Amostragem Gaussiana na Criptografia Baseada em Reticulados

Jheyne N. Ortiz[1], **Diego F. Aranha**[1], **Ricardo Dahab**[1]

[1]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Campinas – SP – Brasil

Abstract. *In Lattice-Based Cryptography, some cryptosystems require sampling lattice points and integers over a distribution that, conventionally, follows a Gaussian function. This work presents a constant-time implementation of the Knuth-Yao method for Gaussian sampling over integers. For Gaussian sampling over lattices, the results refer to the sampling over NTRU lattices. Our experiments target an Intel Ivybridge desktop and all implementations are described using C++ language with support of the NTL library of Victor Shoup.*

Resumo. *Na Criptografia Baseada em Reticulados, diversos esquemas requerem a amostragem de vetores de um reticulado e de inteiros seguindo uma distribuição que, convencionalmente, é uma função Gaussiana. Este trabalho apresenta uma implementação com tempo de execução constante para o método Knuth-Yao apropriado à amostragem Gaussiana sobre os inteiros. Para a amostragem de vetores de um reticulado, os resultados se referem à amostragem sobre reticulados NTRU. Os experimentos têm como alvo um computador pessoal Intel Ivybridge e as implementações são descritas em linguagem C++ com aporte da biblioteca NTL de Victor Shoup.*

1. Introdução

Esquemas sobre reticulados como, por exemplo, Encriptação Baseada em Identidades, funções de resumo seguidas de assinatura e Encriptação Baseada em Atributos requerem a amostragem de vetores de um reticulado. A amostragem sobre os inteiros é um passo em métodos de amostragem de pontos de um reticulado além de ser necessária no esquema de encriptação [Lyubashevsky et al. 2012] e no esquema de assinatura [Ducas et al. 2013] baseados no problema Ring-LWE. Usualmente, ambas amostragens sobre reticulados e sobre os inteiros seguem uma distribuição Gaussiana.

No esquema de encriptação baseado no problema Ring-LWE, a amostragem de ruídos conforme a Gaussiana ocorre na fase de encriptação que comumente é implementada em um dispositivo vulnerável a ataques por canais laterais. Similarmente, como demonstrado por [Bruinderink et al. 2016], o esquema de assinatura BLISS é suscetível a ataques por canais laterais bem como o amostrador Gaussiano utilizado na fase de assinatura. Por isso, é importante que implementações de tais algoritmos sejam não somente eficientes, como também incorporem técnicas de proteção contra ataques de canais laterais. Neste trabalho, somente ataques por canais de tempo são tratados.

Objetivos. Neste trabalho, os objetivos consistem na implementação de um algoritmo específico para amostragem Gaussiana sobre reticulados NTRU bem como a implementação em software de um algoritmo para amostragem sobre os inteiros tendo em vista eficiência e resistência contra ataques por canais de tempo.

2. Implementação de Amostragem de Gaussianas Discretas

A função Gaussiana determina a probabilidade de amostragem dos elementos de um determinado domínio. Para o conjunto dos inteiros, a distribuição Gaussiana, parametrizada pelo centro $c \in \mathbb{R}$ e um fator $\sigma \in \mathbb{R}^+$, é uma função que determina a probabilidade de um valor inteiro x ser amostrado em relação aos demais, conforme a Equação 1.

$$\mathcal{D}_{\mathbb{Z},\sigma,c}(x) = \frac{\rho_{\sigma,c}(x)}{\sum_{y=-\infty}^{\infty} \rho_{\sigma,c}(y)}, \text{ com } \rho_{\sigma,c}(x) := \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{1}$$

No contexto deste trabalho, devido à estrutura isométrica da base curta que gera o reticulado NTRU, a ortogonalização de Gram-Schmidt da base do reticulado é obtida através da chamada ao procedimento BLOCK-GSO proposto por [Lyubashevsky and Prest 2015]. O método híbrido de [Ducas and Prest 2015] para amostragem Gaussiana de pontos de um reticulado, denotado HYBRID-SAMPLER, é otimizado para amostragem de reticulados sobre o anel de inteiros de um corpo ciclotômico e consiste em uma adaptação do método de Klein [Gentry et al. 2008] sobre anéis. Em uma etapa intermediária, o método HYBRID-SAMPLER requer inteiros amostrados conforme uma distribuição Gaussiana, tarefa desempenhada pelo algoritmo Knuth-Yao [Knuth and Yao 1976].

