

Some Algorithms for LPN and LWE Problems

Haibin Kan

School of Computer Science, Fudan University

Learning Parity with Noise (LPN)

A central problem in learning theory.

- $\mathbf{s} \in \mathbb{F}_2^n$
- $\mathbf{a} \stackrel{\$}{\leftarrow} \mathbb{F}_2^n$
- $e \leftarrow \text{Ber}_\eta, 0 < \eta < \frac{1}{2}$
- Denote by $A_{\mathbf{s},\eta}$ the distribution of $(a, \langle \mathbf{s}, \mathbf{a} \rangle + e)$
(and the oracle outputting samples of $A_{\mathbf{s},\eta}$ as well)

The *decisional* LPN problem asks to distinguish $A_{\mathbf{s},\eta}$ from the uniform distribution of $\mathbb{F}_2^n \times \mathbb{F}_2$.

The *search* LPN problem asks to find \mathbf{s} , given access to the oracle $A_{\mathbf{s},\eta}$.

Learning with Errors (LWE)

Introduced by Oded Regev¹.

- $\mathbf{s} \in \mathbb{Z}_q^n$
- $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ for an odd prime q
- e is a discrete Gaussian variable on \mathbb{Z}_q

The LWE problem asks to find \mathbf{s} , given independent samples of $(\mathbf{a}, \langle \mathbf{s}, \mathbf{a} \rangle + e)$.

LWE is a generalization of LPN to larger moduli.

¹O. Regev, On lattices, learning with errors, random linear codes, and cryptography, in Proceedings of the thirty-seventh annual ACM symposium on Theory of computing - STOC 05, 2005, p. 84.

Features

- Hardness: Both are believed to be hard.
- Efficiency: More efficient cryptographic primitives, compared to number theoretical ones.
- Versatility: Fully Homomorphic Encryption (LWE), light-weight cryptography (LPN).
- Post-quantum security: No polynomial time quantum algorithm is known.

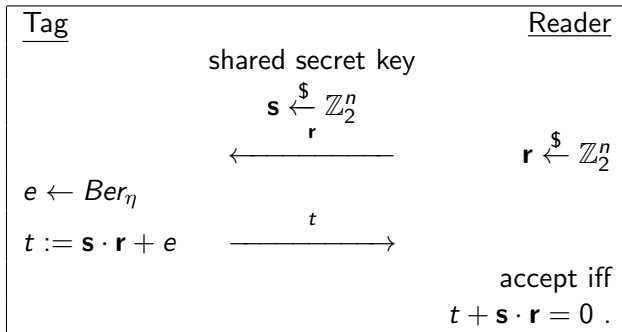
Some Simple Cryptosystems Based on LPN/LWE

- HB: A 2-round LPN-based authentication protocol.²
- HB+: A 3-round modification of HB, with improved security.³
- Regev's first LWE-based public-key encryption scheme.⁴

²Nicholas J. Hopper, Manuel Blum: Secure Human Identification Protocols. ASIACRYPT 2001: 52-66

³Ari Juels, Stephen A. Weis: Authenticating Pervasive Devices with Human Protocols. CRYPTO 2005: 293-308.

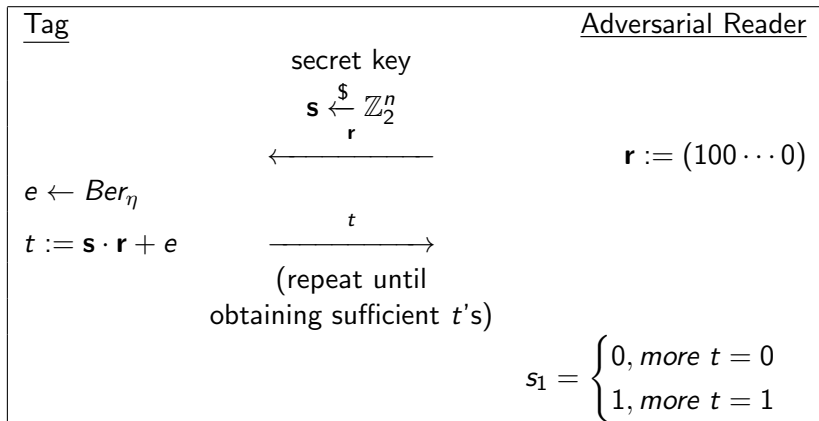
⁴Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. J. ACM, 56(6):140, 2009. Preliminary version in STOC 2005.



