Lattice Gaussian Sampling by Markov Chain Monte Carlo: Bounded Distance Decoding and Trapdoor Sampling

Zheng Wang, *Member, IEEE*, and Cong Ling, *Member, IEEE*

***Abstract*— Sampling from the lattice Gaussian distribution plays an important role in various research fields. In this paper, the Markov chain Monte Carlo (MCMC)-based sampling technique is advanced in several fronts. First, the spectral gap for the independent Metropolis-Hastings-Klein (MHK) algorithm is derived, which is then extended to Peikert's algorithm and rejection sampling; we show that independent MHK exhibits faster convergence. Then, the performance of bounded distance decoding (BDD) using MCMC is analyzed, revealing a flexible trade-off between the decoding radius and complexity. MCMC is further applied to trapdoor sampling, again offering a trade-off between security and complexity. Finally, the independent multiple-try Metropolis-Klein (MTMK) algorithm is proposed to enhance the convergence rate. The proposed algorithms allow parallel implementation, which is beneficial for practical applications.**

***Index Terms*— Lattice decoding, lattice Gaussian sampling, Markov chain Monte Carlo, bounded distance decoding, large-scale MIMO detection, trapdoor sampling.**

I. Introduction

NOWADAYS, lattice Gaussian sampling has drawn a lot of attention in various research fields. In mathematics, Banaszczyk was the first to apply it to prove the transference theorems for lattices [1]. In coding, lattice Gaussian distribution was employed to obtain the full shaping gain for lattice coding [2], [3], and to achieve the capacity of the Gaussian channel [4]. It was also used to achieve information-theoretic security in the Gaussian wiretap channel [5], [6] and in the bidirectional relay channel [7], respectively. In cryptography, the lattice Gaussian distribution has become a central tool in the construction of many primitives [8]–[10]. Specifically, lattice Gaussian sampling lies at the core of signature schemes in the Gentry, Peikert and Vaikuntanathan (GPV) paradigm [11]. Furthermore, lattice Gaussian sampling with a suitable variance allows to solve the closest vector problem (CVP) and the shortest vector problem (SVP) [12], [13].

However, in sharp contrast to the continuous Gaussian density, it is by no means trivial even to sample from a low-dimensional discrete Gaussian distribution. For some special lattices, there are rather efficient algorithms for Gaussian sampling [4], [14]. As the default sampling algorithm for general lattices, Klein's algorithm [15] only works when the standard deviation $\sigma = \sqrt{\omega(\log n)}\cdot\max_{1\leq i\leq n}\|\widehat{\mathbf{b}}_i\|$ [11], where $\omega(\log n)$ is a superlogarithmic function, n denotes the lattice dimension and $\widehat{\mathbf{b}}_i$'s are the Gram-Schmidt vectors of the lattice basis $\mathbf{B}$. Peikert gave an efficient lattice Gaussian sampler in [16] for parallel implementation, which however requires larger values of σ. On the other hand, the lattice Gaussian sampling algorithm proposed by Aggarwal *et al.* in [12] and [13] to solve CVP and SVP has a lower bound 2^n on both space and time complexity; it actually obtains samples for small σ by combining original samples for $\sigma = \sqrt{\omega(\log n)}\cdot\max_{1\leq i\leq n}\|\widehat{\mathbf{b}}_i\|$. Although the algorithm in [17] provides a trade-off between (exponential) time and space complexity, its complexity is still too high to be practical.

In order to sample from a target lattice Gaussian distribution with arbitrary $\sigma > 0$, Markov chain Monte Carlo (MCMC) methods were introduced in [18]. In principle, it randomly generates the next Markov state conditioned on the previous one; after the burn-in time, which is normally measured by the *mixing time*, the Markov chain will step into a stationary distribution, when samples from the target distribution can be obtained [19]. It has been demonstrated that Gibbs sampling, which employs univariate conditional sampling to build the chain, yields an ergodic Markov chain [20]. In [18], we proposed an independent Metropolis-Hastings (MH) algorithm incorporating Klein's algorithm (namely, the independent MHK algorithm) to generate a proposal distribution, which is shown to be uniformly ergodic (converging exponentially fast to the stationary distribution). Meanwhile, the associated convergence rate of the Markov chain is derived, resulting

Manuscript received April 9, 2017; revised July 26, 2018; accepted January 17, 2019. Date of publication February 26, 2019; date of current version May 20, 2019. This work was supported in part by the open research fund of the Key Laboratory of Dynamic Cognitive System of Electromagnetic Spectrum Space, Ministry of Industry and Information Technology, Nanjing University of Aeronautics and Astronautics under Grant KF20181913, in part by the open research fund of the National Mobile Communications Research Laboratory, Southeast University under Grant 2019D04, in part by the National Natural Science Foundation of China under Grant 61801216, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20180420. This work was presented in part at the 2016 IEEE International Symposium on Information Theory (ISIT).

Z. Wang is with the Key Laboratory of Dynamic Cognitive System of Electromagnetic Spectrum Space, Ministry of Industry and Information Technology, College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing 21106, China and also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: z.wang@ieee.org).

C. Ling is with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mail: cling@ieee.org).

Communicated by A. G. Dimakis, Associate Editor for Coding Techniques.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2019.2901497

in a tractable estimation of the mixing time. Differently from the algorithms of [12], [13], [17], the independent MHK sampling algorithm only requires polynomial space. In this paper, we advance the state of the art of MCMC-based lattice Gaussian sampling in several fronts.

Firstly, we refine the analysis and extend the independent MHK algorithm of [18]. We obtain the spectral gap of the transition matrix and demonstrate uniformly ergodicity. We extend the independent MH algorithm to a version where Peikert's algorithm [16] is used to generate the proposal distribution. We then compare these MCMC algorithms with rejection sampling from statistics. By deriving their rates of convergence, we show the advantage of the independent MHK. Rejection sampling achieves the same convergence rate only if its normalizing constant is carefully chosen, which is generally rather difficult.

Secondly, we apply the independent MHK algorithm to bounded distance decoding (BDD). BDD is a variant of the CVP where the input is within a certain distance to the lattice. With a careful selection of the standard deviation σ during the sampling process, we improve the result of Klein from $\eta = O(1/n)$ to $\eta = O(\sqrt{\log n}/n)$ in terms of $\eta-$BDD.[1] References [21], [22] achieved a larger value $\eta = O(\sqrt{\log n/n})$, at the expense of a pre-processing stage where Gaussian samples are taken from the dual lattice with standard deviation σ equal to its smoothing parameter. However, sampling at the smoothing parameter is in general a difficult problem with no efficient solutions nowadays. For algorithms of general SVP/CVP such as enumeration and sieving, we refer the readers to the comprehensive survey [23].

Thirdly, we examine the impact of MCMC to trapdoor sampling in the GPV paradigm. In cryptographic applications, the standard deviation σ of the sampler is the main parameter governing the security level. Namely, the smaller σ, the higher security. This is because for a signature system to be secure, it must be hard for an adversary to find lattice points of length about $\sigma\sqrt{n}$. We show that, at moderate costs of increased complexity, MCMC is able to sample with smaller σ, thereby increasing the security level relative to Klein's algorithm [11] and Peikert's algorithm [16].

Finally, to improve the convergence rate of the Markov chain, the independent multiple-try Metropolis-Klein (MTMK) algorithm is proposed, which fully exploits the trial samples generated from the proposal distribution. Uniform ergodicity is demonstrated and the enhanced convergence rate is also given. Since independent MHK is only a special case of independent MTMK, the decoding performance can also be improved due to the usage of trial samples. The proposed sampling algorithm allows a parallel implementation and is easily adopted to MIMO detection to achieve near-optimal performance. With the development of 5G, the demand for large-scale MIMO systems will increase in the next decade, which has triggered research activities towards low complexity decoding algorithms for large-scale MIMO detection [24]–[26]. Therefore, there has been considerable interest in MCMC sampling for the efficient decoding of MIMO systems [27]–[32].

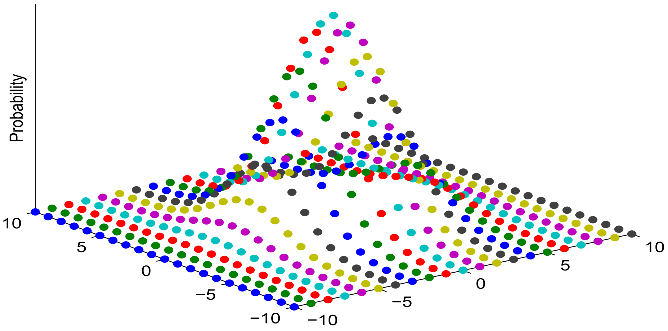

Fig. 1. Illustration of a two-dimensional lattice Gaussian distribution.

The rest of this paper is organized as follows. Section II introduces the lattice Gaussian distribution and briefly reviews the basics of MCMC. In Section III, we derive the spectral gaps of the Markov chains associated with independent MHK and rejection sampling-based lattice Gaussian sampling, and show their uniform ergodicity as well as convergence rates. An extension to Peikert's algorithm is also given. Then, the decoding complexity of BDD using independent MHK algorithm is derived in Section IV. Section V addresses trapdoor sampling using MCMC. In Section VI, the independent MTMK algorithm is proposed to further strength the convergence performance. Simulation results for MIMO detection are presented in Section VII. Finally, Section VIII concludes the paper.

Notation: Matrices and column vectors are denoted by upper and lowercase boldface letters, and the transpose, inverse, pseudoinverse of a matrix $\mathbf{B}$ by $\mathbf{B}^T$, $\mathbf{B}^{-1}$, and $\mathbf{B}^{\dagger}$, respectively. $\mathbf{I}$ denotes the identity matrix. We use $\mathbf{b}_i$ for the ith column of the matrix $\mathbf{B}$, $\widehat{\mathbf{b}}_i$ for the ith Gram-Schmidt vector of the matrix $\mathbf{B}$, $b_{i,j}$ for the entry in the ith row and jth column of the matrix $\mathbf{B}$. A symmetric matrix $\mathbf{B}$ is written as $\mathbf{B} \succ \mathbf{0}$ if it is positive definite. Similarly, we say $\mathbf{B}_1 \succ \mathbf{B}_2$ if $(\mathbf{B}_1 - \mathbf{B}_2) \succ \mathbf{0}$. $\lceil x \rfloor$ denotes rounding to the integer closest to x. If x is a complex number, $\lceil x \rfloor$ rounds the real and imaginary parts separately. In addition, we use the standard *small omega* notation $\omega(\cdot)$, i.e., $|\omega(g(n))| > k \cdot |g(n)|$ for every fixed positive number $k > 0$. Finally, in this paper, the computational complexity is measured by the number of Markov moves.

II. Preliminaries

In this section, we introduce the background and mathematical tools needed to describe and analyze the proposed lattice Gaussian sampling algorithms.

A. Lattice Gaussian Distribution

Let matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \subset \mathbb{R}^n$ consist of n linearly independent column vectors. The n-dimensional lattice Λ generated by $\mathbf{B}$ is defined by

$$\Lambda = \{\mathbf{B}\mathbf{x} : \mathbf{x} \in \mathbb{Z}^n\}, \tag{1}$$

[1] In η-BDD ($\eta < 1/2$), we are given a lattice basis $\mathbf{B}$ and a query point $\mathbf{c}$, and we are asked to find a lattice point within distance $\eta \cdot \lambda_1$ from the target, where λ_1 denotes the first minimum of the lattice.

where $\mathbf{B}$ is called the lattice basis. We define the Gaussian function centered at $\mathbf{c} \in \mathbb{R}^n$ for standard deviation $\sigma > 0$ as

$$\rho_{\sigma,\mathbf{c}}(\mathbf{z}) = e^{-\frac{\|\mathbf{z}-\mathbf{c}\|^2}{2\sigma^2}}, \tag{2}$$

for all $\mathbf{z} \in \mathbb{R}^n$. When $\mathbf{c}$ or σ are not specified, we assume that they are $\mathbf{0}$ and 1 respectively. Then, the *discrete Gaussian distribution* over Λ is defined as

$$D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{Bx})}{\rho_{\sigma,\mathbf{c}}(\Lambda)} = \frac{e^{-\frac{1}{2\sigma^2}\|\mathbf{Bx}-\mathbf{c}\|^2}}{\sum_{\mathbf{x}\in\mathbb{Z}^n} e^{-\frac{1}{2\sigma^2}\|\mathbf{Bx}-\mathbf{c}\|^2}} \tag{3}$$

for all $\mathbf{x} \in \mathbb{Z}^n$, where $\rho_{\sigma,\mathbf{c}}(\Lambda) \triangleq \sum_{\mathbf{Bx}\in\Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{Bx})$ is just a scaling to obtain a probability distribution. We remark that this definition differs slightly from the one in [8], where σ is scaled by a constant factor $\sqrt{2\pi}$ (i.e., $s = \sqrt{2\pi}\sigma$). In fact, the discrete Gaussian resembles a continuous Gaussian distribution, but is only defined over a lattice. It has been shown that discrete and continuous Gaussian distributions share similar properties, if the *flatness factor* is small [5].

B. Decoding by Sampling

Consider the decoding of an $n \times n$ real-valued system. The extension to the complex-valued system is straightforward [33]. Let $\mathbf{x} \in \mathbb{Z}^n$ denote the transmitted signal. The corresponding received signal $\mathbf{c}$ is given by

$$\mathbf{c} = \mathbf{Bx} + \mathbf{w} \tag{4}$$

where $\mathbf{w}$ is the noise vector with zero mean and variance σ_w^2, $\mathbf{B}$ is an $n \times n$ full column-rank matrix of channel coefficients. Typically, the conventional maximum likelihood (ML) reads

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{Z}^n} \|\mathbf{c} - \mathbf{Bx}\|^2 \tag{5}$$

where $\|\cdot\|$ denotes the Euclidean norm. Clearly, ML decoding corresponds to the CVP. If the received signal $\mathbf{c}$ is the origin, then ML decoding reduces to SVP.

Intuitively, the CVP given in (5) can be solved by the lattice Gaussian sampling. Since the distribution is centered at the query point $\mathbf{c}$, the closest lattice point $\mathbf{Bx}$ to $\mathbf{c}$ is assigned the largest sampling probability. Therefore, by multiple samplings, the solution of CVP is the most likely to be returned. It has been demonstrated that lattice Gaussian sampling is equivalent to CVP via a polynomial-time dimension-preserving reduction [34]. Meanwhile, by adjusting the sample size, the sampling decoder enjoys a flexible trade-off between performance and complexity.

In [15], Klein introduced an algorithm which performs sampling from a Gaussian-like distribution (see Algorithm 1). It is shown in [15], [33], and [35] that Klein's algorithm is able to find the closest lattice point when it is close to the input vector: this technique is known as BDD in coding literature, which corresponds to a restricted variant of CVP.

