Gaussian Sampling Precision and Information Leakage in Lattice Cryptography

Markku-Juhani O. Saarinen *

Centre for Secure Information Technologies (CSIT)
ECIT, Queen's University Belfast, UK
m.saarinen@qub.ac.uk

Abstract. Security parameters and attack countermeasures for Lattice-based cryptosystems have not yet matured to the level that we now expect from RSA and Elliptic Curve implementations. Many modern Ring-LWE and other lattice-based public key algorithms require high precision random sampling from the Discrete Gaussian distribution. The sampling procedure often represents the biggest implementation bottleneck due to its memory and computational requirements. We examine the stated requirements of precision for Gaussian samplers, where statistical distance to the theoretical distribution is typically expected to be below 2^{-90} or 2^{-128} for 90 or 128 "bit" security level. We argue that such precision is excessive and give precise theoretical arguments why half of the precision of the security parameter is almost always sufficient. This leads to faster and more compact implementations; almost halving implementation size in both hardware and software. We observe that many of the proposed algorithms for discrete Gaussian sampling may leak significant amounts of secret information in easily mounted timing attacks. We further offer new recommendations for practical samplers.

Keywords: Gaussian Sampling, Timing attacks, Lattice Side-Channel Attacks, Quantum Resistant Cryptography.

1 Introduction

With the recent announcement of a pending quantum-resistant suite of cryptographic algorithms for U.S. Government use [1,20], renewed interest has been placed on various Quantum-Resistant Cryptography (QRC) primitives that do not rely on (Elliptic Curve) Discrete Logarithm or Integer Factorisation (RSA) problems. New quantum-resistant algorithms can be soon expected to enter the realm of payment systems, identification methods, and mainstream communications security. However, in the words of the CESG team that created – and then broke via quantum methods – a lattice-based algorithm named Soliloquy [5]:

> One conclusion of this work is that designing quantum-resistant cryptography is a difficult task. [..] As of late 2014, when novel types of quantum-resistant cryptography are being developed for real-world deployment, we caution that much care and patience will be required to ensure that each design receives a thorough security assessment.

* This work was funded by the European Union H2020 SAFEcrypto project (grant no. 644729).

Most modern Ring-LWE and other lattice-based cryptographic algorithms require variables to be sampled from the Discrete Gaussian distribution. For many implementations the sampling procedure represents the biggest performance bottleneck due to its memory or computational requirements. This is especially the case for embedded or lightweight targets such as smart cards [4,7,8,11,16,25].

Structure of this paper and our contributions. In Sections 2 and 3 we discuss the discrete Gaussian distribution, sampling, and precision. In Section 4 we argue that the common requirements for precision in Gaussian sampling are excessive; essentially only half of the bits are required, enabling faster and more compact implementations. Section 5 argues for constant-time samplers. We conclude with sampler recommendations in Section 6.

2 The Discrete Gaussian Distribution

For simplicity we use zero mean $c = 0$ throughout this paper. Discrete Gaussian distributions D_σ are then defined solely by deviation parameter σ. The probabilities for $x \in \mathbb{Z}$ (Figure 1) are proportional to

$$f_\sigma(x) = e^{-\frac{x^2}{2\sigma^2}}. \tag{1}$$

We define a one-sided cumulative function $S_\sigma(b)$ for $b \geq 0$ as $S_\sigma(0) = 0$,

$$S_\sigma(b) = \sum_{k=-b+1}^{b-1} f_\sigma(k) \;\; for \;\; b \geq 1. \tag{2}$$

Due to symmetry $f_\sigma(x) = f_\sigma(-x)$ we have $S_\sigma(b) = 1 + 2\sum_{k=1}^{b-1} e^{-\frac{k^2}{2\sigma^2}}$ for $b \geq 1$. Since the limit for total scaling mass $S_\sigma(\infty)$ is very closely approximated by $\sigma\sqrt{2\pi}$ when σ grows, we may use this scaling value in practical computations. Let P be a discrete random variable on sample space $\mathbb{Z}$. The probability mass for any $x \in \mathbb{Z}$ is

$$\rho_\sigma(x) = Pr(P = x) = \frac{f_\sigma(x)}{S_\sigma(\infty)} \approx \frac{e^{-\frac{x^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}. \tag{3}$$

Discrete Sampling. Sampling algorithms convert unbiased random bits into non-uniformly distributed integer samples from a given distribution. In case of Gaussian distribution, this is fully characterised by the deviation parameter σ. There is no closed, non-approximate algebraic formula for sampling that does not require evaluation of integrals or series. Hence specialist algorithms are needed.