LPN-based authentication protocols are suitable for RFID tags because of their small code size and low communication complexity.

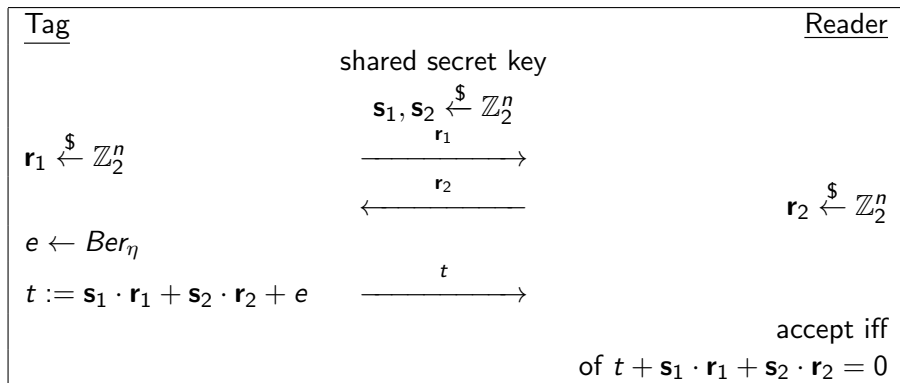
A Simple Active Attack on HB

HB is proved to be secured (assuming the hardness of LPN) against *passive* eavesdroppers. However, it is easily broken by an *active* adversary who is allowed to interact with tags.



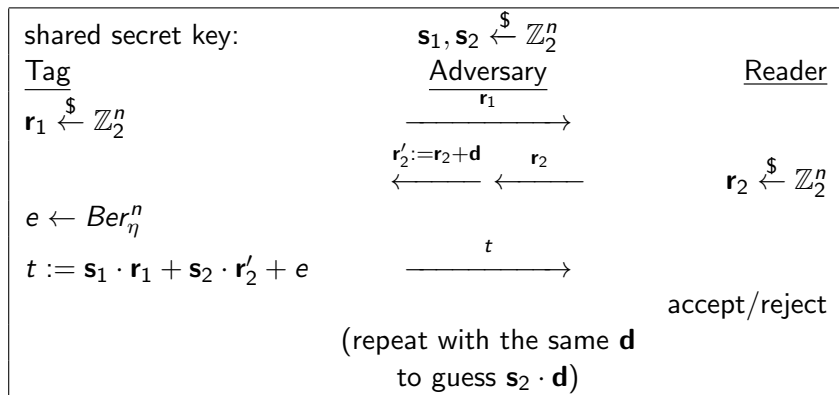
HB+

Hence, HB+ is proposed to achieve the security against the previous mentioned adversary.



HB+ is vulnerable to the following man-in-the-middle attack.

An Attack on HB+ ⁵



⁵[1] H. Gilbert, M. J. B. Robshaw, and H. Sibert, An Active Attack Against HB+ - A Provably Secure Lightweight Authentication Protocol, IACR Cryptol. ePrint Arch., vol. 2005, p. 237, 2005.

Regev's LWE Cryptosystem

- Key-generation:

- Secret key: $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$
- Public key: $m \approx (n+1) \log q$ LWE samples, in matrix form

$$\mathbf{A} = \begin{bmatrix} \bar{\mathbf{A}} \\ \mathbf{b}^t \end{bmatrix} \in \mathbb{Z}_q^{(n+1) \times m}$$

where $\mathbf{b}^t = \mathbf{s}^t \bar{\mathbf{A}} + \mathbf{e}^t \pmod q$.