Algorithm 1 Klein's Algorithm

```
Input: $\mathbf{B}, \sigma, \mathbf{c}$
Output: $\mathbf{Bx} \in \Lambda$
let $\mathbf{B} = \mathbf{QR}$ and $\mathbf{c}' = \mathbf{Q}^\dagger \mathbf{c}$
for $i = n, \ldots, 1$ do
    let $\sigma_i = \frac{\sigma}{|r_{i,i}|}$ and $\widetilde{x}_i = \frac{c'_i - \sum_{j=i+1}^n r_{i,j} x_j}{r_{i,i}}$
    sample $x_i$ from $D_{\mathbb{Z},\sigma_i,\widetilde{x}_i}$
end for
return $\mathbf{Bx}$
```

C. Classical MH Algorithms

In [36], the original Metropolis algorithm was extended to a more general scheme known as the Metropolis-Hastings (MH) algorithm. In particular, let us consider a target invariant distribution π together with a proposal distribution $q(\mathbf{x}, \mathbf{y})$. Given the current state $\mathbf{x}$ for Markov chain $\mathbf{X}_t$, a state candidate $\mathbf{y}$ for the next Markov move $\mathbf{X}_{t+1}$ is generated from the proposal distribution $q(\mathbf{x}, \mathbf{y})$. Then the acceptance ratio α is computed by

$$\alpha = \min\left\{1, \frac{\pi(\mathbf{y})q(\mathbf{y}, \mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x}, \mathbf{y})}\right\}, \tag{6}$$

and $\mathbf{y}$ will be accepted as the new state with probability α. Otherwise, $\mathbf{x}$ will be retained. In this way, a Markov chain $\{\mathbf{X}_0, \mathbf{X}_1, \ldots\}$ is established with the transition probability $P(\mathbf{x}, \mathbf{y})$ as follows:

$$P(\mathbf{x}, \mathbf{y}) = \begin{cases} q(\mathbf{x}, \mathbf{y})\alpha & \text{if } \mathbf{y} \neq \mathbf{x}, \\ 1 - \sum_{\mathbf{z}\neq\mathbf{x}} q(\mathbf{x}, \mathbf{z})\alpha & \text{if } \mathbf{y} = \mathbf{x}. \end{cases} \tag{7}$$

It is interesting that in MH algorithms, the proposal distribution $q(\mathbf{x}, \mathbf{y})$ can be any fixed distribution from which we can conveniently draw samples. Therefore, there is large freedom in the choice of $q(\mathbf{x}, \mathbf{y})$ but it is challenging to find a suitable one with satisfactory convergence. In fact, Gibbs sampling can be viewed as a special case of the MH algorithm, whose proposal distribution is a univariate conditional distribution.

As an important parameter to measure the time required by a Markov chain to get close to its stationary distribution, the *mixing time* is defined as [19]

$$t_{\text{mix}}(\epsilon) = \min\{t : \max\|P^t(\mathbf{x}, \cdot) - \pi(\cdot)\|_{TV} \leq \epsilon\}, \tag{8}$$

where $P^t(\mathbf{x}, \cdot)$ denotes a row of the transition matrix $\mathbf{P}$ for t Markov moves and $\|\cdot\|_{TV}$ represents the total variation distance.

D. Independent MHK Algorithm

From the MCMC perspective, lattice Gaussian distribution can be viewed as a complex target distribution lacking direct sampling methods. In order to obtain samples from $D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x})$, the independent MHK sampling was proposed in [18]. Specifically, a state candidate $\mathbf{y}$ for the next Markov move $\mathbf{X}_{t+1}$ is generated by Klein's algorithm, via the following backward one-dimensional conditional sampling (for $i = n, n-1, \ldots, 1$):

$$\begin{aligned} P(y_i|\overline{\mathbf{y}}_{[-i]}) &= P(y_i|y_{i+1}, \ldots, y_n) \\ &= \frac{e^{-\frac{1}{2\sigma^2}\|\overline{\mathbf{c}}' - \overline{\mathbf{R}}\overline{\mathbf{y}}\|^2}}{\sum_{y_i\in\mathbb{Z}} e^{-\frac{1}{2\sigma^2}\|\overline{\mathbf{c}}' - \overline{\mathbf{R}}\overline{\mathbf{y}}\|^2}} \end{aligned}$$

$$
\begin{aligned}
&= \frac{e^{-\frac{1}{2\sigma^2}\|c'_i - \sum_{j=i}^{n} r_{i,j} y_j\|^2}}{\sum_{y_i \in \mathbb{Z}} e^{-\frac{1}{2\sigma^2}\|c'_i - \sum_{j=i}^{n} r_{i,j} y_j\|^2}} \\
&= \frac{e^{-\frac{1}{2\sigma_i^2}\|y_i - \widetilde{y}_i\|^2}}{\sum_{y_i \in \mathbb{Z}} e^{-\frac{1}{2\sigma_i^2}\|y_i - \widetilde{y}_i\|^2}} \\
&= D_{\mathbb{Z},\sigma_i,\widetilde{y}_i}(y_i),
\end{aligned}
\tag{9}
$$

where $\widetilde{y}_i = \frac{c'_i - \sum_{j=i+1}^{n} r_{i,j} y_j}{r_{i,i}}$, $\sigma_i = \frac{\sigma}{|r_{i,i}|}$, $\mathbf{c}' = \mathbf{Q}^{\dagger}\mathbf{c}$ and $\mathbf{B} = \mathbf{QR}$ by QR decomposition with $\|\widehat{\mathbf{b}}_i\| = |r_{i,i}|$. Note that $\overline{\mathbf{y}}_{[-i]} = [y_{i+1}, \ldots, y_n]$, $\overline{\mathbf{R}}$, $\overline{\mathbf{c}}'$ and $\overline{\mathbf{y}}$ are the $(n-i+1)$ segments of $\mathbf{R}$, $\mathbf{c}'$ and $\mathbf{y}$ respectively (i.e., $\overline{\mathbf{R}}$ is a $(n-i+1) \times (n-i+1)$ submatrix of $\mathbf{R}$ with $r_{i,i}$ to $r_{n,n}$ in the diagonal).

Given the current state $\mathbf{x}$, the proposal distribution $q(\mathbf{x}, \mathbf{y})$ in the independent MHK sampling is given by

$$
\begin{aligned}
q(\mathbf{x}, \mathbf{y}) &= \prod_{i=1}^{n} P(y_{n+1-i} | \overline{\mathbf{y}}_{[-(n+1-i)]}) \\
&= \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{By})}{\prod_{i=1}^{n} \rho_{\sigma_{n+1-i},\widetilde{y}_{n+1-i}}(\mathbb{Z})} \\
&= q(\mathbf{y}),
\end{aligned}
\tag{10}
$$

where the proposal distribution $q(\mathbf{x}, \mathbf{y})$ is actually independent of $\mathbf{x}$. Therefore, the connection between two consecutive Markov moves is only due to the decision stage.

With the state candidate $\mathbf{y}$, the acceptance ratio α is obtained by substituting (10) into (6)

$$
\alpha = \min\left\{1, \frac{\prod_{i=1}^{n} \rho_{\sigma_{n+1-i},\widetilde{y}_{n+1-i}}(\mathbb{Z})}{\prod_{i=1}^{n} \rho_{\sigma_{n+1-i},\widetilde{x}_{n+1-i}}(\mathbb{Z})}\right\},
\tag{11}
$$

where $\widetilde{x}_i = \frac{c'_i - \sum_{j=i+1}^{n} r_{i,j} x_j}{r_{i,i}}$ and we note that $\pi = D_{\Lambda,\sigma,\mathbf{c}}$ in (6) (these notations will be followed throughput the context). The sampling procedure is summarized in Algorithm 2. Note that the initial state $\mathbf{x}_0$ for $\mathbf{X}_0$ can be chosen from $\mathbb{Z}^n$ arbitrarily or from the output of a suboptimal algorithm.

Thanks to the celebrated coupling technique, the uniformly ergodicity was demonstrated in [18]. Nevertheless, the spectral gap of the transition matrix, which serves as an important metric for the mixing time of the underlying Markov chain, has not been determined yet.

III. Convergence Analysis

In this section, the spectrum of the Markov chain induced by independent MHK sampling is analyzed, followed by the extensions to Peikert's algorithm and rejection sampling. As a common way to evaluate the mixing time, the *spectral gap* $\gamma = 1 - |\tau_1| > 0$ of the transition matrix is preferred for convergence analysis in MCMC [19]. Here, τ_1 represents the second largest eigenvalue in magnitude of the transition matrix $\mathbf{P}$ [37].

A. Spectral Gap of Independent MHK Algorithm

Theorem 1: Given the invariant lattice Gaussian distribution $D_{\Lambda,\sigma,\mathbf{c}}$, the Markov chain induced by independent MHK sampling exhibits a spectral gap

$$
\gamma \geq \delta = \frac{\rho_{\sigma,\mathbf{c}}(\Lambda)}{\prod_{i=1}^{n} \rho_{\sigma_i}(\mathbb{Z})}.
\tag{12}
$$

Algorithm 2 Independent MHK Sampling Algorithm

```
Input: B, $\sigma$, c, $\mathbf{x}_0$, $t_{\mathrm{mix}}(\epsilon)$;
Output: $\mathbf{x} \sim D_{\Lambda,\sigma,\mathbf{c}}$;
let $\mathbf{X}_0 = \mathbf{x}_0$
for $t$ =1,2, …,  do
    let x denote the state of $\mathbf{X}_{t-1}$
    sample y from the proposal distribution $q(\mathbf{x}, \mathbf{y})$ in (10)
    calculate the acceptance ratio $\alpha(\mathbf{x}, \mathbf{y})$ in (11)
    generate a sample $u$ from the uniform density $U[0, 1]$
    if $u \leq \alpha(\mathbf{x}, \mathbf{y})$ then
        let $\mathbf{X}_t = \mathbf{y}$
    else
        $\mathbf{X}_t = \mathbf{x}$
    end if
    if $t \geq t_{\mathrm{mix}}(\epsilon)$ then
        output x
    end if
end for
```

Proof: From (10) and (11), the transition probability $P(\mathbf{x}, \mathbf{y})$ of each Markov move in the independent MHK sampling is given by

$$
P(\mathbf{x}, \mathbf{y}) = \begin{cases} \min\left\{q(\mathbf{y}), \frac{\pi(\mathbf{y})q(\mathbf{x})}{\pi(\mathbf{x})}\right\} & \text{if } \mathbf{y} \neq \mathbf{x}, \\ 1 - \sum\limits_{\mathbf{z} \neq \mathbf{x}} \min\left\{q(\mathbf{z}), \frac{\pi(\mathbf{z})q(\mathbf{x})}{\pi(\mathbf{x})}\right\} & \text{if } \mathbf{y} = \mathbf{x}. \end{cases}
\tag{13}
$$

For notational simplicity, we define the *importance weight* $w(\mathbf{x})$ as

$$
w(\mathbf{x}) = \frac{\pi(\mathbf{x})}{q(\mathbf{x})}.
\tag{14}
$$

Then the transition probability can be rewritten as

$$
P(\mathbf{x}, \mathbf{y}) = \begin{cases} q(\mathbf{y}) \cdot \min\left\{1, \frac{w(\mathbf{y})}{w(\mathbf{x})}\right\} & \text{if } \mathbf{y} \neq \mathbf{x}, \\ q(\mathbf{x}) + \sum\limits_{\mathbf{z} \neq \mathbf{x}} q(\mathbf{z}) \cdot \max\left\{0, 1 - \frac{w(\mathbf{z})}{w(\mathbf{x})}\right\} & \text{if } \mathbf{y} = \mathbf{x}. \end{cases}
\tag{15}
$$

Without loss of generality, we label the countably infinite state space $\Omega = \mathbb{Z}^n$ as $\Omega = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\infty\}$ and assume that these states are sorted according to their importance weights, namely,

$$
w(\mathbf{x}_1) \geq w(\mathbf{x}_2) \geq \cdots \geq w(\mathbf{x}_\infty).
\tag{16}
$$

From (15) and (16), the transition matrix $\mathbf{P}$ of the Markov chain can be exactly expressed as

$$
\mathbf{P} = \begin{bmatrix} q(\mathbf{x}_1) + \eta_1 & \frac{\pi(\mathbf{x}_2)}{w(\mathbf{x}_1)} & \frac{\pi(\mathbf{x}_3)}{w(\mathbf{x}_1)} & \cdots & \frac{\pi(\mathbf{x}_\infty)}{w(\mathbf{x}_1)} \\ q(\mathbf{x}_1) & q(\mathbf{x}_2) + \eta_2 & \frac{\pi(\mathbf{x}_3)}{w(\mathbf{x}_2)} & \cdots & \frac{\pi(\mathbf{x}_\infty)}{w(\mathbf{x}_2)} \\ q(\mathbf{x}_1) & q(\mathbf{x}_2) & q(\mathbf{x}_3) + \eta_3 & \cdots & \frac{\pi(\mathbf{x}_\infty)}{w(\mathbf{x}_3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ q(\mathbf{x}_1) & q(\mathbf{x}_2) & q(\mathbf{x}_3) & \cdots & q(\mathbf{x}_\infty) \end{bmatrix}
$$

where

$$
\eta_j = \sum_{i=j}^{\infty} \left(q(\mathbf{x}_i) - \frac{\pi(\mathbf{x}_i)}{w(\mathbf{x}_j)} \right)
\tag{17}
$$

stands for the probability of being rejected in the decision stage with the current state $\mathbf{x}_j$ for $\mathbf{X}_t$.

Let $\mathbf{q} = [q(\mathbf{x}_1), q(\mathbf{x}_2), \ldots]^T$ denote the vector of proposal probabilities. Then by decomposition, it follows that

$$\mathbf{P} = \mathbf{G} + \mathbf{e}\mathbf{q}^T, \tag{18}$$

where $\mathbf{e} = [1, 1, \ldots]^T$ and $\mathbf{G}$ is an upper triangular matrix of the form

$$\mathbf{G} = \begin{bmatrix} \eta_1 & \frac{\pi(\mathbf{x}_2)}{w(\mathbf{x}_1)} - q(\mathbf{x}_2) & \cdots & \frac{\pi(\mathbf{x}_\infty)}{w(\mathbf{x}_1)} - q(\mathbf{x}_\infty) \\ 0 & \eta_2 & \cdots & \frac{\pi(\mathbf{x}_\infty)}{w(\mathbf{x}_2)} - q(\mathbf{x}_\infty) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}.$$

It is well-known that for a Markov chain, the largest eigenvalue of the transition matrix $\mathbf{P}$ always equals 1. Here, as $\mathbf{e}$ is a common right eigenvector for both $\mathbf{P}$ and $\mathbf{P} - \mathbf{G}$, it naturally corresponds to the largest eigenvalue 1. Meanwhile, since the rank of $\mathbf{P} - \mathbf{G}$ is 1, the other eigenvalues of $\mathbf{G}$ are exactly the same as those of $\mathbf{P}$.

Thanks to the ascending order in (16), it is easy to verify that the spectral radius τ_1 is exactly given by

$$\tau_1 = \eta_1 \tag{19}$$

and

$$1 > |\eta_1| \geq |\eta_2| \geq \cdots > 0, \tag{20}$$

thereby raising the interest of identifying the value of τ_1.

Therefore, according to (17) and (19), we can easily get that

$$\begin{aligned} \tau_1 &= \sum_{i=1}^{\infty} \left(q(\mathbf{x}_i) - \frac{\pi(\mathbf{x}_i)}{w(\mathbf{x}_1)} \right) \\ &= \sum_{i=1}^{\infty} q(\mathbf{x}_i) - \frac{1}{w(\mathbf{x}_1)} \cdot \sum_{i=1}^{\infty} \pi(\mathbf{x}_i) \\ &= 1 - \frac{1}{w(\mathbf{x}_1)} \\ &= 1 - \frac{q(\mathbf{x}_1)}{\pi(\mathbf{x}_1)}. \end{aligned} \tag{21}$$

In other words, the spectral gap $1 - \tau_1$ is exactly captured by the ratio $q(\mathbf{x}_1)/\pi(\mathbf{x}_1)$. Next, we invoke the following Lemma to lower bound the ratio $q(\mathbf{x})/\pi(\mathbf{x})$ for $\mathbf{x} \in \mathbb{Z}^n$.