Sampling Precision. Let P and Q be two discrete random variables on the same domain. We use shorthand $P(x) = \Pr(P = x)$ and $Q(x) = \Pr(Q = x)$ for their distributions. The *total variation distance* δ between P and Q is defined as:

$$\epsilon = \delta(P, Q) = \frac{1}{2}||P - Q||_1 = \frac{1}{2}\sum_x |P(x) - Q(x)|. \tag{4}$$

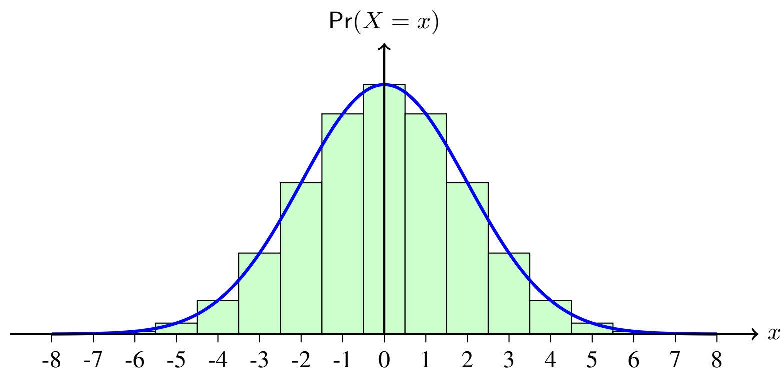

Fig. 1. The Discrete Gaussian distribution D_σ (Equation 3) is defined for all $x \in \mathbb{Z}$ and satisfies $\sum_{x=-\infty}^{\infty} \rho_\sigma(x) = 1$. The green discrete bars illustrate the probability mass whereas the blue line is the corresponding continuous probability density function.

If we set P as the theoretical distribution ("perfect sampler") and Q as the actually generated distribution, we may use the statistical distance between the two to quantify the quality of the Q sampler.

Tail cutting. In tail cutting we ignore the "tail" portion of distribution with $|x| > \beta\sigma$ that has very small total mass, under target distance ϵ or related precision $2^{-\lambda}$. A typical tail cutting bound for cryptographic applications is $\beta = 13.2$ as it is easy to show that for any $\sigma \geq 1$ we have a negligible tail mass:

$$1 - \frac{S_\sigma(13.2\sigma)}{S_\sigma(\infty)} < 2^{-128}. \tag{5}$$

It is easy to see that $\epsilon < 2^{-\lambda}\beta\sigma$ where $\beta\sigma$ is the tail cutting bound. Figure 2 illustrates the relationship between sampling precision and tail cutting bound.

Required distance. It has been widely assumed that for cryptographic applications the sampling distance should be roughly the inverse of the security parameter [9]:

> It is necessary for the rigorous security analysis that the statistical difference between the actual distribution being sampled and the theoretical distribution (as used in the security proof) is negligible, say around 2^{-90} to 2^{-128}.

This is also the precision typically now being implemented (See e.g. [3,6,15,23]). In this paper we set out to show that such precision is essentially unnecessary since *no algorithm* will be able to detect the difference from the non-tail portion of samples; only about half of this precision is actually required in almost all cases.

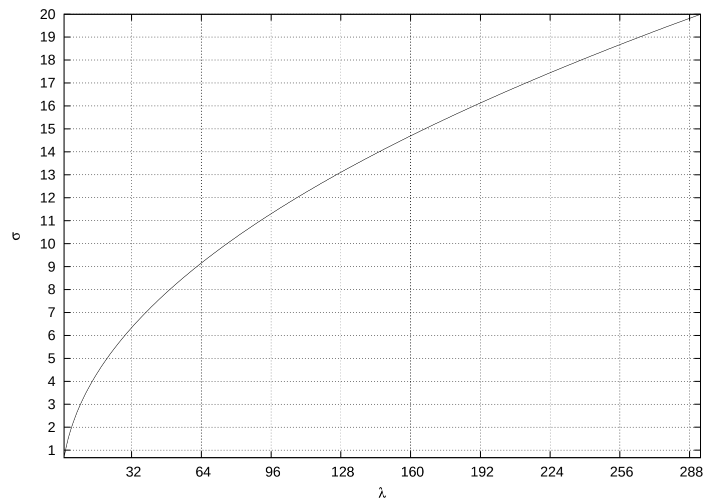

Fig. 2. Sampling precision $2^{-\lambda}$ and tail cutting bound β in multiples of deviation parameter σ.