- Encryption:

$$\mathbf{c} = \text{Enc}(\mu) = \mathbf{A}\mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \mu \lfloor \frac{q}{2} \rfloor \end{bmatrix} \in \mathbb{Z}_q^{n+1}$$

where $\mu \in \mathbb{Z}_2$, $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_2^m$.

Regev's LWE Cryptosystem (Cont'd)


- Decryption: Compute

$$\mu' = [-\mathbf{s}^t \ 1] \mathbf{c} = [-\mathbf{s}^t \ 1] \begin{bmatrix} \bar{\mathbf{A}} \\ \mathbf{b}^t \end{bmatrix} \mathbf{x} + [-\mathbf{s}^t \ 1] \begin{bmatrix} \mathbf{0} \\ \mu \lfloor \frac{q}{2} \rfloor \end{bmatrix} = \mathbf{e}^t \mathbf{x} + \mu \lfloor \frac{q}{2} \rfloor$$

$$\mu = Dec(\mathbf{c}) = \begin{cases} 0, & \mu' \text{ is closer to } 0 \\ 1, & \mu' \text{ is closer to } \lfloor \frac{q}{2} \rfloor \end{cases}$$

BKW Algorithm

- Proposed by Blum, Kalai and Wasserman⁶.
- Solves LPN, and is naturally adopted to LWE.
- Solves LPN in $2^{O(n/\log n)}$ time.

⁶A. Blum, A. Kalai, and H. Wasserman, Noise-tolerant learning, the parity problem, and the statistical query model, J. ACM, vol. 50, no. 4, pp. 506-519, Jul. 2003. 

The Main Idea: Reduction

- Group the samples $\{(\mathbf{a}, y = \langle \mathbf{s}, \mathbf{a} \rangle + e)\}$ by the last b entries of \mathbf{a} .
- For each group, pick one sample (\mathbf{a}_0, y_0) arbitrarily, and subtract it from all other samples of the group.

$$(\mathbf{a}_0, y_0) = (\mathbf{a}_0, \langle \mathbf{s}, [\mathbf{a}'_0, \tilde{\mathbf{a}}] \rangle + e_0)$$

$$(\mathbf{a}_1, y_1) = (\mathbf{a}_1, \langle \mathbf{s}, [\mathbf{a}'_1, \tilde{\mathbf{a}}] \rangle + e_1)$$

$$(\mathbf{a}_1, y_1) := (\mathbf{a}_0, y_0) + (\mathbf{a}_1, y_1) = ([\mathbf{a}'_0 + \mathbf{a}'_1, \mathbf{0}], \langle \mathbf{s}, [\mathbf{a}'_0 + \mathbf{a}'_1, \mathbf{0}] \rangle + e_0 + e_1)$$

Then, discard (\mathbf{a}_0, y_0) .

- Note that the last b entries of all \mathbf{a} 's become zero.
- By applying the reduction for t times, we can reduce the dimension of the LPN instance (i.e. $\dim(\mathbf{s})$) by bt .
- However, the level of the noise increased.

Piling-Up Lemma

Let X_1, X_2, \dots, X_n be independent Bernoulli variables with $\Pr[X_i = 0] = \frac{1}{2}(1 + \epsilon_i)$. Then,

$$\Pr\left[\sum_i X_i = 0\right] = \frac{1}{2}\left(1 + \prod_i \epsilon_i\right)$$

ϵ_i is called the bias of X_i .

- By the piling-up lemma, the bias of the noise becomes ϵ^{2^t} after t reductions.
- Thus too large t is not plausible as it makes the noise reach a high level.

Final Step

- Standard approach by BKW
- LF1 by Leveil & Fouque⁷
- Covering code approach by Guo, Johansson & Löndahl⁸

⁷. Leveil and P.-A. Fouque, An Improved LPN Algorithm, in SCN 2006: Security and Cryptography for Networks, vol. 4116, R. De Prisco and M. Yung, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 348-359.