Lemma 1 ([18]): In the independent MHK algorithm

$$\frac{q(\mathbf{x})}{\pi(\mathbf{x})} \geq \delta \tag{22}$$

for all $\mathbf{x} \in \mathbb{Z}^n$, where δ is defined as

$$\delta \triangleq \frac{\rho_{\sigma,\mathbf{c}}(\Lambda)}{\prod_{i=1}^{n} \rho_{\sigma_i}(\mathbb{Z})}. \tag{23}$$

The proof is completed by combining (21) and (22). ■

By using the coupling technique, it is shown in [18] that the Markov chain converges exponentially fast to the stationary distribution in total variational distance:

$$\| P^t(\mathbf{x}, \cdot) - D_{\Lambda,\sigma,\mathbf{c}}(\cdot) \|_{TV} \leq (1 - \delta)^t, \tag{24}$$

The mixing time of the Markov chain is given by

$$t_{\text{mix}}(\epsilon) = \frac{\ln \epsilon}{\ln(1 - \delta)} < (-\ln \epsilon) \cdot \left(\frac{1}{\delta} \right), \quad \epsilon < 1 \tag{25}$$

which is proportional to $1/\delta$, and becomes $O(1)$ if $\delta \to 1$.

B. Extension to Peikert's Algorithm

Klein's sampling algorithm is a randomized variant of Babai's nearest-plane algorithm for lattice decoding [38]. Babai also proposed a simpler decoding scheme by direct rounding,[2] which was further randomized by Peikert in [16]. Although Peikert's algorithm requires a higher value of σ, it is parallelizable and can be more attractive in practical implementation. In fact, Peikert's algorithm can also be incorporated into the Metropolis-Hastings algorithm to overcome the limitation of σ.

Specifically, given the standard deviation $\sigma > 0$ and a basis $\mathbf{B}$, one chooses a positive definite matrix $\Sigma_1 = r^2 \cdot \mathbf{B}\mathbf{B}^T \prec \Sigma = \sigma^2 \cdot \mathbf{I}$ for $r > 0$ (i.e., $\Sigma_2 = \Sigma - \Sigma_1$ is positive definite). Then, the proposed sample $\mathbf{z} \in \Lambda$ is taken from the distribution $\mathbf{c} + \mathbf{z}' + D_{\Lambda - \mathbf{c} - \mathbf{z}', \sqrt{\Sigma_1}}$, where $\mathbf{z}' \in \mathbb{R}^n$ is sampled from the continuous distribution $D_{\sqrt{\Sigma_2}}$. Note the lattice Gaussian distribution $D_{\Lambda - \mathbf{c} - \mathbf{z}', \sqrt{\Sigma_1}}$ is expressed as

$$D_{\Lambda - \mathbf{c} - \mathbf{z}', \sqrt{\Sigma_1}}(\mathbf{B}\mathbf{x}) = \frac{\rho_{\sqrt{\Sigma_1}}(\mathbf{B}\mathbf{x} - \mathbf{c} - \mathbf{z}')}{\rho_{\sqrt{\Sigma_1}}(\Lambda - \mathbf{c} - \mathbf{z}')} \tag{26}$$

with

$$\rho_{\sqrt{\Sigma_1}}(\mathbf{y}) = e^{-\frac{1}{2}\mathbf{y}^T \Sigma_1^{-1} \mathbf{y}}, \quad \mathbf{y} \in \mathbb{R}^n. \tag{27}$$

The joint probability distribution of $\mathbf{z} \in \Lambda$ and $\mathbf{z}' \in \mathbb{R}^n$ is given by

$$\begin{aligned} P(\mathbf{z}, \mathbf{z}') &= D_{\Lambda - \mathbf{c} - \mathbf{z}', \sqrt{\Sigma_1}}(\mathbf{z} - \mathbf{c} - \mathbf{z}') \cdot D_{\sqrt{\Sigma_2}}(\mathbf{z}') \\ &= \frac{\rho_{\sqrt{\Sigma_1}}(\mathbf{z} - \mathbf{c} - \mathbf{z}')}{\rho_{\sqrt{\Sigma_1}}(\Lambda - \mathbf{c} - \mathbf{z}')} \cdot \frac{\rho_{\sqrt{\Sigma_2}}(\mathbf{z}')}{\sqrt{\det(2\pi \Sigma_2)}} \\ &\overset{(a)}{=} \frac{\rho_{\sqrt{\Sigma_1}}(\mathbf{z}' - \mathbf{z} + \mathbf{c})}{\rho_{\sqrt{\Sigma_1}}(\Lambda - \mathbf{c} - \mathbf{z}')} \cdot \frac{\rho_{\sqrt{\Sigma_2}}(\mathbf{z}')}{\sqrt{\det(2\pi \Sigma_2)}} \\ &\overset{(b)}{=} \frac{\rho_{\sqrt{\Sigma}}(\mathbf{z} - \mathbf{c}) \cdot \rho_{\sqrt{\Sigma_3}}(\mathbf{z}' - \mathbf{c}')}{\rho_{\sqrt{\Sigma_1}}(\Lambda - \mathbf{c} - \mathbf{z}') \cdot \sqrt{\det(2\pi \Sigma_2)}}, \end{aligned} \tag{28}$$

where (a) is due to the symmetry of $\rho_{\sqrt{\Sigma_1}}$, and (b) follows from [16, Fact 2.1] with positive definite matrix $\Sigma_3^{-1} = \Sigma_1^{-1} + \Sigma_2^{-1}$ and $\mathbf{c}' = \Sigma_3 \Sigma_1^{-1}(\mathbf{z} - \mathbf{c})$. Consequently, the marginal distribution of $\mathbf{z}$ is

$$P(\mathbf{z}) = \frac{\rho_{\sqrt{\Sigma}}(\mathbf{z} - \mathbf{c})}{\sqrt{\det(2\pi \Sigma_2)}} \cdot \int \frac{\rho_{\sqrt{\Sigma_3}}(\mathbf{z}' - \mathbf{c}')}{\rho_{\sqrt{\Sigma_1}}(\Lambda - \mathbf{c} - \mathbf{z}')} d\mathbf{z}'. \tag{29}$$

As $\mathbf{z} = \mathbf{B}\mathbf{x}$ for $\mathbf{x} \in \mathbb{Z}^n$, we have

$$P(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{B}\mathbf{x})}{\sqrt{\det(2\pi \Sigma_2)}} \cdot \int \frac{\rho_{\sqrt{\Sigma_3}}(\mathbf{z}' - \mathbf{c}')}{\rho_{\sqrt{\Sigma_1}}(\Lambda - \mathbf{c} - \mathbf{z}')} d\mathbf{z}'. \tag{30}$$

[2] In communications, Babai's nearest-plane algorithm is known as successive interference cancelation (SIC) while the direct rounding algorithm is referred to as zero-forcing (ZF).

Clearly, $P(\cdot)$ can be used as a proposal distribution $q(\cdot)$ in the MH algorithm to obtain the state candidate $\mathbf{y} \in \mathbb{Z}^n$. In this case, the acceptance ratio α can be calculated by

$$\alpha = \min\left\{1, \frac{\pi(\mathbf{y})P(\mathbf{x})}{\pi(\mathbf{x})P(\mathbf{y})}\right\}, \tag{31}$$

followed by a decision to accept $\mathbf{X}_{t+1} = \mathbf{y}$ or not. To summarize, its operation procedure is shown in Algorithm 3.

Lemma 2: In the independent MH algorithm using Peikert's algorithm, there exists a constant $\delta' > 0$ such that

$$\frac{q(\mathbf{x})}{\pi(\mathbf{x})} \geq \delta' \tag{32}$$

for all $\mathbf{x} \in \mathbb{Z}^n$, where

$$\delta' = \frac{\rho_{\sigma,\mathbf{c}}(\Lambda)}{\rho_r(\mathbb{Z}^n)} \cdot \frac{r^n}{\sigma^n} \cdot |\det(\mathbf{B})|. \tag{33}$$

Proof: To start with, we have

$$\begin{aligned}\frac{q(\mathbf{x})}{\pi(\mathbf{x})} &= \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{Bx})}{\sqrt{\det(2\pi\Sigma_2)}} \cdot \int \frac{\rho_{\sqrt{\Sigma_3}}(\mathbf{z}'-\mathbf{c}')}{\rho_{\sqrt{\Sigma_1}}(\Lambda-\mathbf{c}-\mathbf{z}')} d\mathbf{z}' \cdot \frac{\rho_{\sigma,\mathbf{c}}(\Lambda)}{\rho_{\sigma,\mathbf{c}}(\mathbf{Bx})} \\ &\overset{(c)}{\geq} \frac{\rho_{\sigma,\mathbf{c}}(\Lambda)}{\sqrt{\det(2\pi\Sigma_2)}} \cdot \frac{1}{\rho_{\sqrt{\Sigma_1}}(\Lambda)} \cdot \int \rho_{\sqrt{\Sigma_3}}(\mathbf{z}'-\mathbf{c}') d\mathbf{z}' \\ &= \frac{\rho_{\sigma,\mathbf{c}}(\Lambda)}{\rho_{\sqrt{\Sigma_1}}(\Lambda)} \cdot \frac{\sqrt{\det(\Sigma_3)}}{\sqrt{\det(\Sigma_2)}} \\ &= \frac{\rho_{\sigma,\mathbf{c}}(\Lambda)}{\rho_r(\mathbb{Z}^n)} \cdot \frac{\sqrt{\det(\Sigma_3)}}{\sqrt{\det(\Sigma_2)}},\end{aligned} \tag{34}$$

where inequality (c) comes from the fact that $\rho_{\sqrt{\Sigma}}(\Lambda - \mathbf{c}) \leq \rho_{\sqrt{\Sigma}}(\Lambda)$.

The Lemma is proven by showing that

$$\begin{aligned}\frac{\sqrt{\det(\Sigma_3)}}{\sqrt{\det(\Sigma_2)}} &\overset{(d)}{=} \sqrt{\det(\Sigma_3)} \cdot \sqrt{\det(\Sigma_2^{-1})} \\ &\overset{(e)}{=} \sqrt{\det(\Sigma_3\Sigma_2^{-1})} \\ &= \sqrt{\det(\Sigma\Sigma_1^{-1})} \\ &= \sqrt{\det\left(\frac{r^2}{\sigma^2} \cdot \mathbf{BB}^T\right)} \\ &= \frac{r^n}{\sigma^n} \cdot |\det(\mathbf{B})|.\end{aligned} \tag{35}$$

Here, (d) and (e) follow from the properties of determinant that

$$\frac{1}{\det(\mathbf{A})} = \det(\mathbf{A}^{-1}) \tag{36}$$

and

$$\det(\mathbf{A})\det(\mathbf{B}) = \det(\mathbf{AB}), \tag{37}$$

respectively, for square matrices $\mathbf{A}$ and $\mathbf{B}$ of equal sizes. ■

To satisfy the condition that $\sigma^2\mathbf{I} \succ r^2 \cdot \mathbf{BB}^T$, we require

$$\sigma > rs_1(\mathbf{B}), \tag{38}$$

where $s_1(\mathbf{B})$ denotes the largest singular value of the basis $\mathbf{B}$. It is readily verified that

$$s_1(\mathbf{B}) \geq \max_{1\leq i\leq n} \|\mathbf{b}_i\| \geq \max_{1\leq i\leq n} \|\widehat{\mathbf{b}}_i\|. \tag{39}$$

Algorithm 3 Independent MH Sampling Using Peikert's Algorithm

```
Input: $\mathbf{B}, \sigma, \mathbf{c}, \mathbf{x}_0, t_{\text{mix}}(\epsilon), \Sigma > \Sigma_1 = r^2 \cdot \mathbf{B} \cdot \mathbf{B}^T$;
Output: $\mathbf{x} \sim D_{\Lambda,\sigma,\mathbf{c}}$;
let $\mathbf{X}_0 = \mathbf{x}_0$
for $t$ =1,2, …, do
    let $\mathbf{x}$ denote the state of $\mathbf{X}_{t-1}$
    sample $\mathbf{y}$ from the proposal distribution $q(\mathbf{y})$ in (30)
    calculate the acceptance ratio $\alpha_s(\mathbf{x}, \mathbf{y})$ in (31)
    generate a sample $u$ from the uniform density $U[0, 1]$
    if $u \leq \alpha(\mathbf{x}, \mathbf{y})$ then
        let $\mathbf{X}_t = \mathbf{y}$
    else
        $\mathbf{X}_t = \mathbf{x}$
    end if
    if $t \geq t_{\text{mix}}(\epsilon)$ then
        output $\mathbf{x}$
    end if
end for
```

Lemma 3: For independent MH samplings based on Peikert's algorithm and on Klein's algorithm, the following relation holds:

$$\delta' \leq \delta. \tag{40}$$

Proof: According to (12) and (33), in order to show $\delta' \leq \delta$, we need to prove that

$$\rho_r(\mathbb{Z}^n) \cdot \frac{\sigma^n}{r^n} \cdot \frac{1}{|\det(\mathbf{B})|} \geq \prod_{i=1}^{n} \rho_{\sigma_i}(\mathbb{Z}). \tag{41}$$

Next, by recalling the *Jacobi theta function* $\vartheta_3(\tau) = \sum_{n=-\infty}^{+\infty} e^{-\pi\tau n^2}$ with $\tau > 0$, we have

$$\rho_r(\mathbb{Z}) = \vartheta_3\left(\frac{1}{2\pi r^2}\right) \tag{42}$$

and the left-hand side of (41)

$$\begin{aligned}&= \frac{1}{(\sqrt{2\pi}r)^n}\vartheta_3^n\left(\frac{1}{2\pi r^2}\right) \cdot (\sqrt{2\pi})^n \cdot \frac{\sigma^n}{|\det(\mathbf{B})|} \\ &\overset{(f)}{=} \vartheta_3^n(2\pi r^2) \cdot (\sqrt{2\pi})^n \cdot \prod_{i=1}^{n} \sigma_i,\end{aligned} \tag{43}$$

where (f) utilizes the symmetry property of *Theta series* for isodual lattice $\mathbb{Z}$

$$\vartheta_3\left(\frac{1}{\tau^2}\right) = \tau\vartheta_3(\tau^2). \tag{44}$$

Moreover, as $\vartheta_3(\tau)$ is monotone decreasing with τ, the following relation holds:

$$\begin{aligned}\vartheta_3(2\pi r^2) &\geq \vartheta_3\left(2\pi r^2 \frac{s_1^2(\mathbf{B})}{\|\widehat{\mathbf{b}}_i\|^2}\right) \\ &\geq \vartheta_3\left(2\pi \frac{\sigma^2}{\|\widehat{\mathbf{b}}_i\|^2}\right) \\ &= \vartheta_3(2\pi\sigma_i^2)\end{aligned} \tag{45}$$

due to (38) and (39).

