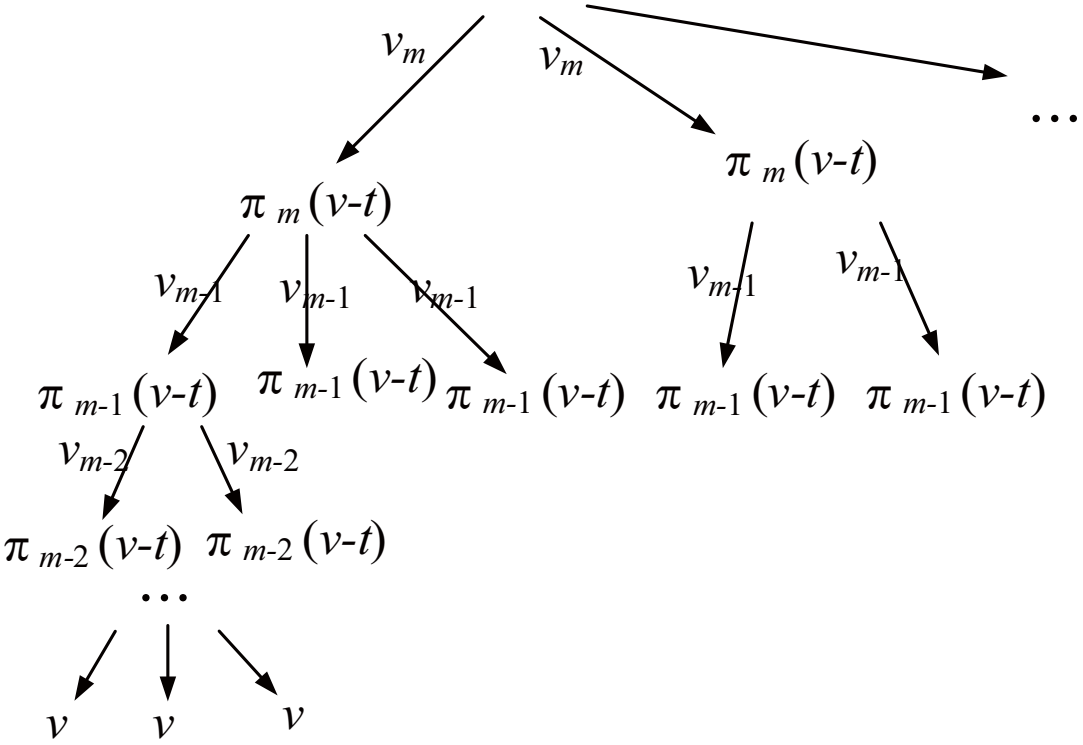
Algorithm

Repeat the following:

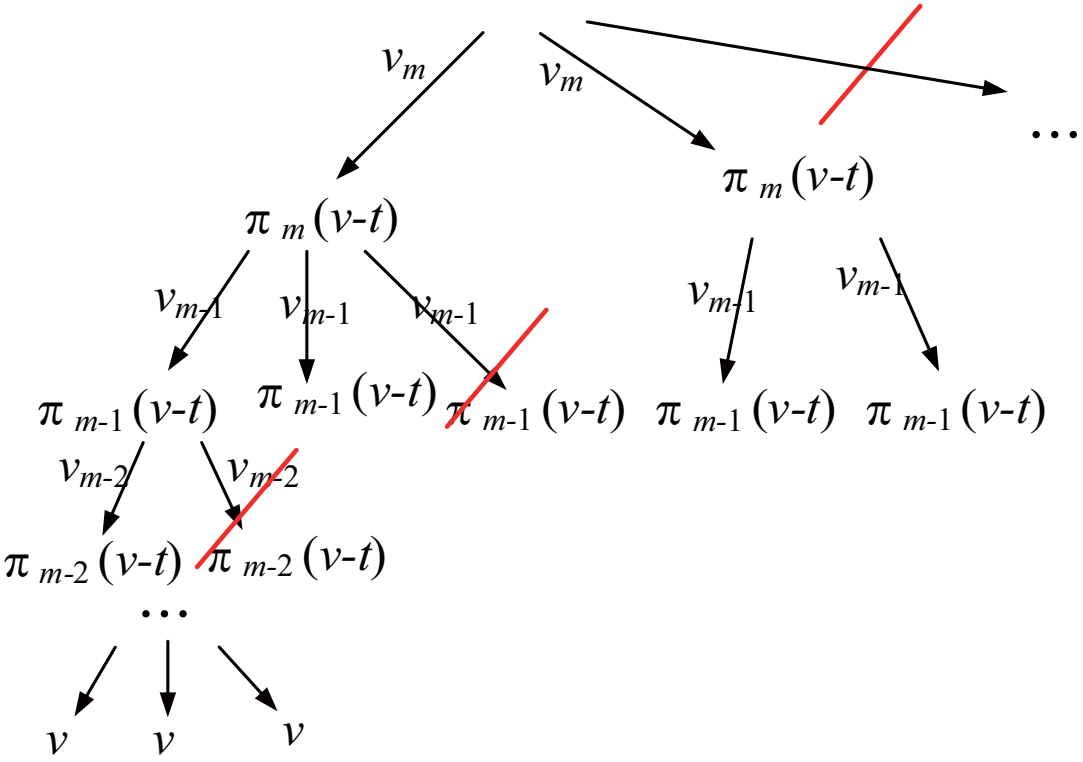
1. Generate a reduced basis

2. Do pruned enumeration

Prunned Technique



Prunned Technique

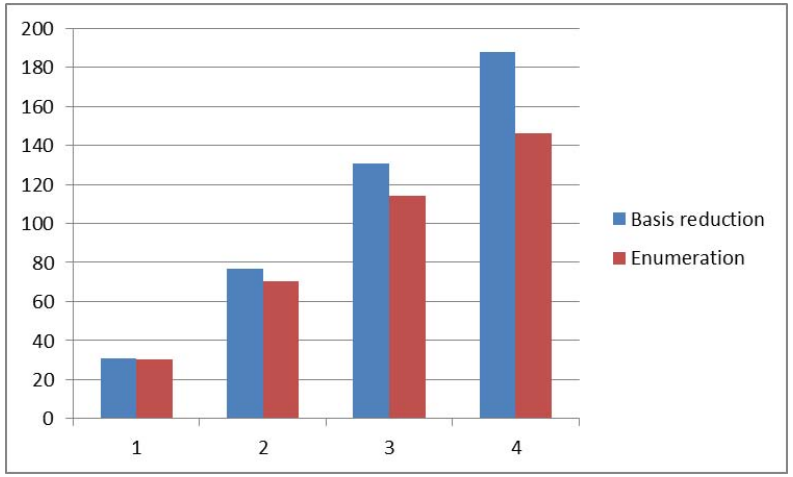


Complexity of Enumeration

- The complexity of enumeration is, up to a polynomial factor, the number of lattice points in all projected lattices inside the enumeration range.
- By the Gaussian heuristic, this number should be
  - Full enumeration : $\approx \Sigma_{1\leq k\leq m} v_k(R)/vol(\pi_{m-k+1}(L))$, where $v_k(R)$ is the volume of the k-dim ball of radius R.
  - Pruning technique: $\approx \Sigma_{1\leq k\leq m} vol(C_{R_1,R_2,\ldots,R_k})/vol(\pi_{m-k+1}(L))$, where $C_{R_1,R_2,\ldots,R_k} = \{(x_1, x_2, \ldots, x_k) \in \mathbb{R}^k, \forall j \leq k, \sum_{l=1}^{j} x_l^2 \leq R_j^2\}$.
- Total cost = $\frac{T_{redu}+T_{Enum}}{p_{succ}(R_1,\ldots,R_m)}$ where T_{Redu} is the basis reduction time, T_{Enum} is the time for enumerating

Randomizing the NearestPlanes Algorithm

- Similar to pruning, we randomize the Nearestplanes algorithm by repeating several times the basic algorithm with different randomized reduced bases
- The numerical data from [LP11] is far from optimal:the running times of basis reduction and enumeration are not totally balanced