⁸Q. Guo, T. Johansson, and C. Lndahl, Solving LPN Using Covering Codes, in Advances in Cryptology ASIACRYPT 2014, no. 61170208, P. Sar and T. Iwata, Eds. Springer Berlin Heidelberg, 2014, pp. 1-20.

- Keep the samples where $\mathbf{a} = \mathbf{e}_j$ only.
- For each j , determine s_j by majority voting.

A simple approach, but wasting a lot of samples.

A guessing-and-verifying approach

- For every $\mathbf{s}' \in \mathbb{F}_2^{n'}$
 - Compute $\langle \mathbf{s}', \mathbf{a} \rangle + y$ for all samples (\mathbf{a}, y) .
 - Do majority voting.
 - $\text{bias} \approx 0 \rightarrow$ wrong
 - $\text{bias} \approx \epsilon^{2^t} \rightarrow$ correct

LF1 (Cont'd)

$$\langle \mathbf{s}', \mathbf{a} \rangle + y = \langle \mathbf{s}' + \mathbf{s}, \mathbf{a} \rangle + e$$

- If $\mathbf{s}' = \mathbf{s}$,
 $\langle \mathbf{s}', \mathbf{a} \rangle + y = e$, bias = ϵ^{2t}
- If $\mathbf{s}' \neq \mathbf{s}$,
 $\langle \mathbf{s}' + \mathbf{s}, \mathbf{a} \rangle = \sum_{i|s'_i \neq s_i} a_i$, which is a sum of $Ber_{1/2}$ variables, i.e. bias = 0.
By piling-up lemma, $\langle \mathbf{s}' + \mathbf{s}, \mathbf{a} \rangle + e$ is 0-biased as well.

Covering Code Approach

- Gaussian elimination (preprocessing).
- Reduction
- Partial guessing
- Decoding
- Guessing & verifying

Gaussian Elimination

For m samples (in matrix form)

$$\mathbf{y} = \mathbf{s}\mathbf{A}_{n \times m} + \mathbf{e} = \mathbf{s}[\mathbf{T}_{n \times n} | \mathbf{A}'_{n \times (m-n)}] + \mathbf{e}$$

- Assume \mathbf{T} is invertible.
(This condition can be satisfied with a high probability by shuffling the samples.)
- $\mathbf{y} = \mathbf{s}\mathbf{T}\mathbf{T}^{-1}[\mathbf{T} | \mathbf{A}'] + \mathbf{e} = (\mathbf{s}\mathbf{T})[\mathbf{I} | \mathbf{T}^{-1}\mathbf{A}'] + \mathbf{e}$
- $\mathbf{y} + [y_1, \dots, y_n][\mathbf{I} | \mathbf{T}^{-1}\mathbf{A}'] = (\mathbf{s}\mathbf{T} + [y_1, \dots, y_n])[\mathbf{I} | \mathbf{T}^{-1}\mathbf{A}'] + \mathbf{e}$
- Obtain a new instance $\hat{\mathbf{y}} = \hat{\mathbf{s}}\hat{\mathbf{A}} + \mathbf{e}$.
 - $\hat{\mathbf{A}}$ is systematic.
 - $\hat{y}_1 = \dots = \hat{y}_n = 0$, thus $\hat{\mathbf{s}} = [e_1, \dots, e_n]$.

By this means the upper bound of $wt(\mathbf{s})$ is lowered.

Reduction

Applying the iterative reduction on \mathbf{A} 's non-systematic columns for t times, We get an instance of smaller dimension as

$$\mathbf{y} = \mathbf{s}\mathbf{A}_{n' \times m'} + \mathbf{e}, \quad n' = n - tb, \quad e_i \sim \text{Ber}_{\frac{1}{2}}(1 - \epsilon^{2t})$$

Partial Secret Guessing

Proposed by Bernstein & Lange⁹.