Other metrics and related work. Recently, proofs of some Lattice based schemes have been reworked using Rényi distance [2] to require less precision in implementations. Furthermore, Pöppelmann, Ducas, and Güneysu used the Kullback-Leibler divergence to reduce storage requirements in a hardware sampler implementation [23].

3 Approximate Sampling

Perfect sampler. First consider an arbitrary-precision sampler that converts an uniformly random number $x \in \mathbb{R}$, $0 \leq x < 1$ into the Discrete Gaussian distribution by finding the "bin" $i \in \mathbb{Z}, i \geq 0$ in Cumulative Distribution Table (CDT) satisfying

$$\frac{S_\sigma(i)}{S_\sigma(\infty)} \leq x < \frac{S_\sigma(i+1)}{S_\sigma(\infty)}. \tag{6}$$

If $i = 0$, output 0, otherwise i or $-i$, depending on a single additional random bit. It is easy to show that this creates samples exactly from the distribution D_σ.

Approximate sampler with precision $\boldsymbol{\lambda}$. We define an approximation where we use a λ-bit uniform random integer $j \in \mathbb{Z}$, $0 \leq j < 2^\lambda$ to approximate the discrete Gaussian Distribution. Here we again find the correct bin i via

$$\frac{S_\sigma(i)}{S_\sigma(\infty)} \leq 2^{-\lambda} j < \frac{S_\sigma(i+1)}{S_\sigma(\infty)}. \tag{7}$$

Now for a sampling error to occur at all, $2^{-\lambda} j$ must fit exactly on one of the threshold values i so that λ leftmost bits match with the cumulative distribution function:

$$2^{-\lambda} j \leq \frac{S_\sigma(i)}{S_\sigma(\infty)} \leq 2^{-\lambda}(j+1). \tag{8}$$

In practice, the probability of sampling error is almost directly proportional to sampling precision $2^{-\lambda}$ and total variation distance ϵ (Equation 4). See Figure 3 for an illustration of sampling error and the resulting statistical distance.

Binary Search in Cumulative Distribution Table. Since each half of the distribution function is monotonically decreasing (or increasing), we may perform a binary search on it in with at most $\lceil \log_2 n \rceil < \lambda$ steps, where n is the number of entries in the table (integers with greater than "tail cutting" probability). This approach is widely used in real-life implementations [3,23].

Other Gaussian Sampling Algorithms. High precision non-uniform continuous random sampling is a classic problem [19]. Many of the algorithms of the continuous case also apply to discrete cryptographic applications. Methods such as Inversion Sampling [22], Knuth-Yao Sampling [9,13], The Ziggurat Method [4,10,17,18], Kahn-Karney Sampling [12], and "Bernoulli" sampling [7] have also been proposed for lattice cryptography. For more (non-cryptographic) methods, see [26].

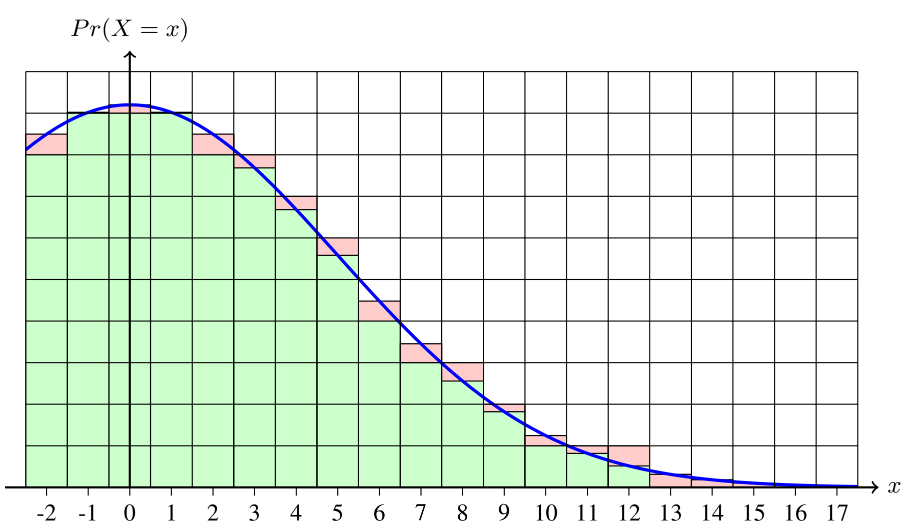

Fig. 3. The discrete Gaussian distribution and sampling precision. The red area illustrates the total variation distance between the ideal distribution and sampling with the precision of the grid.