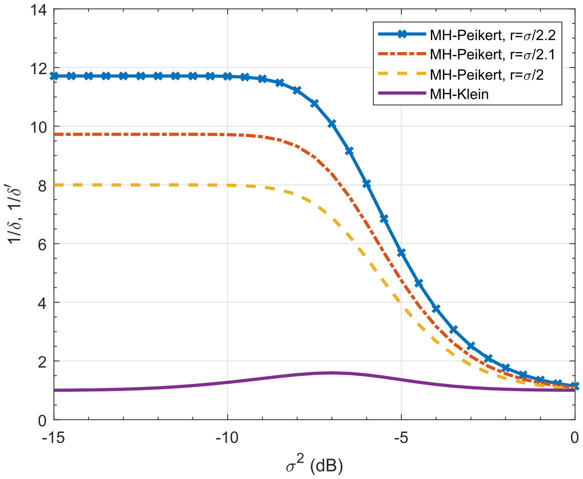

Fig. 2. Comparison of $1/\delta$ and $1/\delta'$ for independent MH samplings based on Klein's and Peikert's algorithms for lattice D_4 with $\sigma^2 = -8$dB and $\mathbf{c} = \mathbf{0}$.

Hence, we finally have that the left-hand side of (41)

$$\begin{aligned} &\geq \prod_{i=1}^{n} (\sqrt{2\pi}\sigma_i) \cdot \vartheta_3(2\pi\sigma_i^2) \\ &= \prod_{i=1}^{n} \vartheta_3\left(\frac{1}{2\pi\sigma_i^2}\right) \\ &= \prod_{i=1}^{n} \rho_{\sigma_i}(\mathbb{Z}), \end{aligned} \tag{46}$$

thus completing the proof. ■

Similarly to independent MHK, it is easy to verify that the proposed algorithm is also uniformly ergodic.

Theorem 2: Given $D_{\Lambda,\sigma,\mathbf{c}}$, the Markov chain induced by independent MH sampling using Peikert's algorithm converges exponentially fast:

$$\|P^t(\mathbf{x},\cdot) - D_{\Lambda,\sigma,\mathbf{c}}(\cdot)\|_{TV} \leq (1-\delta')^t. \tag{47}$$

By Lemma 3, we can see that the independent MH sampling based on Peikert's algorithm converges slower than that based on Klein's algorithm. This is numerically confirmed in Fig. 2 for checkerboard lattice D_4, where a comparison of the coefficients $1/\delta$ and $1/\delta'$ is given. Clearly, in the whole range of r, the independent MH-Peikert sampling requires more iterations than independent MHK.

C. Extension to Rejection Sampling

The classic rejection sampling is able to generate independent samples from the target distribution, but requires a normalizing constant for the application of a proposal distribution [39]. Given the target distribution $\pi(\mathbf{x}) = D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x})$, its operation consists of the following three steps:

1) *Generate a candidate sample* $\mathbf{y}$ *from distribution* $q(\mathbf{y})$ *using Klein's algorithm or Peikert's algorithm.*

2) *Calculate a normalizing constant* ω_0 *such that*

$$\omega_0 \cdot q(\mathbf{x}) \geq \pi(\mathbf{x}) \tag{48}$$

for all $\mathbf{x} \in \mathbb{Z}^n$.

3) *Output* $\mathbf{y}$ *with probability*

$$\alpha = \frac{\pi(\mathbf{y})}{\omega_0 \cdot q(\mathbf{y})} = \frac{\omega(\mathbf{y})}{\omega_0} \tag{49}$$

and otherwise repeat.

Generally, rejection sampling is not directly comparable with MCMC sampling as it requires the normalizing constant ω_0 for calibrating, which is not realistic in many cases of interest. Nevertheless, with a certain choice of ω_0, it is possible to interpret it as a particular MCMC algorithm.

Definition 1: Given the target distribution $\pi(\mathbf{x}) = D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x})$, the Markov chain arising from the above rejection sampler with $\omega_0 \geq \pi(\mathbf{x})/q(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{Z}^n$ is reversible, irreducible and aperiodic, with transition probability

$$P(\mathbf{x},\mathbf{y}) = \begin{cases} q(\mathbf{y}) \cdot \frac{\omega(\mathbf{y})}{\omega_0} & \text{if } \mathbf{y} \neq \mathbf{x}, \\ 1 - \sum\limits_{\mathbf{z}\neq\mathbf{x}} q(\mathbf{z}) \cdot \frac{\omega(\mathbf{z})}{\omega_0} & \text{if } \mathbf{y} = \mathbf{x}. \end{cases} \tag{50}$$

Clearly, the algorithm based on rejection sampling converges when the first acceptance takes place. The samples after the acceptance are naturally independently and identically distributed (i.i.d.). Similarly to the setting in (16), the transition matrix $\mathbf{P}_\mathrm{r}$ of this Markov chain is exactly given by

$$\mathbf{P}_\mathrm{r} = \begin{bmatrix} \frac{\pi_1}{\omega_0} + (1 - \frac{1}{\omega_0}) & \frac{\pi_2}{\omega_0} & \frac{\pi_3}{\omega_0} & \cdots & \frac{\pi_\infty}{\omega_0} \\ \frac{\pi_1}{\omega_0} & \frac{\pi_2}{\omega_0} + (1 - \frac{1}{\omega_0}) & \frac{\pi_3}{\omega_0} & \cdots & \frac{\pi_\infty}{\omega_0} \\ \frac{\pi_1}{\omega_0} & \frac{\pi_2}{\omega_0} & \frac{\pi_3}{\omega_0} + (1 - \frac{1}{\omega_0}) & \cdots & \frac{\pi_\infty}{\omega_0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\pi_1}{\omega_0} & \frac{\pi_2}{\omega_0} & \frac{\pi_3}{\omega_0} & \cdots & \frac{\pi_\infty}{\omega_0} + (1 - \frac{1}{\omega_0}) \end{bmatrix}$$

which can be further decomposed into

$$\mathbf{P}_\mathrm{r} = \mathbf{P}_\mathrm{r}(\cdot,\cdot|\text{accept}) \cdot P_\text{accept} + \mathbf{P}_\mathrm{r}(\cdot,\cdot|\text{reject}) \cdot P_\text{reject} \tag{51}$$

where

$$\mathbf{P}_\mathrm{r}(\cdot,\cdot|\text{accept}) = \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 & \cdots & \pi_\infty \\ \pi_1 & \pi_2 & \pi_3 & \cdots & \pi_\infty \\ \pi_1 & \pi_2 & \pi_3 & \cdots & \pi_\infty \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \pi_1 & \pi_2 & \pi_3 & \cdots & \pi_\infty \end{bmatrix} \tag{52}$$

$$\mathbf{P}_\mathrm{r}(\cdot,\cdot|\text{reject}) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{53}$$

and

$$\begin{cases} P_\text{accept} = 1/\omega_0, \\ P_\text{reject} = 1 - 1/\omega_0. \end{cases} \tag{54}$$

Here, P_accept and P_reject denote the acceptance and rejection probabilities of a new candidate in the next move.

Similarly to the analysis of independent MHK, we have the following Lemma, whose proof is omitted due to simplicity.

Lemma 4: The eigenvalues η_i's of the transition matrix $\mathbf{P}_\mathrm{r}$ satisfy that

$$1 > |\eta_1| = |\eta_2| = \cdots > 0 \tag{55}$$

with

$$\eta_i = 1 - \frac{1}{\omega_0} \tag{56}$$

for $i = 1, \ldots, \infty$.

Furthermore, we arrive at the following Theorem.

Theorem 3: Given the invariant lattice Gaussian distribution $\pi = D_{\Lambda,\sigma,\mathbf{c}}$, the Markov chain induced by rejection sampling converges exponentially fast as

$$\|P^t(\mathbf{x},\cdot) - D_{\Lambda,\sigma,\mathbf{c}}(\cdot)\|_{TV} = (1-\pi(\mathbf{x}))\cdot(\tau_1)^t, \tag{57}$$

where the spectral radius $\tau_1 = \eta_1 = 1 - \frac{1}{\omega_0}$.

Proof: Let A_t denote the number of acceptances during consecutive t moves. Then

$$\begin{aligned}
\|P^t(\mathbf{x},\cdot) - D_{\Lambda,\sigma,\mathbf{c}}(\cdot)\|_{TV} &= \|P^t(\mathbf{x},\cdot|A_t=0)\cdot P(A_t=0) + \\
&\quad P^t(\mathbf{x},\cdot|A_t>0)\cdot P(A_t>0) - D_{\Lambda,\sigma,\mathbf{c}}(\cdot)\|_{TV}\\
&= \left\| P^t(\mathbf{x},\cdot|A_t=0)\cdot\left(1-\frac{1}{\omega_0}\right)^t - D_{\Lambda,\sigma,\mathbf{c}}\cdot\left(1-\frac{1}{\omega_0}\right)^t\right\|_{TV}\\
&= \left\|[P^t(\mathbf{x},\cdot|A_t=0) - D_{\Lambda,\sigma,\mathbf{c}}]\cdot\left(1-\frac{1}{\omega_0}\right)^t\right\|_{TV}\\
&= (1-D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}))\cdot\left(1-\frac{1}{\omega_0}\right)^t\\
&= (1-\pi(\mathbf{x}))\cdot\tau_1^t,
\end{aligned}$$

where $P(A_t=0) = (1-1/\omega_0)^t$, $P(A_t>0) = 1-(1-1/\omega_0)^t$, and $P(\mathbf{x},\cdot|A_t>0)$ has converged to $D_{\Lambda,\sigma,\mathbf{c}}$ after the first acceptance. ■

According to Theorem 3, the convergence rate of rejection sampling depends on the choice of the normalizing constant ω_0. Because $\omega_0 \geq \pi(\mathbf{x})/q(\mathbf{x})$ for all $\mathbf{x}\in\mathbb{Z}^n$, the spectral radius $\tau_1 = \eta_1$ of rejection sampling achieves the minimum when $\omega_0 = \omega_{\max}(\mathbf{x}) = \omega(\mathbf{x}_1)$, namely,

$$\tau_1 = 1 - \frac{1}{\omega(\mathbf{x}_1)} \leq 1-\delta, \tag{58}$$

thus leading to

$$\|P^t(\mathbf{x},\cdot) - D_{\Lambda,\sigma,\mathbf{c}}(\cdot)\|_{TV} \leq (1-\pi(\mathbf{x}))\cdot(1-\delta)^t. \tag{59}$$

From (24) and (59), it is worth noting that only when $\omega_0 = \omega(\mathbf{x}_1)$, rejection sampling and independent MH have the same convergence rate. However, the former requires the knowledge of ω_0 while the latter does not.

Remark 1: Another algorithm for lattice Gaussian sampling based on rejection sampling was proposed in [40]. However, it was only concerned with values of σ required by Klein's algorithm. Its goal is to use rejection sampling to produce exact Gaussian samples, since Klein's algorithm only approximates the target distribution. In contrast, our goal is to sample with smaller values of σ. The algorithm of [40] computes a certain normalizing constant in polynomial time and needs just a few steps on average to produce an exact sample. It is possible to extend their algorithm to smaller values of σ, but its running time will blow up.

Algorithm 4 BDD using Independent MHK Sampling

```
Input: $\mathbf{B}, \sigma, \mathbf{c}, \mathbf{x}_0, t$;
Output: $\widehat{\mathbf{x}}$;
let $\widehat{\mathbf{x}} = \mathbf{x}_0$ and $\mathbf{X}_0 = \mathbf{x}_0$
for $i$ =1, ..., $t$ do
    let $\mathbf{x}$ denote the state of $\mathbf{X}_{t-1}$
    sample $\mathbf{y}$ from the proposal distribution $q(\mathbf{x},\mathbf{y})$ in (10)
    calculate the acceptance ratio $\alpha(\mathbf{x},\mathbf{y})$ in (11)
    generate a sample $u$ from the uniform density $U[0,1]$
    if $u \leq \alpha(\mathbf{x},\mathbf{y})$ then
        let $\mathbf{X}_i = \mathbf{y}$ and $\mathbf{x}' = \mathbf{y}$
        if $\|\mathbf{c}-\mathbf{B}\mathbf{x}'\| < \|\mathbf{c}-\mathbf{B}\widehat{\mathbf{x}}\|$ then
            update $\widehat{\mathbf{x}} = \mathbf{x}'$
        end if
    else
        $\mathbf{X}_i = \mathbf{x}$
    end if
end for
output $\widehat{\mathbf{x}} = \mathbf{x}'$
```

IV. Complexity of BDD

In this section, we apply the independent MHK sampling to BDD and analyze its complexity. The analysis for independent MH-Peikert and rejection sampling is similar, by changing the value δ. As mentioned before, the decoding complexity of MCMC is evaluated by the number of Markov moves.

In MCMC, samples from the stationary distribution tend to be correlated with each other. Thus one leaves a gap, which is the mixing time t_{mix}, to pick up the desired independent samples (alternatively, one can run multiple Markov chains in parallel to guarantee i.i.d. samples). Therefore, we define the complexity of solving BDD by MCMC as follows.

Definition 2: Let $d(\Lambda,\mathbf{c}) = \min_{\mathbf{x}\in\mathbb{Z}^n}\|\mathbf{B}\mathbf{x}-\mathbf{c}\|$ denote the Euclidean distance between the query point $\mathbf{c}$ and the lattice Λ with basis $\mathbf{B}$, and let $\widehat{\mathbf{x}}$ be the lattice point achieving $d(\Lambda,\mathbf{c})$. The complexity (i.e., the number of Markov moves t) of solving BDD by MCMC is

$$C_{\text{BDD}} \triangleq \frac{t_{\text{mix}}}{D_{\Lambda,\sigma,\mathbf{c}}(\widehat{\mathbf{x}})}. \tag{60}$$

Then, C_{BDD} can be upper bounded by

$$\begin{aligned}
C_{\text{BDD}} &< \log\left(\frac{1}{\epsilon}\right)\cdot\frac{1}{\delta}\cdot\frac{\rho_{\sigma,\mathbf{c}}(\Lambda)}{\rho_{\sigma,\mathbf{c}}(\mathbf{B}\widehat{\mathbf{x}})}\\
&\leq \log\left(\frac{1}{\epsilon}\right)\cdot\frac{\prod_{i=1}^n\rho_{\sigma_i}(\mathbb{Z})}{\rho_{\sigma,\mathbf{c}}(\Lambda)}\cdot\frac{\rho_{\sigma,\mathbf{c}}(\Lambda)}{\rho_{\sigma,\mathbf{c}}(\mathbf{B}\widehat{\mathbf{x}})}\\
&= \log\left(\frac{1}{\epsilon}\right)\cdot\frac{\prod_{i=1}^n\rho_{\sigma_i}(\mathbb{Z})}{\rho_{\sigma,\mathbf{c}}(\mathbf{B}\widehat{\mathbf{x}})}\\
&= \log\left(\frac{1}{\epsilon}\right)\cdot C,
\end{aligned} \tag{61}$$

where

$$C = \frac{\prod_{i=1}^n\rho_{\sigma_i}(\mathbb{Z})}{\rho_{\sigma,\mathbf{c}}(\mathbf{B}\widehat{\mathbf{x}})}. \tag{62}$$

Theorem 4: The complexity of solving BDD by the independent MHK algorithm is bounded above as

$$C_{\text{BDD}} \leq \log\left(\frac{1}{\epsilon}\right) \cdot 1.0039^n \cdot e^{\frac{2\pi \cdot d^2(\Lambda,\mathbf{c})}{\min_i \|\widehat{\mathbf{b}}_i\|^2}}. \tag{63}$$

Proof: To start with, let us examine the numerator in (62)

$$\begin{aligned}\prod_{i=1}^{n} \rho_{\sigma_i}(\mathbb{Z}) &= \prod_{i=1}^{n} \sum_{x_i \in \mathbb{Z}} e^{-\frac{1}{2\sigma_i^2}\|x_i\|^2} \\ &= \prod_{i=1}^{n} \vartheta_3(\|\widehat{\mathbf{b}}_i\|^2/2\pi\sigma^2) \qquad (64)\end{aligned}$$

where we apply the Jacobi theta function ϑ_3 [41].