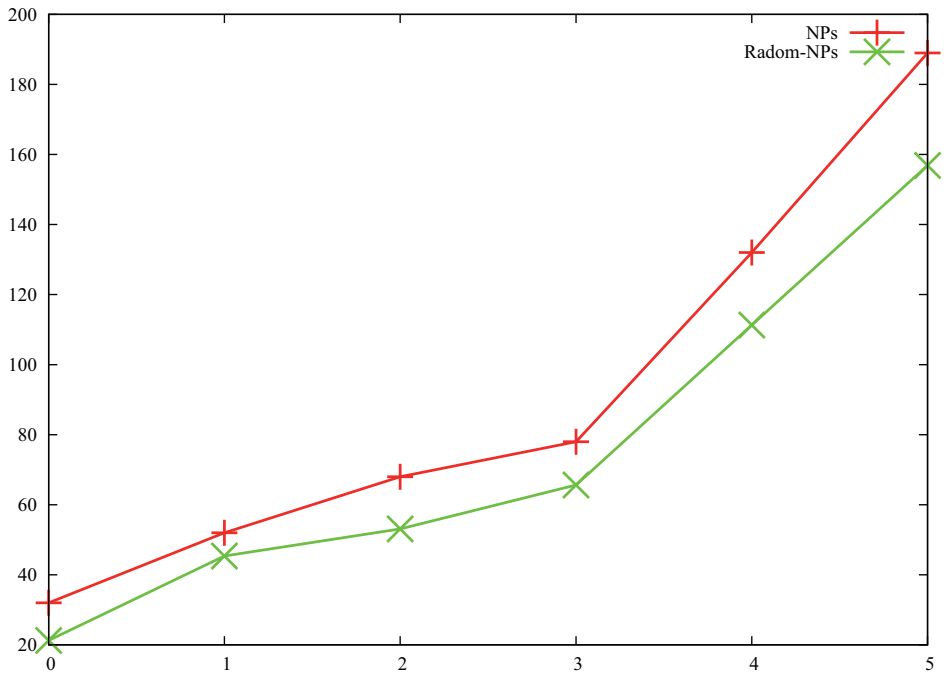
Numerical Comparison

- Our algorithm can get a better trade off.
- In the LWE parameters [LP11] consider, the comparison is following

			NearestPlanes				Randomized-NP					Log
n	q	s	δ	red	enum	Cost	δ	red	enum	pro	Cost	Speedup
128	2053	6.67	1.0089	30.8	47	32	1.0104	10.59	27	−9.65	21.24	10.76
192	4093	8.87	1.0067	76.84	87	78	1.0077	52.65	68.87	−11.99	65.64	12.36
256	4093	8.35	1.0052	130.55	131	132	1.006	98.56	114.96	−11.76	111.32	20.68
320	4093	8	1.0042	187.69	163	189	1.005	140.16	156.88	−15.7	156.86	32.14
136	2003	13.01	1.0071	66.35	82	68	1.0083	40.94	56.94	−11.17	53.11	14.89
214	16381	7.37	1.0078	50.58	66	52	1.0085	37.41	53.77	−6.94	45.35	6.65

Numerical Comparison

- Our algorithm can get a better trade off.
- In the LWE parameters [LP11] consider, the comparison is following



Outline

Motivation

Algorithm for BDD
- BDD Problem
- Babai NearestPlane Algorithm
- Lindner-Peikert's NearestPlanes Algorithm Revisited

Solving BDD by (GNR) Pruned Enumeration
- Randomizing the NearestPlanes Algorithm
- GNR Pruned Enumeration to BDD

Conclusion

Adapt Enumeration with Pruning to BDD

- Further improvement can be obtained by Pruned enumeration
- Previous: best algorithm for BDD is embedding method, our experiments show pruned enumeration performs better
- For SVP, the shortest vector is assumed uniformly distribute in the ball with radius R-expected length. For BDD, the shape of error affects the success probability, which make a theoretical analysis difficult. So compute this probability experimentally (by sampling) is essential

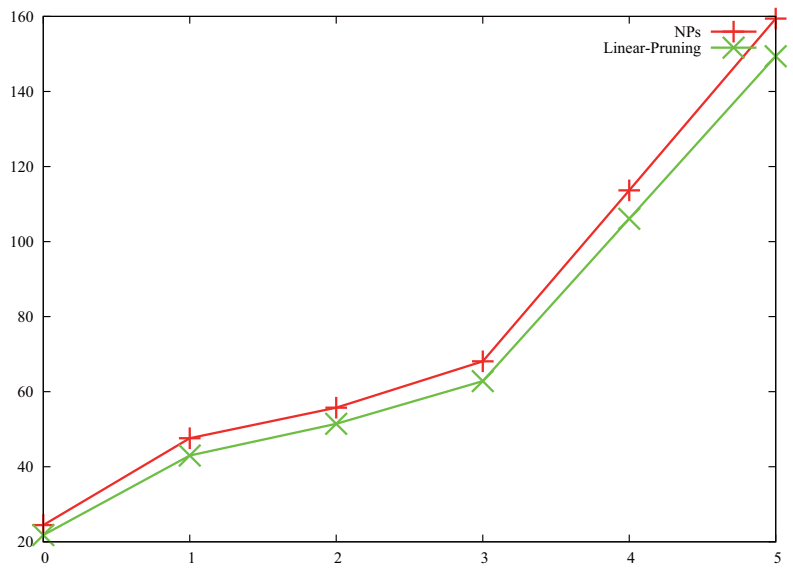
Application to LWE

- Bounding function: Linear, search $R^2 = c \cdot ||\text{expected length of error}||^2$

			Randomized-NP					Linear Pruning						Log
n	q	s	δ	red	enum	pro	Cost	δ	red	enum	pro	f-redius	Cost	Speedup
128	2053	6.67	1.0101	14.15	31.18	−9.28	24.46	1.0098	17.93	33.97	−2.98	1.08	21.8	2.66
192	4093	8.87	1.0075	56.98	74.04	−10.10	68.10	1.0074	59.23	74.88	−2.99	1.08	62.8	5.3
256	4093	8.35	1.006	98.56	115.56	−14.09	113.66	1.0059	102.09	116.8	−3.58	1.04	106.1	7.56
320	4093	8	1.005	140.16	157.17	−18.24	159.42	1.0049	145.25	159.76	−3.92	1.03	149.4	10.02
136	2003	13.01	1.0082	42.78	59.73	−11.95	55.73	1.008	46.58	63	−3.92	1.01	51.44	4.29
214	16381	7.37	1.0084	39.15	56.11	−7.43	47.59	1.0084	39.15	54.77	−3.24	1.06	43	4.59