4 Distinguishing Distributions

When determining the appropriate sampling precision λ, we are led to ask "*What is the minimum statistical distance or precision λ that can be detected by an adversary?*". If an approximation cannot be distinguished from true distribution with reasonable effort, there should not be any reason not to use it.

Biased coins and distribution identity testing. In distribution testing literature the task of determining whether a given black-box distribution is equal to a known distribution is known as *identity testing*.

Example 1. To get an intuitive feel for this problem one may consider the classical question "*how many coin tosses are required to determine that a coin is biased ?*" We set the probability of heads/tails as $\frac{1 \pm \epsilon}{2}$. Incidentally, this bias ϵ is equal to total variation distance (Equation 4) in this case. Consider the χ^2 test-statistic for N coin tosses of which M are heads:

$$\chi^2 = \frac{1}{N}\big(2M - N\big)^2 \tag{9}$$

Substituting our known bias $M = N\frac{1+\epsilon}{2}$ and solving N for arbitrary confidence level χ^2 we obtain

$$N = \frac{\chi^2}{\epsilon^2}. \tag{10}$$

Note the ϵ^2 term. We see that for a coin that has $\epsilon = 0.01$ bias (50.5 % heads and 49.5 % tails or vice versa), 9360 coin tosses would be required to distinguish it from a fair coin even at the very low $2/3$ confidence level (corresponding in the single degree of freedom cumulative distribution function to $\chi^2 = 0.936$).

This is of course a gross simplification since the χ^2 statistic is not an optimal statistical testing tool. However, exact analysis using the binomial distribution supports these findings. Equivalent tight bounds have also been found for general uniformity testing; Paninski [21] offers $\Theta(\frac{\sqrt{n}}{\epsilon^2})$ (upper and lower) complexity where n the size of the discrete domain. Generally speaking the number of required probes grows quadratically to the inverse of the bias in the uniform case; $O(\epsilon^{-2})$ probes are required.

Tight bounds for distribution identity testing. We quote the following definitions and a recent result (Theorem 1 from [27,28]) which offers very tight asymptotic bounds for the sample complexity of distribution identity testing:

Definition 1. *For a distribution P, let $P^{-\max}$ denote the vector of probabilities obtained by removing the entry corresponding to the element of largest probability.*

Definition 2. *For a vector P and $\epsilon > 0$, define $P_{-\epsilon}$ as the vector obtained from P by iteratively removing the smallest domain elements and stopping before more than ϵ probability mass is removed.*

We observe that Definition 1 corresponds to removing the distribution centre ($c = 0$) and Definition 2 corresponds to tail cutting (Section 2). Therefore these cases need to be handled specially.

Theorem 1 (Theorem 1 of [27,28]). *There exist constants c_1, c_2 such that for any $\epsilon > 0$ and any known distribution P, for any unknown distribution Q on the same domain, our tester will distinguish $P = Q$ from $||P - Q||_1 \geq \epsilon$ with probability $2/3$ when run on a set of at least $c_1 \frac{||P^{-\max}_{-\epsilon/16}||_{2/3}}{\epsilon^2}$ samples and no tester can do this task with probability at least $2/3$ with a set of fewer than $c_2 \frac{||P^{-\max}_{-\epsilon}||_{2/3}}{\epsilon^2}$ samples.*

The tight $\Theta\big(\frac{||p||_{2/3}}{\epsilon^2}\big)$ sample complexity of "Valiant-Valiant" (Theorem 1) not only implies bounds for traditional computational complexity, but also the minimum oracle query complexity of attack regardless of the computational model used. This is essentially an information theoretic bound.