By substituting (64) to (62), the complexity C is upper bounded as

$$C \leq \prod_{i=1}^{n} \vartheta_3(\|\widehat{\mathbf{b}}_i\|^2/2\pi\sigma^2) \cdot e^{\frac{1}{2\sigma^2}\|\mathbf{B}\widehat{\mathbf{x}}-\mathbf{c}\|^2}. \tag{65}$$

Now, let us recall some facts about Jacobi theta function $\vartheta_3(\tau)$. $\vartheta_3(\tau)$ is monotonically decreasing with τ, and particularly

$$\lim_{\tau\to\infty} \inf \vartheta_3(\tau) = 1. \tag{66}$$

By simple calculation, we can get that

$$\vartheta_3(2) = \sum_{n=-\infty}^{+\infty} e^{-2\pi n^2} = \frac{\sqrt[4]{6\pi + 4\sqrt{2}\pi}}{2\Gamma(\frac{3}{4})} = 1.0039, \tag{67}$$

where $\Gamma(\cdot)$ stands for the *Gamma function*. Clearly, if

$$\frac{\min_{1\leq i\leq n} \|\widehat{\mathbf{b}}_i\|^2}{2\pi\sigma^2} \geq 2 \tag{68}$$

it turns out that the following term

$$\prod_{i=1}^{n} \vartheta_3(\|\widehat{\mathbf{b}}_i\|^2/2\pi\sigma^2) \leq \vartheta_3^n(2) = 1.0039^n \tag{69}$$

is rather small even for values of n up to hundreds (e.g., $1.0039^{100} = 1.4467$). The key point here is that the pre-exponential factor is close to 1. For better accuracy, $\vartheta_3(3) = 1.00037$ (or $\vartheta_3(4)$ etc.) can be applied so that $1.00037^{1000} = 1.4476$. More options about ϑ_3 can be found in Table I.

Therefore, if σ satisfies the condition (68), namely

$$\sigma \leq \min_{1\leq i\leq n} \|\widehat{\mathbf{b}}_i\|/(2\sqrt{\pi}), \tag{70}$$

then we have

$$C \leq 1.0039^n \cdot e^{\frac{1}{2\sigma^2}\|\mathbf{B}\widehat{\mathbf{x}}-\mathbf{c}\|^2}. \tag{71}$$

Setting $\sigma = \min_i \|\widehat{\mathbf{b}}_i\|/(2\sqrt{\pi})$, we finally arrive at the following result

$$C_{\text{BDD}} \leq \log\left(\tfrac{1}{\epsilon}\right) \cdot 1.0039^n \cdot e^{\frac{2\pi}{\min_i \|\widehat{\mathbf{b}}_i\|^2}\|\mathbf{B}\widehat{\mathbf{x}}-\mathbf{c}\|^2}, \tag{72}$$

completing the proof. ■

Let us highlight the significance of lattice reduction. Lattice reduction is able to significantly improve $\min_i \|\widehat{\mathbf{b}}_i\|$ while reducing $\max_i \|\widehat{\mathbf{b}}_i\|$ [42]. Therefore, increasing $\min_i \|\widehat{\mathbf{b}}_i\|$ will significantly decrease the complexity shown above.

TABLE I

VALUES OF ϑ_3

$\vartheta_3(1)$	$\sum_{n=-\infty}^{+\infty} e^{-1\pi n^2}$	$\frac{\sqrt[4]{\pi}}{\Gamma(\frac{3}{4})}$	1.087
$\vartheta_3(2)$	$\sum_{n=-\infty}^{+\infty} e^{-2\pi n^2}$	$\frac{\sqrt[4]{6\pi+4\sqrt{2}\pi}}{2\Gamma(\frac{3}{4})}$	1.0039
$\vartheta_3(3)$	$\sum_{n=-\infty}^{+\infty} e^{-3\pi n^2}$	$\frac{\sqrt[4]{27\pi+18\sqrt{3}\pi}}{3\Gamma(\frac{3}{4})}$	1.00037
$\vartheta_3(4)$	$\sum_{n=-\infty}^{+\infty} e^{-4\pi n^2}$	$\frac{\sqrt[4]{8\pi}+2\sqrt[4]{\pi}}{4\Gamma(\frac{3}{4})}$	1.0002
$\vartheta_3(5)$	$\sum_{n=-\infty}^{+\infty} e^{-5\pi n^2}$	$\frac{\sqrt[4]{225\pi+100\sqrt{5}\pi}}{5\Gamma(\frac{3}{4})}$	1.0001

Remark 2: In fact, such an analysis also holds for Klein's algorithm, where the probability of sampling $\mathbf{x}$ follows a Gaussian-like distribution [15]

$$P(\mathbf{x}) \geq \frac{e^{-\frac{1}{2\sigma^2}\|\mathbf{B}\mathbf{x}-\mathbf{c}\|^2}}{\prod_{i=1}^{n} \vartheta_3(\|\widehat{\mathbf{b}}_i\|^2/2\pi\sigma^2)}. \tag{73}$$

Klein chose $\sigma = \min_i \|\widehat{\mathbf{b}}_i\|/\sqrt{2\log n}$, which corresponds to $O(n^{d^2(\Lambda,\mathbf{c})/\min_i \|\widehat{\mathbf{b}}_i\|^2})$ complexity. Here, we have shown that the decoding complexity can be further reduced to $O(e^{d^2(\Lambda,\mathbf{c})/\min_i^2 \|\widehat{\mathbf{b}}_i\|})$, by setting $\sigma = \min_i \|\widehat{\mathbf{b}}_i\|/(2\sqrt{\pi})$. With the help of HKZ reduction, $\min_i \|\widehat{\mathbf{b}}_i\| \geq \frac{1}{n}\lambda_1(\Lambda)$ [43]. Thus, Klein's algorithm allows to solve the η-BDD with $\eta = O(1/n)$ in polynomial time, while our result shown in (72) improves it to $\eta = O(\sqrt{\log n}/n)$.

According to (63), we have

$$d(\Lambda, \mathbf{c}) = \sqrt{\frac{1}{2\pi} \cdot \ln \frac{C_{\text{BDD}}}{a}} \cdot \min_{1\leq i\leq n} \|\widehat{\mathbf{b}}_i\|. \tag{74}$$

where $a = \log\left(\frac{1}{\epsilon}\right) \cdot 1.0039^n \approx \log\left(\frac{1}{\epsilon}\right)$. Clearly, the decoding radius increases with C_{BDD}, implying a flexible trade-off between the decoding performance and complexity. In addition, the significance of lattice reduction can be seen due to an increased value of $\min_i \|\widehat{\mathbf{b}}_i\|$.

V. Trapdoor Sampling

The core technique underlying GPV's signature scheme is discrete Gaussian sampling over a trapdoor lattice [11]. Its security crucially relies on the property that the output distribution of discrete Gaussian sampling is oblivious to any particular basis used in the sampling process, therefore preventing leakage of the private key. The original GPV signature scheme was based on Klein's algorithm, which was subsequently extended to Peikert's algorithm [16] (see also [44, Ch. 6] for sampling over structured lattices). In fact, any good Gaussian sampling algorithms can be applied to GPV signatures. In this Section, we demonstrate the security advantage of MCMC in GPV signatures, thanks to smaller parameters it can reach.

Firstly, we provide a high-level introduction to the GPV signature (see [11], [16] for details). In key generation, one generates a hard public basis for a random lattice Λ, together with a short private basis of Λ. The public basis serves as the

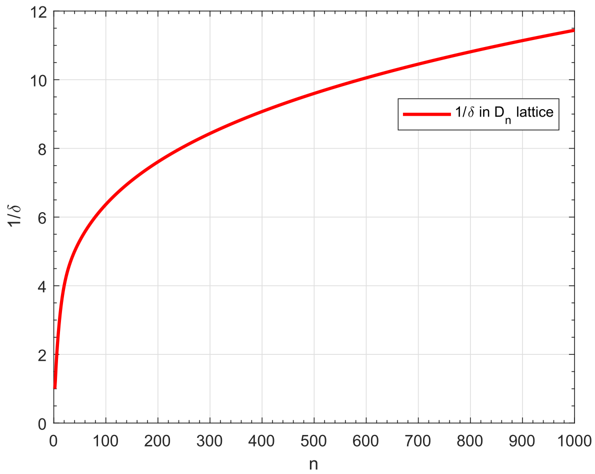

Fig. 3. $\frac{1}{\delta}$ as a function of n for lattice D_n with $\mathbf{c} = 0$ and $\sigma^2 = -8$ dB.

public key, while the private basis serves as the private key. Given a message $\mathbf{m}$ (or rather a digest of $\mathbf{m}$), one uses the private basis to sample a point $\mathbf{x}$ from $D_{\Lambda+\mathbf{m},\sigma}$ with parameter σ. The signature of $\mathbf{m}$ is $\mathbf{x}$. The verifier checks that $\mathbf{x}$ is short and that $\mathbf{x} - \mathbf{m} \in \Lambda$ using the public basis.

It is shown in [11] that the security of GPV signing can be reduced to the hardness of the inhomogeneous short integer solution (ISIS) problem[3] with approximation factor $\sqrt{n}\sigma$. Therefore, the width σ is the most important property of a discrete Gaussian sampler in this context.

Obviously, there is a tradeoff between security and running time in trapdoor sampling with MCMC. A small parameter σ gives higher security, but require longer running time. Next, we examine the impact of decreasing σ on the mixing time. Again, we focus on the independent MHK algorithm. Recall it's the mixing time is proportional to

$$\frac{1}{\delta} = \frac{\prod_{i=1}^{n} \rho_{\sigma_i}(\mathbb{Z})}{\rho_{\sigma,\mathbf{c}}(\Lambda)}. \tag{75}$$

Our intuition here is that if a good basis is available (as in the case of trapdoor sampling), then $\frac{1}{\delta}$ will not blow up as n grows. To give an impression, Fig. 3 shows $\frac{1}{\delta}$ as a function of n for checkerboard lattice D_n with $\mathbf{c} = \mathbf{0}$ and $\sigma^2 = -8$ dB, using its well-known basis [41, p.117, (86)]. It is seen that $\frac{1}{\delta}$ merely grows to 12 for n up to 1000.

What if $\mathbf{c} \neq \mathbf{0}$? Then the denominator of (75) can be unpredictable in general. Fortunately, it can be bounded if σ is above the smoothing parameter. Recall that for a lattice Λ and for $\varepsilon > 0$, the smoothing parameter[4] $\eta_\varepsilon(\Lambda)$ is defined as the smallest $\sigma > 0$ such that $\sum_{\mathbf{x}^* \in \Lambda^* \backslash \{\mathbf{0}\}} e^{-2\pi^2\sigma^2\|\mathbf{x}^*\|^2} \leq \varepsilon$. If $\varepsilon < 1$, we have $\frac{\rho_{\sigma,\mathbf{c}}(\Lambda)}{(\sqrt{2\pi}\sigma)^n} \in \frac{1}{\mathrm{Vol}(\Lambda)}[1-\varepsilon, 1+\varepsilon]$, $\forall \mathbf{c}$.

Here, we are concerned with the parameter region $\sigma \in [\eta_\varepsilon(\Lambda), \sqrt{\omega(\log\, n)} \cdot \max\|\widehat{\mathbf{b}}_i\|]$, below GPV's parameter [11] but above the smoothing parameter. This is because we anticipate only moderate growth in mixing time but significant increase of security for values of σ just below GPV's parameter.

Let I denote the subset of indexes i with $\sqrt{2\pi}\sigma_i > 1$ (i.e., $\sqrt{2\pi}\sigma > \|\widehat{\mathbf{b}}_i\|$), $i \in \{1, 2, \ldots, n\}$, $|I| = m$. It is not difficult to derive the following bound, similarly to [18, Proposition 4]:

$$\begin{aligned}\frac{1}{\delta} &= \frac{\prod_{i=1}^{n} \vartheta_3(\frac{1}{2\pi\sigma_i^2})}{\rho_{\sigma,\mathbf{c}}(\Lambda)} \\ &\in \frac{\prod_{i=1}^{n} \sqrt{2\pi}\sigma_i \vartheta_3(2\pi\sigma_i^2)}{(\sqrt{2\pi}\sigma)^n/\mathrm{Vol}(\Lambda)}[1-2\varepsilon, 1+2\varepsilon] \\ &= \prod_{i=1}^{n} \vartheta_3(2\pi\sigma_i^2)[1-2\varepsilon, 1+2\varepsilon] \\ &\leq \vartheta_3(1)^m \cdot \prod_{i \notin I} \frac{2}{\sqrt{2\pi}\sigma_i} \cdot (1+2\varepsilon)\end{aligned} \tag{76}$$

where we use the identity $\vartheta_3\left(\frac{1}{\tau^2}\right) = \tau\vartheta_3(\tau^2)$ and assume $\varepsilon < 1/2$ in the second step, and $\vartheta_3(\tau) \leq 1 + \sqrt{\frac{1}{\tau}}$ in the last step.

Particularly, if $\sqrt{2\pi}\sigma \geq \sqrt{\alpha}\max_{1\leq i\leq n}\|\widehat{\mathbf{b}}_i\|$ for some $\alpha \geq 1$, we derive

$$\frac{1}{\delta} \leq \vartheta_3(\alpha)^n(1+2\varepsilon). \tag{77}$$

Again, our key observation is that the mixing time $\vartheta_3(\alpha)^n$ grows rather slowly for values of α that are not too small. For example, when $\alpha = 2$, we have $\vartheta_3(2)^n = 1.0039^{1000} = 49$ for $n = 1000$. This means that with roughly 49 iterations, our MCMC sampler is able to reduce the parameter from $\sqrt{\omega(\log n)}\max_{1\leq i\leq n}\|\widehat{\mathbf{b}}_i\|$ to $\frac{1}{\sqrt{\pi}}\max_{1\leq i\leq n}\|\widehat{\mathbf{b}}_i\|$. Therefore, if one is willing to use a slower signature scheme in return for higher security, MCMC offers such an option.

Example 1 (FALCON): FALCON [45] is a GPV signature scheme instantiated by NTRU lattices. Let m be a power of two, $n = \varphi(m)$ where $\varphi(\cdot)$ is Euler's totient function, $q \in \mathbb{N}$. The secret key consists of two polynomials f and g in ring $R = \mathbb{Z}[x]/(x^n + 1)$ where f is invertible. Find G and F such that

$$fG - gF = q \mod x^n + 1.$$

The NTRU lattice of dimension $2n$ is generated by the private basis

$$\mathbf{B} = \begin{pmatrix} \mathcal{C}(g) & -\mathcal{C}(f) \\ \mathcal{C}(G) & -\mathcal{C}(F) \end{pmatrix}^T$$

where $\mathcal{C}(\cdot)$ denotes an $n \times n$ nega-cyclic matrix whose first row consists of the coefficients of a polynomial. The public basis is given by

$$\mathbf{A} = \begin{pmatrix} -\mathcal{C}(h) & \mathbf{I}_n \\ q\mathbf{I}_n & \mathbf{O}_n \end{pmatrix}^T$$

where $h = g/f \mod q$. Both bases $\mathbf{B}$ and $\mathbf{A}$ generate the same lattice

$$\Lambda = \{(\mathbf{u}, \mathbf{v}) \in R^2 | \mathbf{u} + \mathbf{vh} = 0 \mod q\}$$

[3] In the language of coding theory, this is to find a short vector in a coset of a linear code.