O algoritmo Knuth-Yao utiliza o conceito de árvore DDG (*Discrete Distribution Generating tree*) para gerar uma amostra, que representa o rótulo de um nó desta árvore. A árvore DDG é descrita pela matriz de probabilidades $\mathbf{P} \in \mathbb{Z}_2^{2t\sigma\times\lambda}$, que consiste na expansão binária das probabilidades dos elementos no intervalo $[c-t\sigma, c+t\sigma]\cap\mathbb{Z}$. Sendo um método finito, as probabilidades são calculadas com λ bits de precisão e podem ser armazenadas preliminarmente para uso na amostragem.

Grosso modo, o método Knuth-Yao consiste em percorrer a matriz de probabilidades computando a distância entre o nó sendo visitado e o nó mais à direita na árvore DDG. Para que a implementação tenha tempo de execução independente da entrada, a execução do algoritmo é extrapolada para o pior caso sem que haja sobre-escrita de amostras. Neste caso, uma solução é a combinação de uma somatória com operações aritméticas que substituem desvios condicionais inerentes ao algoritmo. As operações binárias testam se a distância, que determina que uma amostra foi encontrada, é igual a -1, utilizando o Código 1 e definem o valor lógico da variável de controle `enable` como verdadeiro em caso afirmativo.

Código 1. Teste de igualdade em tempo constante.

```
unsigned isEqual(int x, int y) {
  unsigned equal = (x-y);
  return (1 ^ ((equal | -equal) >> 31) & 1);
}
```

Uma variável lógica, `hit`, guarda a informação de que uma amostra já foi aceita. Com esses dois valores inteiros, um seletor determina se o valor inválido `invalidSample`, que se encontra fora do intervalo de amostragem $[c-t\sigma, c+t\sigma]\cap\mathbb{Z}$, ou o valor do índice da coluna `col`, que é o rótulo da amostra, será somado à variável de saída denotada por `S`. O seletor é uma função com tempo de execução constante para atribuição condicional que é determinada pela variável `bit` conforme o Código 2.

Código 2. Seletor para atribuição condicional.

```
int Select(int a, int b, unsigned bit) {
    unsigned mask = -bit;
    return (mask & (a ^ b)) ^ a;
}
```

No término do algoritmo, o valor de `S` é da forma $(k \cdot \texttt{invalidSample} + \texttt{col})$, com $k \in \mathbb{Z}$. Assim sendo, a redução módulo `invalidSample` remove o erro e a operação $S = S - (t + \sigma) + c$ obtém o valor da amostra corrigido.

3. Resultados Experimentais

A Figura 1 apresenta os tempos de execução em segundos para a amostragem de coeficientes inteiros para um polinômio com diferentes dimensões utilizando o método Knuth-Yao. Na versão em tempo constante do método Knuth-Yao, denotada por Knuth-YaoC, a execução prossegue é extrapolada para o pior caso e, então, toda a estrutura que representa a matriz de probabilidades é visitada. Portanto, esse é o principal fator na perda de desempenho da versão Knuth-YaoC, cuja redução no tempo de execução pode chegar em até sete vezes em comparação com a implementação direta do método Knuth-Yao quando o polinômio tem 8192 coeficientes. Todos os resultados desta seção foram obtidos de uma máquina Intel Ivybridge com processador Intel®Core™i5-3570 @ 3.40 GHz e 8 GB de memória física.

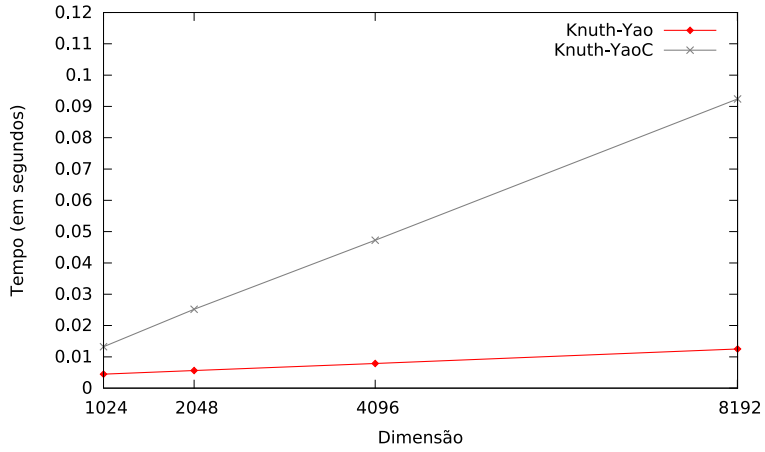

Figura 1. Comparação entre as implementações para o Knuth-Yao com $\sigma = 3,19$ e $\lambda = 107$.