Application to LWE

- Bounding function: Linear, search $R^2 = c \cdot ||\text{expected length of error}||^2$



Application to GGH

- Key recovery challenge: m BDD-instance with error vector chosen uniformly at random from $[-4, ..., +3]^m$

Dimension	200	250	300	300
BKZ blocksize	60	60	90	90
Bounding function	linear	linear	linear	optimized
Estimated Nb of Nodes	800	5.84×10^8	7.58×10^{10}	1.33×10^9
Average Nb of Nodes	666	5.93×10^8	-	1.35×10^9
Success probability	0.0418	0.0409	0.0371	0.0185
Nb of Success	4	11	-	2

Table 1: Key-recovery for GGH Challenges

Application to GGH

- Message recovery challenge: In 1999, Nguyen showed it can be reduce to a BDD instance with error in $\{-\frac{1}{2}, \frac{1}{2}\}^m$. Using the embedding method and BKZ 20 basis , he solved dimension 200,250 and 300 instances. For dimension 350, pruned-BKZ with blocksize 60 is required.
- For dimension 350, we do enumerate with pruning to recover 350-dimension message
  - A bounding function according to the error distribution. $R_k^2 = \min\{E(X_k) + 3D(X_k), 1\}R_m^2$, $E(X_k) = \frac{k}{m}$, $D(X_k) = \sqrt{\frac{k(m-k)}{m^2(\frac{m}{2}+1)}}$.
  - Success probability 92%
  - BKZ-20 basis
  - Time: 2935.421s in a single 3-Ghz Intel-Core2 core

Application to DSA

- Each DSA signature generation require the use of a one-time key k modulo q, where q is usually a 160-bit prime number.
- In 2002, Nguyen and Shparlinski showed that disclosing ℓ bits of each one-time key k for several (message,signature) pair to recover the DSA secret key is a HNP problem $\Longrightarrow$ BDD instance
- They used embedding method to recover the DSA secret key in a few hours, given the $\ell = 3$ least significant bits of each one-time key for about 100 signatures, but the attack failed for $\ell = 2$.
- Enumeration with pruning: using BKZ-90 reduction[CN11] and linear pruning, $\ell = 2$ case, given about 100 signatures, 4185 seconds, success probability 23%

Application to DSA

- Choose the enumeration radius
- Here, the x-coordinate is the ratio between the (squared) enumeration radius and the (squared) expected length the error vector. The y-coordinate is $\log_2$(Number of enumeration nodes/Success probability)

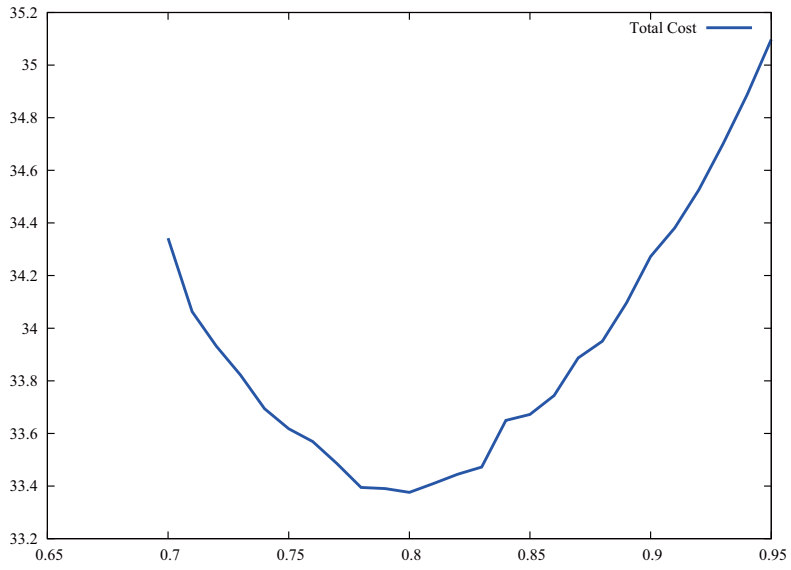


Figure 1: Total cost for DSA

Conclusion

- Our work shows that any security estimate of BDD-based cryptosystems must take into account enumeration attacks
- NearestPlanes BDD algorithm [LP11] does not seem to offer any practical advantage over GNR pruning [GNR10], despite having appeared later.
- BDD enumeration can be practical even in high dimension like 350

Thank you for your attention!