[4] Note again the difference from the definition in [8], where σ is scaled by a constant factor $\sqrt{2\pi}$.

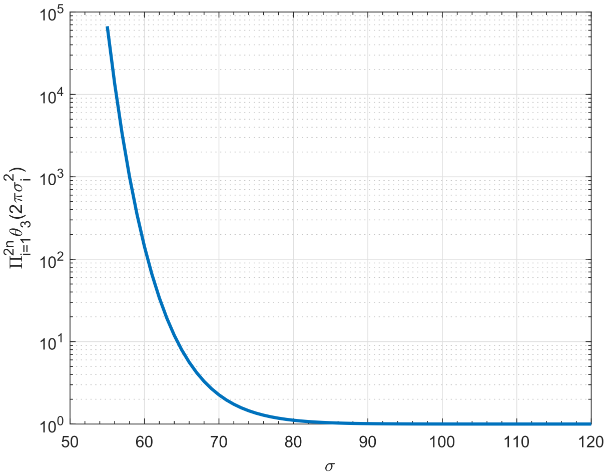

Fig. 4. $\prod_{i=1}^{2n}\theta_3(2\pi\sigma_i^2)$ as a function of σ for an NTRU lattice with $n = 512$.

We consider the parameters $n = 512$ and $q = 12289$ in FALCON. The coefficients of polynomials f and g are randomly sampled from $D_{\mathbb{Z},4.05}$. For a particular instance randomly generated, we find $\max_{1\leq i\leq n}\|\widehat{\mathbf{b}}_i\| = 127$. In Fig. 4, we show as a function of σ the term $\prod_{i=1}^{2n}\theta_3(2\pi\sigma_i^2)$ in (76), which characterizes the complexity $1/\delta$ above the smoothing parameter. It is seen that MCMC is able to significantly reduce the parameter σ, with quite moderate increase in complexity. Specifically, the term $\prod_{i=1}^{2n}\theta_3(2\pi\sigma_i^2)$ merely grows to about 20, even if σ is halved relative to $\max_{1\leq i\leq n}\|\widehat{\mathbf{b}}_i\|$. Recall that GPV sampling requires $\sigma = \sqrt{\omega(\log\ n)}\cdot\max_{1\leq i\leq n}\|\widehat{\mathbf{b}}_i\|$.

Note that it is possible for MCMC to incorporate the fast Fourier sampler [45], which would speed up the sampling process for structured lattices. The security levels of various samplers have been evaluated in [44, Ch. 6]. We leave evaluation of the concrete security of MCMC samplers to future work.

VI. Multiple-Try Metropolis-Klein Algorithm

In this section, the independent multiple-try Metropolis-Klein (MTMK) algorithm is proposed to enhance the mixing. We firstly prove its validity and then show its uniform ergodicity with an improved convergence rate.

A. Multiple-Try Metropolis Method

Rather than directly generating the state candidate $\mathbf{y}$ from the proposal distribution $q(\mathbf{x}, \mathbf{y})$, the multiple-try Metropolis (MTM) method selects $\mathbf{y}$ among a set of i.i.d. trial samples from $q(\mathbf{x}, \mathbf{y})$, which significantly expands the searching region of proposals [46]. In particular, the MTM method consists of the following steps:

1) *Given the current state* $\mathbf{X}_t = \mathbf{x}$, *draw* k *i.i.d. state candidates* $\mathbf{y}_1, \ldots, \mathbf{y}_k$ *from the proposal distribution* $q(\mathbf{x}, \mathbf{y})$.

2) *Select* $\mathbf{y} = \mathbf{y}_c$ *among* $\{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$ *with probability proportional to the weight*

$$\omega(\mathbf{y}_i, \mathbf{x}) = \pi(\mathbf{y}_i)q(\mathbf{y}_i, \mathbf{x})\lambda(\mathbf{y}_i, \mathbf{x}), \quad i = 1, \ldots, k, \tag{78}$$

where $\lambda(\mathbf{y}, \mathbf{x})$ *is a nonnegative symmetric function of* $\mathbf{y}$ *and* $\mathbf{x}$ *defined initially.*

Algorithm 5 Independent Multiple-try Metropolis-Klein Sampling Decoder

```
Input: $\mathbf{B}, \sigma, \mathbf{c}, \mathbf{x}_0, t$;
Output: $\mathbf{x} \sim D_{\Lambda,\sigma,\mathbf{c}}$;
let $\widehat{\mathbf{x}} = \mathbf{x}^0$ and $\mathbf{X}_0 = \mathbf{x}_0$
for $i$ =1, ..., $t$ do
    let $\mathbf{x}$ denote the state of $\mathbf{X}_{t-1}$
    sample $k$ trial samples $\mathbf{y}_1 \ldots \mathbf{y}_k$ from $q(\mathbf{x}, \mathbf{y})$ in (81)
    select $\mathbf{y} = \mathbf{y}_c$ from $\mathbf{y}_1 \ldots \mathbf{y}_k$ based on $\omega(\mathbf{y}_i)$ in (82)
    calculate the acceptance ratio $\alpha(\mathbf{x}, \mathbf{y})$ in (83)
    generate a sample $u$ from the uniform density $U[0, 1]$
    if $u \leq \alpha(\mathbf{x}, \mathbf{y})$ then
        let $\mathbf{X}_i = \mathbf{y}$ and $\mathbf{x}' = \mathbf{y}$
        if $\|\mathbf{c} - \mathbf{B}\mathbf{x}'\| < \|\mathbf{c} - \mathbf{B}\widehat{\mathbf{x}}\|$ then
            update $\widehat{\mathbf{x}} = \mathbf{x}'$
        end if
    else
        $\mathbf{X}_i = \mathbf{x}$
    end if
end for
output $\widehat{\mathbf{x}} = \mathbf{x}'$
```

3) *Draw* $k-1$ *i.i.d. reference candidates* $\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}$ *from the proposal distribution* $q(\mathbf{y}, \mathbf{x})$ *and let* $\mathbf{x}_k = \mathbf{x}$.

4) *Accept* $\mathbf{y} = \mathbf{y}_c$ *as the state of* $\mathbf{X}_{t+1}$, *i.e.,* $\mathbf{X}_{t+1} = \mathbf{y}$ *with probability*

$$\alpha_{\mathrm{MTM}} = \min\left\{1, \frac{\omega(\mathbf{y}_1, \mathbf{x}) + \ldots + \omega(\mathbf{y}_k, \mathbf{x})}{\omega(\mathbf{x}_1, \mathbf{y}) + \ldots + \omega(\mathbf{x}_k, \mathbf{y})}\right\}, \tag{79}$$

otherwise, with probability $1 - \alpha_{\mathrm{MTM}}$, *let* $\mathbf{X}_{t+1} = \mathbf{X}_t = \mathbf{x}$.

By exploring the search region more thoroughly, an improvement of convergence can be achieved by MTM. Based on a number of trial samples generated from the proposal distribution, the Markov chain enjoys a large step-size jump within every single move without lowering the acceptance rate. It should be noticed that the $k-1$ reference samples $\mathbf{x}_i$'s are involved only for the validity of MTM by satisfying the detailed balance condition [46]

$$\pi(\mathbf{x})P(\mathbf{x}, \mathbf{y}) = \pi(\mathbf{y})P(\mathbf{y}, \mathbf{x}). \tag{80}$$

Clearly, the efficiency of MTM relies on the number of trial samples k while the traditional MH sampling is a special case with $k = 1$. Similar to MH sampling, there is considerable flexibility in the choice of the proposal distribution $q(\mathbf{x}, \mathbf{y})$ in MTM [47]. Actually, it is even possible to use different proposal distributions to generate trial samples without altering the ergodicity of the Markov chain [48]. Meanwhile, the nonnegative symmetric function $\lambda(\mathbf{x}, \mathbf{y})$ in (78) is also flexible, where the only requirement is that $\lambda(\mathbf{x}, \mathbf{y}) > 0$ whenever $q(\mathbf{x}, \mathbf{y}) > 0$.

B. The Proposed Algorithm

With the great flexibility offered by $q(\mathbf{x}, \mathbf{y})$ and $\lambda(\mathbf{x}, \mathbf{y})$, we now propose the independent multiple-try Metropolis-Klein (MTMK) algorithm, which is described by the following steps:

1) *Given the current state* $\mathbf{X}_t = \mathbf{x}$, *use Klein's algorithm to draw k i.i.d. state candidates* $\mathbf{y}_1, \ldots, \mathbf{y}_k$ *from the independent proposal distribution in (10)*

$$q(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^{n} P(y_{n+1-i} | \overline{\mathbf{y}}_{[-(n+1-i)]}) = q(\mathbf{y}). \tag{81}$$

2) *Let* $\lambda(\mathbf{x}, \mathbf{y}) = [q(\mathbf{x}, \mathbf{y})q(\mathbf{y}, \mathbf{x})]^{-1} = [q(\mathbf{y})q(\mathbf{x})]^{-1}$. *Then select* $\mathbf{y} = \mathbf{y}_c$ *among* $\{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$ *with probability proportional to the importance weight*

$$\omega(\mathbf{y}_i, \mathbf{x}) = \frac{\pi(\mathbf{y}_i)}{q(\mathbf{y}_i)} = \omega(\mathbf{y}_i), \quad i = 1, \ldots, k. \tag{82}$$

3) *Accept* $\mathbf{y} = \mathbf{y}_c$ *as the state of* $\mathbf{X}_{t+1}$ *with acceptance rate*

$$\alpha_{\text{MTM}} = \min\left\{1, \frac{\omega(\mathbf{y}_c) + \sum_{j=1, j\neq c}^{k} \omega(\mathbf{y}_j)}{\omega(\mathbf{x}) + \sum_{j=1, j\neq c}^{k} \omega(\mathbf{y}_j)}\right\}, \tag{83}$$

otherwise, with probability $1 - \alpha_{\text{MTM}}$, *let* $\mathbf{X}_{t+1} = \mathbf{X}_t = \mathbf{x}$.

In the proposed algorithm, the basic formulation of MTM is modified in three aspects. First, Klein's algorithm is applied to generate trial state candidates from the independent proposal distribution $q(\mathbf{x}, \mathbf{y}) = q(\mathbf{y})$. Then, by setting $\lambda(\mathbf{x}, \mathbf{y}) = [q(\mathbf{x}, \mathbf{y})q(\mathbf{y}, \mathbf{x})]^{-1}$, $\omega(\mathbf{x}, \mathbf{y})$ becomes the *importance weight* of $\mathbf{x}$ that we have defined in (14). Finally and interestingly, thanks to the independent proposals, the generation of reference samples $\mathbf{x}_i$'s can be removed without changing the ergodicity of the chain.

In the case of independent proposals, because both the trial samples $\mathbf{y}_i$'s and the reference samples $\mathbf{x}_i$'s are generated independently from the identical distribution $q(\cdot)$, the generation of reference samples can be greatly simplified by just setting $\mathbf{x}_i = \mathbf{y}_i$ for $i = 1, \ldots, c-1, c+1, \ldots, k$ and $\mathbf{x}_c = \mathbf{x}$. Actually, with the same arguments, the trial samples generated in the previous Markov moves can also be used by $\mathbf{x}_i$ [49].

It is well known that a Markov chain which is irreducible and aperiodic will be ergodic if the detailed balance condition is satisfied [19]. Since irreducible and aperiodic are easy to verify, we show the validity of the proposed algorithm by demonstrating the detailed balance condition.

Theorem 5: Given the target lattice Gaussian distribution $D_{\Lambda,\sigma,\mathbf{c}}$, the Markov chain induced by the independent MTMK algorithm is ergodic.

Proof: To start with, let us specify the transition probability $P(\mathbf{x}, \mathbf{y})$ of the underlying Markov chain. For ease of presentation, we only consider the case of $\mathbf{x} \neq \mathbf{y}$, since the case $\mathbf{x} = \mathbf{y}$ is trivial. The transition probability $P(\mathbf{x}, \mathbf{y})$ can be expressed as

$$P(\mathbf{x}, \mathbf{y} = \mathbf{y}_c) = \sum_{i=1}^{k} p(\mathbf{y}_c | \mathbf{x}, c = i). \tag{84}$$

Here, $p(\mathbf{y}_c | \mathbf{x}, c = i)$ represents the probability of accepting $\mathbf{y} = \mathbf{y}_c$ as the new state of $\mathbf{X}_{t+1}$ given the previous one $\mathbf{X}_t = \mathbf{x}$ when the cth candidate among $\mathbf{y}_i$'s is selected. Moreover, as $\mathbf{y}_i$ is exchangeable and independent, it follows that $p(\mathbf{y}_i | \mathbf{x}, i) = p(\mathbf{y}_j | \mathbf{x}, j)$ by symmetry, namely,

$$P(\mathbf{x}, \mathbf{y} = \mathbf{y}_c) = k \cdot p(\mathbf{y}_c | \mathbf{x}, c). \tag{85}$$

In contrast to MH algorithms, the generation of the state candidate $\mathbf{y} = \mathbf{y}_c$ for Markov move $\mathbf{X}_{t+1}$ in MTM actually follows a distribution formed by $q(\mathbf{x}, \mathbf{y})$ and $\omega(\mathbf{y}, \mathbf{x})$ together [46]. More precisely, $p(\mathbf{y}_c | \mathbf{x}, c)$ can be further expressed as (86), at the bottom of this page, where the terms inside the sum correspond to $q(\mathbf{x}, \mathbf{y})$, $\omega(\mathbf{y}, \mathbf{x})$ and α respectively.

From (86), it is straightforward to verify the term $\pi(\mathbf{x})p(\mathbf{y}_c | \mathbf{x}, c)$ is symmetric in $\mathbf{x}$ and $\mathbf{y}_c$, namely

$$\pi(\mathbf{x})p(\mathbf{y}_c | \mathbf{x}, c) = \pi(\mathbf{y}_c)p(\mathbf{x} | \mathbf{y}_c, c). \tag{87}$$

Then, by simple substitution, the detailed balance condition is satisfied as

$$\pi(\mathbf{x})P(\mathbf{x}, \mathbf{y} = \mathbf{y}_c) = \pi(\mathbf{y})p(\mathbf{y} = \mathbf{y}_c, \mathbf{x}), \tag{88}$$

completing the proof. ■

C. Convergence Analysis

Theorem 6: Given the invariant lattice Gaussian distribution $D_{\Lambda,\sigma,\mathbf{c}}$, the Markov chain induced by the independent MTMK sampling algorithm converges exponentially fast to the stationary distribution:

$$\|P^t(\mathbf{x}, \cdot) - D_{\Lambda,\sigma,\mathbf{c}}(\cdot)\|_{TV} \leq (1 - \delta_{\text{MTM}})^t \tag{89}$$

with

$$\delta_{\text{MTM}} = \frac{k}{k - 1 + \frac{1}{\delta}}. \tag{90}$$

The proof of Theorem 6 is provided in Appendix I.