- Divide \mathbf{s} into $[\mathbf{s}_1, \mathbf{s}_2]$ appropriately.
- Guess all possible values of \mathbf{s}_2 satisfying $wt(\mathbf{s}_2) \leq w_0$ for a preset w_0 .
 - Update y of each sample accordingly.
 - Leave \mathbf{s}_1 to be decided by the following steps.

⁹D. J. Bernstein and T. Lange, Never Trust a Bunny, in Radio Frequency Identification. Security and Privacy Issues, vol. 7739 LNCS, J.-H. Hoepman and I. Verbauwhede, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 137-148.

Decoding

- Assume we have enough samples $z = (\mathbf{a}_z, y_z) \leftarrow A_{\mathbf{s}, \eta}$ where \mathbf{s} and η are consistent with the results after the previous steps.
- Assume we have constructed a $[k, l]$ -linear code \mathcal{C} with good covering property, i.e. the covering radius $d_{\mathcal{C}}$ is small enough. ($\dim(\mathbf{s}) = k$)
- Treat each \mathbf{a}_z as an erroneous codeword of \mathcal{C} , and decode it:
 $\mathbf{a}_z = \mathbf{c}_z + \mathbf{d}_z, \quad wt(\mathbf{d}_z) \leq d_{\mathcal{C}}$
- Denote by \mathbf{G} the generator matrix of \mathcal{C} ; assume $\mathbf{c}_z = \mathbf{u}_z \mathbf{G}$.

$$\begin{aligned} y_z &= \langle \mathbf{s}, \mathbf{a}_z \rangle + e_z \\ &= \langle \mathbf{s}, \mathbf{c}_z + \mathbf{d}_z \rangle + e_z \\ &= \langle \mathbf{s}, \mathbf{c}_z \rangle + \tilde{e}_z \quad (\tilde{e}_z \stackrel{def}{=} \langle \mathbf{s}, \mathbf{d}_z \rangle + e_z) \\ &= \langle \mathbf{s} \mathbf{G}^T, \mathbf{u}_z \rangle + \tilde{e}_z \end{aligned}$$

Analysis of $\tilde{\mathbf{e}}_z$

- As d_C is small, it is convenient to view \mathbf{d}_z as a sparse vector, and $\mathbf{d}_z[i]$'s as independent $Ber_{d_C/k}$ variables.
- Denote the bias of $\mathbf{d}_z[i]$ by γ . Assume $wt(\mathbf{s}) \leq w$. By piling-up lemma, the bias of $\tilde{\mathbf{e}}_z$ is at least $\epsilon^{2^t} \gamma^w$.

Guessing & Verifying

For every $\mathbf{t}' \in \mathbb{F}_2^l$,

- Compute $y_z + \langle \mathbf{t}', \mathbf{u}_z \rangle$ for all z 's.
- Do majority voting.
 - bias $\approx 0 \rightarrow$ wrong
 - bias $\geq \epsilon^{2^t} \gamma^w \rightarrow$ correct

Guessing & Verifying (Cont'd)

$$y_z + \langle \mathbf{s}', \mathbf{c}_z \rangle = \langle \mathbf{s} + \mathbf{s}', \mathbf{c}_z \rangle + \tilde{\mathbf{e}}_z,$$

- If $\mathbf{s}' = \mathbf{s}$,
 $y_z + \langle \mathbf{s}', \mathbf{c}_z \rangle = \tilde{\mathbf{e}}_z$, bias = $\epsilon^{2t} \gamma^w$
- If $\mathbf{s}' \neq \mathbf{s}$,
 $\langle \mathbf{s}', \mathbf{c}_z \rangle = \sum_{i|s'_i \neq s_i} c_{zi}$, which is a sum of $Ber_{1/2}$ variables, i.e. bias = 0.