A Tabela 1 descreve os tempos de execução em segundos para todas as etapas da amostragem utilizando a implementação do método híbrido de [Ducas and Prest 2015]. Os tempos de execução para o experimento sobre reticulados NTRU são a média de cem tempos de execução para os algoritmos KEYGEN, BLOCK-GSO, OFFLINE-KLEIN e OFFLINE-PEIKERT. Para a amostragem propriamente dita, os tempos são a média de dez execuções.

O algoritmo KEYGEN é responsável pela geração da base do reticulado NTRU $\mathbf{B} \in \mathbb{Z}^{2N \times 2N}$. Os algoritmos OFFLINE-KLEIN e OFFLINE-PEIKERT denotam as fases de pré-computação associadas aos algoritmos RING-KLEIN, que é instanciado como o método híbrido, e RING-PEIKERT, que fornece amostras do anel de inteiros $\mathcal{R}$ ao algoritmo RING-KLEIN, respectivamente.

	Dimensão (N)			
Algoritmo	**128**	**256**	**512**	**1024**
KeyGen	$0,10$	$0,44$	$2,20$	$11,02$
Block-GSO	$0,11$	$0,45$	$2,37$	$7,43$
Offline-Klein	$3,60$	$17,15$	$97,46$	$505,39$
Offline-Peikert	$3,19$	$5,60$	$20,94$	$82,37$
Hybrid-Sampler	$15,58$	$72,38$	$373,23$	$2042,71$

Tabela 1. Tempos de execução, em segundos, para o método híbrido com variação no valor de N.

Em suma, os tempos de execução para o método híbrido [Ducas and Prest 2015] representam a ineficiência de algoritmos para amostragem Gaussiana sobre reticulados. Por outro lado, as fases de pré-computação são invocadas com baixa frequência uma vez que a base do reticulado usualmente é a chave privada mestre do sistema criptográfico. Os resultados na Figura 1 ilustram a dificuldade em desenvolver implementações eficientes e seguras contra ataques por canais laterais para métodos para amostragem Gaussiana sobre os inteiros. A necessidade de extrapolar a execução do método Knuth-Yao para o pior caso visando uma implementação com tempo de execução constante reflete o balanço entre segurança e eficiência na implementação de métodos para amostragem Gaussiana discreta.

Referências

Bruinderink, L. G., Hülsing, A., Lange, T., and Yarom, Y. (2016). Flush, Gauss, and Reload – A Cache Attack on the BLISS Lattice-Based Signature Scheme. Cryptology ePrint Archive, Report 2016/300. `http://eprint.iacr.org/`.

Ducas, L., Durmus, A., Lepoint, T., and Lyubashevsky, V. (2013). *Advances in Cryptology – CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, chapter Lattice Signatures and Bimodal Gaussians, pages 40–56. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ducas, L. and Prest, T. (2015). A Hybrid Gaussian Sampler for Lattices over Rings. Cryptology ePrint Archive, Report 2015/660. `http://eprint.iacr.org/`.

Gentry, C., Peikert, C., and Vaikuntanathan, V. (2008). Trapdoors for Hard Lattices and New Cryptographic Constructions. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 197–206, New York, NY, USA. ACM.

Knuth, D. and Yao, A. (1976). *Algorithms and Complexity: New Directions and Recent Results*, chapter The Complexity of Nonuniform Random Number Generation. Academic Press, New York, j. f. traub edition.

Lyubashevsky, V., Peikert, C., and Regev, O. (2012). On Ideal Lattices and Learning with Errors Over Rings. Cryptology ePrint Archive, Report 2012/230. `http://eprint.iacr.org/`.

Lyubashevsky, V. and Prest, T. (2015). Quadratic Time, Linear Space Algorithms for Gram-Schmidt Orthogonalization and Gaussian Sampling in Structured Lattices. Cryptology ePrint Archive, Report 2015/257. `http://eprint.iacr.org/`.