From (90), it can be observed that with the increase of the trial sample size k, the exponential decay coefficient

$$\begin{aligned} p(\mathbf{y}_c | \mathbf{x}, c) &= \sum_{\mathbf{y}_{1:c-1} \in \mathbb{Z}^n} \sum_{\mathbf{y}_{c+1:k} \in \mathbb{Z}^n} \left\{ \left[\prod_{j=1}^{k} q(\mathbf{x}, \mathbf{y}_j) \right] \cdot \frac{\omega(\mathbf{y}_c)}{\sum_{i=1}^{k} \omega(\mathbf{y}_i)} \cdot \min\left\{1, \frac{\omega(\mathbf{y}_c) + \sum_{j=1, j\neq c}^{k} \omega(\mathbf{y}_j)}{\omega(\mathbf{x}) + \sum_{j=1, j\neq c}^{k} \omega(\mathbf{y}_j)}\right\} \right\} \\ &= q(\mathbf{y}_c) \cdot \omega(\mathbf{y}_c) \cdot \sum_{\mathbf{y}_{1:c-1} \in \mathbb{Z}^n} \sum_{\mathbf{y}_{c+1:k} \in \mathbb{Z}^n} \left\{ \left[\prod_{j=1, j\neq c}^{k} q(\mathbf{y}_j) \right] \cdot \min\left\{ \frac{1}{\omega(\mathbf{y}_c) + \sum_{j=1, j\neq c}^{k} \omega(\mathbf{y}_j)}, \frac{1}{\omega(\mathbf{x}) + \sum_{j=1, j\neq c}^{k} \omega(\mathbf{y}_j)} \right\} \right\} \\ &= \pi(\mathbf{y}_c) \cdot \min\left\{ \sum_{\mathbf{y}_{1:c-1} \in \mathbb{Z}^n} \sum_{\mathbf{y}_{c+1:k} \in \mathbb{Z}^n} \left\{ \frac{\prod_{j=1, j\neq c}^{k} q(\mathbf{y}_j)}{\omega(\mathbf{y}_c) + \sum_{j=1, j\neq c}^{k} \omega(\mathbf{y}_j)} \right\}, \sum_{\mathbf{y}_{1:c-1} \in \mathbb{Z}^n} \sum_{\mathbf{y}_{c+1:k} \in \mathbb{Z}^n} \left\{ \frac{\prod_{j=1, j\neq c}^{k} q(\mathbf{y}_j)}{\omega(\mathbf{x}) + \sum_{j=1, j\neq c}^{k} \omega(\mathbf{y}_j)} \right\} \right\} \end{aligned} \tag{86}$$

$\delta_{\text{MTM}} = \frac{k}{k-1+\frac{1}{\delta}}$ will approach 1. In other words, with a sufficiently large k, sampling from the target distribution can be realized efficiently. More importantly, the generation of k trial samples at each Markov move not only allows a fully parallel implementation, but also can be carried out in a preprocessing stage, which is beneficial in practice.

Now, given $\delta_{\text{MTM}} = \frac{k}{k-1+\frac{1}{\delta}}$, the mixing time of the underlying Markov chain can be estimated. Specifically, according to (8) and (89), we obtain

$$\begin{aligned} t_{\text{mix}}^{\text{MTM}}(\epsilon) &= \frac{\ln\epsilon}{\ln(1-\delta_{\text{MTM}})} \\ &\overset{(g)}{<} \log\left(\frac{1}{\epsilon}\right)\cdot\left(\frac{1}{\delta_{\text{MTM}}}\right) \\ &= \log\left(\frac{1}{\epsilon}\right)\cdot\left(\frac{k-1+\frac{1}{\delta}}{k}\right) \\ &\approx \log\left(\frac{1}{\epsilon}\right)\cdot\left(\frac{1}{k\delta}\right), \quad \epsilon < 1, \ \frac{1}{\delta} \gg k \end{aligned} \tag{91}$$

where we again use the bound $\ln(1-\alpha) < -\alpha$ for $0 < \alpha < 1$ in (g). Clearly, the mixing time is proportional to $\frac{1}{k\delta}$, and becomes $O(1)$ if $k\delta \to 1$. Overall, compared with the mixing time given in (25), the mixing time of the independent MTMK is significantly reduced by a factor of k. Since the independent MTMK inherits all the formulations of the independent MHK, we have

$$\begin{aligned} C_{\text{BDD}}^{\text{MTM}} &= \frac{t_{\text{mix}}^{\text{MTM}}(\epsilon)}{D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x})} \\ &\lesssim \frac{1}{k}\cdot\frac{\log\left(\frac{1}{\epsilon}\right)\cdot\left(\frac{1}{\delta}\right)}{D_{\Lambda,\sigma,\mathbf{c}}(\widehat{\mathbf{x}})} \\ &= \frac{1}{k}\cdot\log\left(\frac{1}{\epsilon}\right)\cdot C \\ &= \frac{1}{k}\cdot\log\left(\frac{1}{\epsilon}\right)\cdot 1.0039^n\cdot e^{\frac{2\pi\cdot d^2(\Lambda,\mathbf{c})}{\min_i\|\widehat{\mathbf{b}}_i\|^2}} \end{aligned} \tag{92}$$

for $\sigma = \min_i \|\widehat{\mathbf{b}}_i\|/(2\sqrt{\pi})$.

Following the afore-mentioned derivation, the decoding radius of the independent MTMK algorithm can be easily obtained as

$$R_{\text{MTM}} = \sqrt{\frac{1}{2\pi}\cdot\ln\frac{kC_{\text{BDD}}^{\text{MTM}}}{a}}\cdot\min_{1\leq i\leq n}\|\widehat{\mathbf{b}}_i\|. \tag{93}$$

Remark 3: Although the independent MTMK algorithm is able to reduce the mixing time, its complexity in each move increases due to multiple calls of trial samples. Therefore, parallel implementation or preprocessing is highly desired to ease the complexity burden.

Moreover, it is possible to have a varying k at each Markov move, thereby resulting in an adaptive independent MTMK algorithm as

$$\|P^t(\mathbf{x},\cdot) - D_{\Lambda,\sigma,\mathbf{c}}(\cdot)\|_{TV} \leq \prod_{i=1}^{t}(1-\delta_{\text{MTM}}^{i}), \tag{94}$$

where $\delta_{\text{MTM}}^{i} = \frac{k_i}{k_i-1+\frac{1}{\delta}}$ and k_i denotes the size of trial samples at each Markov move [50].

VII. Experiments of MIMO Detection

In this section, performance of the MCMC decoding algorithms is evaluated in MIMO detection. Specifically, we present simulation results for an $n \times n$ MIMO system with a square channel matrix. Here, the ith entry of the transmitted signal $\mathbf{x}$, denoted as x_i, is a modulation symbol taken independently from an M-QAM constellation $\mathcal{X}$ with Gray mapping. Meanwhile, we assume a flat fading environment, where the channel matrix $\mathbf{H}$ contains uncorrelated complex Gaussian fading gains with unit variance and remains constant over each frame duration. Let E_b represents the average power per bit at the receiver, then the signal-to-noise ratio (SNR) $E_b/N_0 = n/(\log_2(M)\sigma_w^2)$ where M is the modulation level and σ_w^2 is the noise power. Then, we can express the system model as

$$\mathbf{c} = \mathbf{H}\mathbf{x} + \mathbf{w}. \tag{95}$$

Typically, in the case of Gaussian noise $\mathbf{w}$ with zero mean and variance σ_w^2, it follows from (72) that

$$C \approx O(e^{2\pi n\sigma_w^2/\min_i\|\widehat{\mathbf{b}}_i\|^2}) \tag{96}$$

as $\|\mathbf{B}\mathbf{x}-\mathbf{c}\|^2 \approx n\sigma_w^2$ by the law of large numbers. Therefore, the decoding complexity C decrease with the SNR. Note that the noise variance σ_w^2 is different from the standard deviation σ of the lattice Gaussian distribution.[5]

On the other hand, soft-output decoding for MIMO bit interleaver coded modulation (BICM) system is also possible using the samples generated by MCMC. Specifically, the sample candidates can be used to approximate the log-likelihood ratio (LLR), as in [52]. For bit $b_i \in \{0, 1\}$, the approximated LLR is computed as

$$L(b_i|\mathbf{c}) = \log\frac{\sum_{\mathbf{x}:b_i=1}\exp\left(-\frac{1}{2\sigma^2}\parallel\mathbf{c}-\mathbf{H}\mathbf{x}\parallel^2\right)}{\sum_{\mathbf{x}:b_i=0}\exp\left(-\frac{1}{2\sigma^2}\parallel\mathbf{c}-\mathbf{H}\mathbf{x}\parallel^2\right)}, \tag{97}$$

where b_i is the ith information bit associated with sample $\mathbf{x}$. The notation $\mathbf{x} : b_i = \mu$ means the set of all vectors $\mathbf{x}$ for which $\mathbf{x} : b_i = \mu$.

Fig. 5 shows the bit error rate (BER) of MCMC decoding in a 8×8 uncoded MIMO system with 16-QAM, where all the samples generated by MCMC algorithms are taken into account for decoding. This corresponds to a lattice decoding scenario with dimension $n = 16$. The performance of zero-forcing (ZF) and maximum-likelihood (ML) decoding are shown as benchmarks. For a fair comparison, sequential Gibbs sampling is applied here, which performs 1-dimensional conditional sampling of x_i in a backward order,[6] completing a full iteration [27]. This corresponds to one Markov move in the independent MHK and MTMK algorithms, which also update n components of $\mathbf{x}$ in one iteration.

As expected, with $t = 50$ Markov moves (i.e., iterations), independent MHK outperforms Gibbs sampling. As σ has a vital impact on the sampling algorithms, Gibbs sampling is illustrated by tuning σ with different values. Note that

[5] In [27], the noise variance σ_w^2 is used as the sampling variance, but this would lead to a stalling problem at high SNRs [51].

[6] A forward update of x_i in sequential Gibbs sampling is also possible.

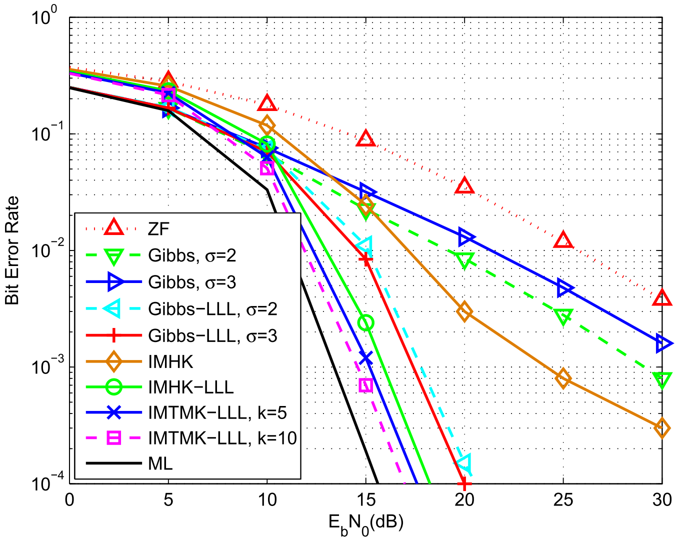

Fig. 5. Bit error rate versus average SNR per bit for the uncoded 8 × 8 MIMO system using 16-QAM under 50 Markov moves.

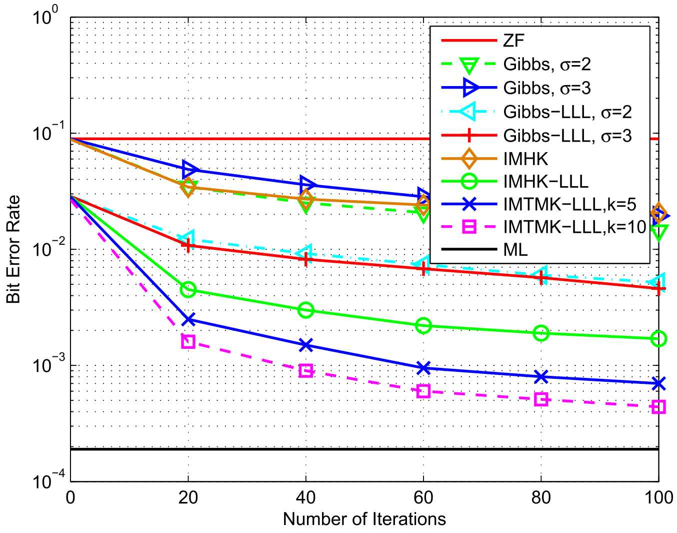

Fig. 6. Bit error rate versus the number of Markov moves for the uncoded 8 × 8 MIMO system using 16-QAM.

the detection performance may be affected due to the finite constellation. Furthermore, as shown in (74), under the help of LLL reduction, the decoding radius of the independent MHK sampling is significantly strengthened by a larger size of $\min_i \|\widehat{\mathbf{b}}_i\|$, thereby leading to a much better decoding performance. As a comparison, LLL reduction is applied in Gibbs sampling as a preprocessing stage to yield the high quality initial starting point. Additionally, compared to independent MHK, further decoding gain can be obtained by the independent MTMK algorithm, where cases with $k = 5$ and $k = 10$ trial samples are illustrated respectively.

On the other hand, in Fig. 6, the BERs of MCMC sampling detectors are evaluated against the number of Markov moves (i.e., iterations) in a 8 × 8 uncoded MIMO system with 16-QAM. The SNR is fixed as $E_b/N_0 = 15$ dB. Clearly, the performances of all the MCMC detectors improve with the number of Markov moves. Meanwhile, with the same number of Markov moves, a substantial performance gain is obtained by LLL reduction. By increasing number of trial samples, better decoding performance can be obtained by the independent MTMK algorithm due to a larger decoding radius shown in (93).

VIII. Conclusion

In this paper, the MCMC-based lattice Gaussian sampling was studied in full details. The spectral gap of the transition matrix of the independent MHK algorithm was derived and analyzed, which leads to a tractable exponential convergence rate of the Markov chain. A comparison with the extensions to Peikert's algorithm and rejection sampling illustrated the advantages of independent MHK. With the tractable mixing time, the decoding complexity of BDD using MCMC was derived and a trade-off between the decoding radius and complexity was established. The potential of MCMC was further demonstrated in trapdoor sampling. After that, by exploiting the potential of trial samples, the independent MTMK algorithm was proposed to enhance the convergence. It supports parallel implementation due to the independent proposal distribution, thus making independent MTMK algorithm promising in practice.

Acknowledgment

The authors would like to thank Dr. Thomas Prest for helpful discussions.

Appendix I
Proof of Theorem 6

Proof: To begin with, let us take a careful look at the term $\min\{\cdot,\cdot\}$ from (86). Here, for ease of presentation, we define

$$A = \sum_{\mathbf{y}_{1:c-1}\in\mathbb{Z}^n}\ \sum_{\mathbf{y}_{c+1:k}\in\mathbb{Z}^n} \left\{ \frac{\prod_{j=1,j\neq c}^{k} q(\mathbf{y}_j)}{\omega(\mathbf{y}_c)+\sum_{j=1,j\neq c}^{k}\omega(\mathbf{y}_j)} \right\} \quad (98)$$

and

$$B = \sum_{\mathbf{y}_{1:c-1}\in\mathbb{Z}^n}\ \sum_{\mathbf{y}_{c+1:k}\in\mathbb{Z}^n} \left\{ \frac{\prod_{j=1,j\neq c}^{k} q(\mathbf{y}_j)}{\omega(\mathbf{x})+\sum_{j=1,j\neq c}^{k}\omega(\mathbf{y}_j)} \right\}. \quad (99)$$

Meanwhile, because the k trial samples from the proposal distribution $q(\cdot)$ are independent of each other, a set Ξ is defined which contains the $k-1$ trial samples $\mathbf{y}_j$, $j \neq c$.