Standard BKW Algorithm for LWE

- The procedure is similar to that of BKW Algorithm for LPN.
- The main difference is the distribution of the error.

Recall: Learning with Errors (LWE)

- $\mathbf{s} \in \mathbb{Z}_q^n$
- $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ for an odd prime q
- e is a discrete Gaussian variable on \mathbb{Z}_q

The LWE problem asks to find \mathbf{s} , given independent samples of $z = (\mathbf{a}, \langle \mathbf{s}, \mathbf{a} \rangle + e)$.

Discrete Gaussian Distribution

- The discrete Gaussian distribution on \mathbb{Z} , with the mean 0 and the variance σ^2 , is denoted by $D_{\mathbb{Z},\sigma}$, whose probability function is defined as

$$p_{D_{\mathbb{Z},\sigma}}(x) \propto \exp(-x^2/2\sigma^2), \quad x \in \mathbb{Z}$$

- The discrete Gaussian distribution \mathcal{X}_σ on \mathbb{Z}_q is defined as

$$p_{\mathcal{X}_\sigma}(x) = \sum_{y \equiv x \pmod q} p_{D_{\mathbb{Z},\sigma}}(y), \quad x \in \mathbb{Z}_q$$

Analysis of the Errors

- Similarly to the case of LPN, the errors of the samples after t reduction iterations is the sum of 2^t independent discrete Gaussian variables.
- The discrete Gaussian does not necessarily behave exactly as the continuous one does. However in our case, it is reasonable to believe

$$X_1 + \cdots + X_n \sim \mathcal{X}_{\sqrt{\sigma_1^2 + \cdots + \sigma_n^2}}$$

for $X_1 \sim \mathcal{X}_{\sigma_1}, \dots, X_n \sim \mathcal{X}_{\sigma_n}$.

- The verifying step is to distinguish $\mathcal{X}_{2^{t/2}\sigma}$ from the uniform distribution.

Coded-BKW Algorithm¹⁰


- Construct a q -ary $[N, k]$ linear code \mathcal{C} .
- For each sample $z = (\mathbf{a}, y)$
 - Partition \mathbf{a} into $[\mathbf{a}', \mathbf{a}'']$ with $\dim(\mathbf{a}'') = N$.
 - Find the codeword \mathbf{c} closest to \mathbf{a}'' .
- Group the samples by $\pm \mathbf{c}$.
- In each group, pick a sample z_0 arbitrarily, and use it to eliminate \mathbf{c} in other samples.

$$z_0 = ([\mathbf{a}'_0, \mathbf{c} + \mathbf{d}_0], y_0)$$

$$z_1 = ([\mathbf{a}'_1, \mathbf{c} + \mathbf{d}_1], y_1)$$

$$z_1 := z_1 - z_0 = ([\mathbf{a}'_1 - \mathbf{a}'_0, \mathbf{d}_1 - \mathbf{d}_0], y_1 - y_0)$$

- Then, discard z_0 .

¹⁰Q. Guo, T. Johansson, and P. Stankovski, Coded-BKW: Solving LWE using lattice codes, in Advances in Cryptology – CRYPTO 2015, vol. 9215, no. 12, R. Gennaro and M. Robshaw, Eds. Springer Berlin Heidelberg, 2015, pp. 2342. 

Coded-BKW Algorithm (Cont'd)

Suppose, for $1 \leq h \leq t$, an $[N_h, k]$ linear code \mathcal{C}_h is used, and N_h positions are "eliminated" in the h -th iteration. For each sample surviving t iterations,

- $e \rightarrow$ a sum of 2^t Gaussian errors.
- $\mathbf{a} = [\mathbf{a}_0, \mathbf{a}_t, \mathbf{a}_{t-1}, \dots, \mathbf{a}_1]$, $\dim(\mathbf{a}_h) = N_h$, $\dim(\mathbf{a}_0) = n - \sum_h N_h$
 - $\mathbf{a}_0 \rightarrow$ uniform.
 - $\mathbf{a}_t \rightarrow$ a sum of 2 coding errors of \mathcal{C}_t .
 - $\mathbf{a}_{t-1} \rightarrow$ a sum of 2^2 coding errors of \mathcal{C}_{t-1} .
 - \vdots
 - $\mathbf{a}_1 \rightarrow$ a sum of 2^t coding errors of \mathcal{C}_1 .