Then we can express A and B as

$$A = \sum_{\Xi} Q(\Xi)\cdot\frac{1}{\omega(\mathbf{y}_c)+\varpi(\Xi)} = \sum_{\Xi} Q(\Xi)\cdot F_A(\Xi) \quad (100)$$

and

$$B = \sum_{\Xi} Q(\Xi)\cdot\frac{1}{\omega(\mathbf{x})+\varpi(\Xi)} = \sum_{\Xi} Q(\Xi)\cdot F_B(\Xi). \quad (101)$$

Here, $Q(\Xi) = \prod_{j=1,j\neq c}^{k} q(\mathbf{y}_j)$ represents a probability distribution that takes all $q(\mathbf{y}_j)$, $j \neq c$ into account as a whole.

On the other hand, $F_A(\Xi)$ and $F_B(\Xi)$ stand for the functions about Ξ, namely,

$$F_A(\Xi) = \frac{1}{\omega(\mathbf{y}_c) + \varpi(\Xi)} \tag{102}$$

and

$$F_B(\Xi) = \frac{1}{\omega(\mathbf{x}) + \varpi(\Xi)}, \tag{103}$$

where

$$\varpi(\Xi) = \sum_{j=1, j\neq c}^{k} \omega(\mathbf{y}_j). \tag{104}$$

Now, let us focus on the term A, and we arrive at

$$\begin{aligned} A &= \sum_{\Xi} Q(\Xi) \cdot F_A(\Xi) \\ &= \mathbb{E}_{Q(\Xi)}[F_A(\Xi)] \\ &\overset{(h)}{\geq} \frac{1}{\mathbb{E}_{Q(\Xi)}[\omega(\mathbf{y}_c) + \varpi(\Xi)]} \\ &= \frac{1}{\omega(\mathbf{y}_c) + \mathbb{E}_{Q(\Xi)}[\varpi(\Xi)]} \\ &\overset{(i)}{=} \frac{1}{k-1+\omega(\mathbf{y}_c)}. \end{aligned} \tag{105}$$

Here, $\mathbb{E}_{u(x)}[v(x)]$ represents the expectation of function $v(x)$ while x is sampled from the distribution $u(x)$, (h) comes from the *Jensen's inequality* in the multi-variable case. Moreover, thanks to the $k-1$ independent samples from $q(\cdot)$, (i) follows the derivations shown below,

$$\begin{aligned} \mathbb{E}_{Q(\Xi)}[\varpi(\Xi)] &= (k-1) \cdot \mathbb{E}_{q(\mathbf{y}_j)}[\omega(\mathbf{y}_j)] \\ &= (k-1) \cdot \sum_{\mathbf{y}_j \in \mathbb{Z}^n} q(\mathbf{y}_j) \cdot \omega(\mathbf{y}_j) \\ &= (k-1) \cdot \sum_{\mathbf{y}_j \in \mathbb{Z}^n} \pi(\mathbf{y}_j) \\ &= k-1. \end{aligned} \tag{106}$$

Similar to A, we can rewrite B as

$$B \geq \frac{1}{k-1+\omega(\mathbf{x})}. \tag{107}$$

Therefore, from (105) and (107), we get

$$\begin{aligned} P(\mathbf{x}, \mathbf{y} = \mathbf{y}_c) &= k \cdot p(\mathbf{y}_c|\mathbf{x}, c) \\ &= k \cdot \pi(\mathbf{y}_c) \cdot \min\{A, B\} \\ &\geq \pi(\mathbf{y}_c) \cdot \min\left\{\frac{k}{k-1+\omega(\mathbf{y}_c)}, \frac{k}{k-1+\omega(\mathbf{x})}\right\} \\ &\geq \pi(\mathbf{y}_c) \cdot \frac{k}{k-1+\omega_{\max}(\mathbf{x})} \\ &\geq \pi(\mathbf{y}_c) \cdot \frac{k}{k-1+\frac{1}{\delta}} \\ &= \delta_{\text{MTM}} \cdot \pi(\mathbf{y}_c), \end{aligned} \tag{108}$$

where $\delta_{\text{MTM}} = k/(k-1+\frac{1}{\delta})$ and

$$\begin{aligned} \omega_{\max}(\mathbf{x}) &\triangleq \sup \omega(\mathbf{x}) = \sup \frac{\pi(\mathbf{x})}{q(\mathbf{x})} \\ &\leq \frac{1}{\delta} \end{aligned} \tag{109}$$

for $\mathbf{x} \in \mathbb{Z}^n$ from (22) in Lemma 1. From (108), it is straightforward to see that all the Markov transitions have a component of size δ_{MTM} in common. Then, uniform ergodicity can be easily demonstrated through spectral gap or coupling technique, which is omitted here for simplicity. ■

References

[1] W. Banaszczyk, "New bounds in some transference theorems in the geometry of numbers," *Math. Ann.*, vol. 296, no. 1, pp. 625–635, Dec. 1993.

[2] G. D. Forney, "Multidimensional constellations. II. Voronoi constellations," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 6, pp. 941–958, Aug. 1989.

[3] F. R. Kschischang and S. Pasupathy, "Optimal nonuniform signaling for Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 913–929, May 1993.

[4] C. Ling and J.-C. Belfiore, "Achieving AWGN channel capacity with lattice Gaussian coding," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5918–5929, Oct. 2014.

[5] C. Ling, L. Luzzi, J.-C. Belfiore, and D. Stehlé, "Semantically secure lattice codes for the Gaussian wiretap channel," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 6399–6416, Oct. 2014.

[6] H. Mirghasemi and J. C. Belfiore, "Lattice code design criterion for MIMO wiretap channels," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Oct. 2015, pp. 277–281.

[7] S. Vatedka, N. Kashyap, and A. Thangaraj, "Secure compute-and-forward in a bidirectional relay," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2531–2556, May 2015.

[8] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," in *Proc. Ann. Symp. Found. Comput. Sci.*, Rome, Italy, Oct. 2004, pp. 372–381.

[9] O. Regev, "On lattice, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, p. 34, Sep. 2009.

[10] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Dept. Comput. Sci. Stanford Univ., Stanford, CA, USA, 2009.

[11] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 40th Ann. ACM Symp. Theory Comput.*, Victoria, BC, Canada, May 2008, pp. 197–206.

[12] D. Aggarwal, D. Dadush, O. Regev, and N. Stephens-Davidowitz, "Solving the shortest vector problem in 2^n time via discrete Gaussian sampling," in *Proc. STOC*, 2015, pp. 733–742.

[13] D. Aggarwal, D. Dadush, and N. Stephens-Davidowitz, "Solving the closest vector problem in 2^n time—The discrete Gaussian strikes again!" in *Proc. FOCS*, Oct. 2015, pp. 563–582.

[14] A. Campello and J.-C. Belfiore, "Sampling algorithms for lattice Gaussian codes," in *Proc. Int. Zurich Seminar Commun. (IZS)*, Zurich, Switzerland, 2016, pp. 1–5.

[15] P. Klein, "Finding the closest lattice vector when it's unusually close," in *Proc. 11st Ann. ACM-SIAM Symp. Discrete Algorithms*, Feb. 2000, pp. 937–941.

[16] C. Peikert, "An efficient and parallel Gaussian sampler for lattices," in *Proc. Ann. Cryptol. Conf.*, Aug. 2010, pp. 80–97.

[17] P. Kirchner and P.-A. Fouque, "Time-memory trade-off for lattice enumeration in a ball," IACR Cryptol. ePrint Arch., Tech. Rep. 2016:222, 2016.

[18] Z. Wang and C. Ling, "On the geometric ergodicity of metropolis-Hastings algorithms for lattice Gaussian sampling," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 738–751, Feb. 2018.

[19] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains Mixing Time*. Providence, RI, USA: AMS, 2008.

[20] Z. Wang, C. Ling, and G. Hanrot, "Markov chain Monte Carlo algorithms for lattice Gaussian sampling," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, Jun. 2014, pp. 1489–1493.

[21] Y.-K. Liu, V. Lyubashevsky, and D. Micciancio, "On bounded distance decoding for general lattices," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Berlin, Germany: Springer, 2006, pp. 450–461.

[22] D. Dadush, O. Regev, and N. Stephens-Davidowitz, "On the closest vector problem with a distance guarantee," in *Proc. IEEE 29th Conf. Comput. Complex. (CCC)*, Jun. 2014, pp. 98–109.

[23] G. Hanrot, X. Pujol, and D. Stehlé, "Algorithms for the shortest and closest lattice vector problems," in *Coding and Cryptology*, Y. M. Chee *et al.*, Eds. Berlin, Germany: Springer, 2011, pp. 159–190.

[24] T. L. Marzetta, “Noncooperative cellular wireless with unlimited numbers of base station antennas,” *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[25] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for next generation wireless systems,” *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[26] F. Rusek *et al.*, “Scaling up MIMO: Opportunities and challenges with very large arrays,” *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[27] B. Hassibi, M. Hansen, A. G. Dimakis, H. Alshamary, and W. Xu, “Optimized Markov chain Monte Carlo for signal detection in MIMO systems: An analysis of the stationary distribution and mixing time,” *IEEE Trans. Signal Process.*, vol. 62, no. 17, pp. 4436–4450, Sep. 2014.

[28] T. Datta, N. A. Kumar, A. Chockalingam, and B. S. Rajan, “A novel Monte-Carlo-sampling-based receiver for large-scale uplink multiuser MIMO systems,” *IEEE Trans. Veh. Technol.*, vol. 62, no. 7, pp. 3019–3038, Sep. 2013.

[29] B. Farhang-Boroujeny, H. Zhu, and Z. Shi, “Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems,” *IEEE Trans. Signal Process.*, vol. 54, no. 5, pp. 1896–1909, May 2006.

[30] R. Chen, J. S. Liu, and X. Wang, “Convergence analyses and comparisons of Markov chain Monte Carlo algorithms in digital communications,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 255–270, Feb. 2002.

[31] P. Aggarwal and X. Wang, “Multilevel sequential Monte Carlo algorithms for MIMO demodulation,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 2, pp. 750–758, Feb. 2007.

[32] H. Zhu, B. Farhang-Boroujeny, and R.-R. Chen, “On performance of sphere decoding and Markov chain Monte Carlo detection methods,” *IEEE Signal Process. Lett.*, vol. 12, no. 10, pp. 669–672, 2005.

[33] S. Liu, C. Ling, and D. Stehle “Decoding by sampling: A randomized lattice algorithm for bounded distance decoding,” *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 5933–5945, Sep. 2011.

[34] N. Stephens-Davidowitz. (Jun. 2015). “Discrete Gaussian sampling reduces to CVP and SVP.” [Online]. Available: https://arxiv.org/abs/1506.07490

[35] Z. Wang, S. Liu, and C. Ling, “Decoding by sampling—Part II: Derandomization and soft-output decoding,” *IEEE Trans. Commun.*, vol. 61, no. 11, pp. 4630–4639, Nov. 2013.

[36] W. K. Hastings, “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, Apr. 1970.

[37] D. Randall, “Rapidly mixing Markov chains with applications in computer science and physics,” *Comput. Sci. Eng.*, vol. 8, no. 2, pp. 30–41, Mar./Apr. 2006.

[38] L. Babai, “On lovász’ lattice reduction and the nearest lattice point problem,” *Combinatorica*, vol. 6, no. 1, pp. 1–13, Mar. 1986.

[39] B. D. Ripley, “Stochastic simulation,” in *Probability and Mathematical Statistics*. Hoboken, NJ, USA: Wiley, 1987.

[40] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, “Classical hardness of learning with errors,” in *Proc. 45th Annu. ACM Symp. Theory Comput.*, Jun. 2013, pp. 575–584.

[41] J. H. Conway and N. A. Sloane, *Sphere Packings, Lattices, and Groups*. New York, NY, USA: Springer-Verlag, 1998.

[42] A. K. Lenstra, H. W. Lenstra, and L. Lovász, “Factoring polynomials with rational coefficients,” *Math. Annalen*, vol. 261, no. 4, pp. 515–534, Dec. 1982.

[43] J. C. Lagarias, W. H. Lenstra, and C. P. Schnorr, “Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice,” *Combinatorica*, vol. 10, no. 4, pp. 333–348, Dec. 1990.

[44] T. Prest, “Gaussian sampling in lattice-based cryptography,” Ph.D. dissertation, ENS, Paris, France, 2015. [Online]. Available: https://tprest.github.io/Publications/ThomasPrestThesis.pdf

[45] P.-A. Fouque *et al.* (2017). *Falcon: Fast-Fourier Lattice-Based Compact Signatures Over NTRU*. Accessed: Jul. 23, 2018. [Online]. Available: https://falcon-sign.info/falcon.pdf

[46] J. Liu, F. Liang, and W. W. Hung, “The multiple-try method and local optimization in metropolis sampling,” *J. Amer. Statist. Assoc.*, vol. 95, no. 4, pp. 121–134, Aug. 2000.

[47] L. Martino and J. Read, “On the flexibility of the design of multiple try metropolis schemes,” *Comput. Statist.*, vol. 28, no. 6, pp. 2797–2823, Dec. 2013.

[48] R. Casarin, R. V. Craiu, and F. Leisen, “Interacting multiple try algorithms with different proposal distributions,” *Statist. Comput.*, vol. 23, no. 2, pp. 185–200, Mar. 2013.

[49] L. Martino and J. Corander. (2014). *On Multiple Try Schemes and the Particle Metropolis-Hastings Algorithm*. [Online]. Available: http://vixra.org/pdf/1409.0051v1.pdf

[50] L. Martino and F. Louzada, “Issues in the multiple try Metropolis mixing,” *Comput. Statist.*, vol. 32, no. 1, pp. 239–252, Mar 2017.

[51] A. Kumar, S. Chandrasekaran, A. Chockalingam, and B. S. Rajan, “Near-optimal large-MIMO detection using randomized MCMC and randomized search algorithms,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–5.

[52] B. M. Hochwald and S. T. Brink, “Achieving near-capacity on a multiple-antenna channel,” *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.

Zheng Wang received the B.S. degree in electronic and information engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2009, and the M.S. degree in communications from the Department of Electrical and Electronic Engineering, University of Manchester, Manchester, U.K., in 2010. He received the Ph.D degree in communication engineering from Imperial College London, UK, in 2015.

From 2015 to 2016 he served as a Research Associate at Imperial College London, UK and from 2016 to 2017 he was an senior engineer with Radio Access Network R&D division, Huawei Technologies Co.. He is currently an Assistant Professor at the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China. His current research interests include lattice methods for wireless communications, cognitive radio and physical layer security.

Cong Ling received the B.S. and M.S. degrees in electrical engineering from the Nanjing Institute of Communications Engineering, Nanjing, China, in 1995 and 1997, respectively, and the Ph.D. degree in electrical engineering from the Nanyang Technological University, Singapore, in 2005.

He is currently a Senior Lecturer in the Electrical and Electronic Engineering Department at Imperial College London. His research interests are coding, signal processing, and security. Before joining Imperial College, he had been on the faculties of Nanjing Institute of Communications Engineering and King’s College.

Dr. Ling is an Associate Editor of IEEE TRANSACTIONS ON COMMUNICATIONS. He has also served as an Associate Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.