Analysis of the Error (Sketch)

Hence the error induced by coding is (summing over all indices i and iterations h)

$$\sum_{i=n-\sum N_{h+1}}^n s_i(E_i^{(1)} + E_i^{(2)} + \dots + E_i^{(t)})$$

where $E_i^{(h)} = \sum_{j=1}^{2^{t-h+1}} d_{i,j}^{(h)}$.

Note that for every index i , at most one $E_i^{(h)}$ term is non-zero.

We simply view each $d_{i,j}^{(h)}$ as a discrete Gaussian of \mathcal{X}_{σ_h} . Thus, $E_i^{(h)}$ is the sum of 2^{t-h+1} independent \mathcal{X}_{σ_h} variables, i.e.

$$E_i^{(h)} \sim \mathcal{X}_{2^{(t-h+1)/2}\sigma_h}$$

Analysis of the Error (Cont'd)

Our goal is to limit the level of the coding errors by carefully choosing the parameters N_h .

If we manage to guarantee $E_i^{(h)} \sim \mathcal{X}_{\sigma_{set}}$ for some preset limit σ_{set} , the total error induced by coding will be of $\mathcal{X}_{\|\mathbf{s}_{[n-\sum N_{h+1}\dots n]}\| \sigma_{set}}$ approximately.

Analysis of the Error (Cont'd)

For each \mathcal{C}_h , we construct a lattice $\Lambda(\mathcal{C}_h)$ (Construction A)

$$\Lambda(\mathcal{C}_h) = \{\lambda \in \mathbb{R}^n : \lambda \equiv \mathbf{c} \pmod{\mathbf{q}}, \mathbf{c} \in \mathcal{C}_h\}$$

The second moment of $\Lambda(\mathcal{C}_h)$, denoted by $\sigma^2(\Lambda(\mathcal{C}_h))$, is a good estimation of σ_h^2 .

$$\sigma^2(\Lambda) \stackrel{\text{def}}{=} \frac{1}{n} \int_{\mathcal{V}} \|\mathbf{x}\|^2 \frac{1}{\text{Vol}(\mathcal{V})} d\mathbf{x}$$

where \mathcal{V} denotes the fundamental Voronoi region of Λ .

$$\mathcal{V} \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \leq \|\mathbf{x} - \mathbf{w}\|, \forall \mathbf{w} \in \Lambda\}$$

Analysis of the Error (Cont'd)

The normalized second moment $G(\Lambda)$ is defined as

$$G(\Lambda) = \frac{\sigma^2(\Lambda)}{\text{Vol}(\mathcal{V})^{\frac{2}{n}}}$$

which is known to be bounded as

$$\frac{1}{2\pi e} < G(\Lambda) \leq \frac{1}{12}$$

Analysis of the Error (Cont'd)

For $\Lambda(\mathcal{C}_h)$, Construction A of \mathcal{C}_h , $\text{Vol}(\mathcal{V}) = q^{N_h - k}$.

Hence,

$$\sigma_h^2 \approx q^{2(1-k/N_h)} G(\Lambda)$$

Thus N_h is determined by

$$\sigma_{\text{set}}^2 = 2^{t-h+1} \sigma_h^2 \approx 2^{t-h+1} q^{2(1-k/N_h)} G(\Lambda)$$

That is why we use different preset values of N_h 's in the iterations.

Conclusions

- The key: Find a better bias-dimension trade-off.
- Coding technique is powerful.
 - Efficient computations.
 - Well-controlled noise.
 - A useful tool to analyze the noise level.

Thank You

Thank you for your attention!