On binary hypothesis testing and randomized rounding. Consider a table of λ-precision approximations $T[0, 1, \ldots, 2^\lambda - 1]$:

$$T[i] = \left\lfloor 2^\lambda \frac{S_\sigma(i)}{S_\sigma(\infty)} \right\rfloor. \tag{11}$$

Theorem 1 requires distribution Q to be unknown but at most ϵ - distant from true distribution P. A static table at exactly ϵ will not be unknown to a distinguisher and will essentially yield a case of binary hypothesis testing. If the table is held in RAM, it is possible to randomize it by adding +1 to each entry during initialization with probability $\frac{1}{2}$; here one comes up with the precise case of an unknown static distribution that has maximum total variation distance $\epsilon < 2^{-\lambda}\beta\sigma$.

In practice we define the precision $2^{-\lambda}$ to have a few more bits of precision than corresponding ϵ; we are actually distinguishing a very large family of distributions from the true one. If an implementor still feels that this is a concern for some severely limited λ, rounding can be further randomized. If the condition of Equation 8 holds and the given random integer j matches all bits of $T[i]$, a randomized rounding sampler will output either i or $i + 1$, depending on an additional random bit.

The tail detector test and conjecture. We note that the potentially infinite tail spread of the Gaussian distribution makes the $P_{-\epsilon}$ term problematic. Indeed, with tail cut at ϵ level (tail mass of ϵ) one could simply test if any of the values of tail appear; such a "tail detector" test would have complexity $O(1/\epsilon)$. This problem is sidestepped by Theorem 1 and we and also ignore this special case in current work. We conjecture that lack of tail has only marginal effect on the entropy of random quantities and the security of the resulting cryptosystem. However, the security impact of tail cutting must be evaluated on case-by-base basis.

Recursive application of Theorem 1 on the tail. An inverse-CDT type generator "knows" when it is supposed to generate values from the tail; in a straightforward implementation the λ-bit random integer j is at the $2^\lambda - 1$ maximum or close to it. One can apply Theorem 1 recursively on the tail by defining P' as the tail portion of the main distribution, and adjusting ϵ' accordingly.

Example 2. Here $P' = P \setminus P_{-\epsilon/16}$ would be a natural choice. Corresponding adjusted precision would be $\epsilon' = \epsilon^2/16$. First step of such a sampling algorithm is to test if uniformly random j satisfies $j \geq 2^\lambda - r$ where r is relatively small. If this is a case, we randomize an another j' and utilize a search algorithm on a table of tail values. Otherwise we proceed normally with the main table. Overall required precision will still be λ bits but the two-step approach removes the problem of tail distinguishers. Naturally the condition makes constant-time implementation (Section 5) more difficult. Note that the secondary tail table will be invoked with very low probability; this part of the code and its tables are quite probably *never actually used* with the parameters proposed in Section 6. This is why we conjecture that it is unnecessary.

Impact on sampling precision in private key operations. In a lattice public key algorithm (such as Ring-LWE based encryption or signature algorithm), the bounds of Theorem 1 directly indicate (up to a constant factor) the number of times the private key oracle must be invoked before *any* algorithm, *quantum or non-quantum*, can determine whether the samples it uses were drawn from perfectly sampled distribution or from one with total variation distance ϵ to it. Since $O(\epsilon^{-2})$ probes are necessarily required, one can generally set the sampling precision to $\lambda = s/2$ where 2^s is the target security level. This greatly simplifies implementation in many cases.

5 The Timing Channel

A timing attack is the "classical" side-channel attack originally considered by Kocher in 1996 [14]. The attack measures the total execution time t required to create a signature and uses t to create forged signatures. Notably this leakage channel is available in challenge-response authentication protocols (essentially all authentication protocols). Most operational cryptosystems are now engineered to be resistant to this attack.

For most lattice sampling algorithms and implementations we have examined, the timing channel reveals large amounts of secret information. By comparison, rounding error in sampling is not really from any precise value; it is from a number which itself is chosen at random. It appears to be very difficult to use sampling quantisation errors in an attack [3,6,4,7,9,10,13,15,23,17,18].

Timing the Gaussian Sampler. If we consider the binary search sampling algorithm of Section 3, it becomes clear that the further you are from the centre $x = 0$, more comparisons are required in a binary search; other algorithms will also exit early when $|x|$ is small. Such early exit strategies for larger intervals (smaller norm) are used in virtually all samplers examined; Knuth-Yao Sampling [9,13], the Ziggurat Method [4,10,17,18], and "Bernoulli" sampling [7]

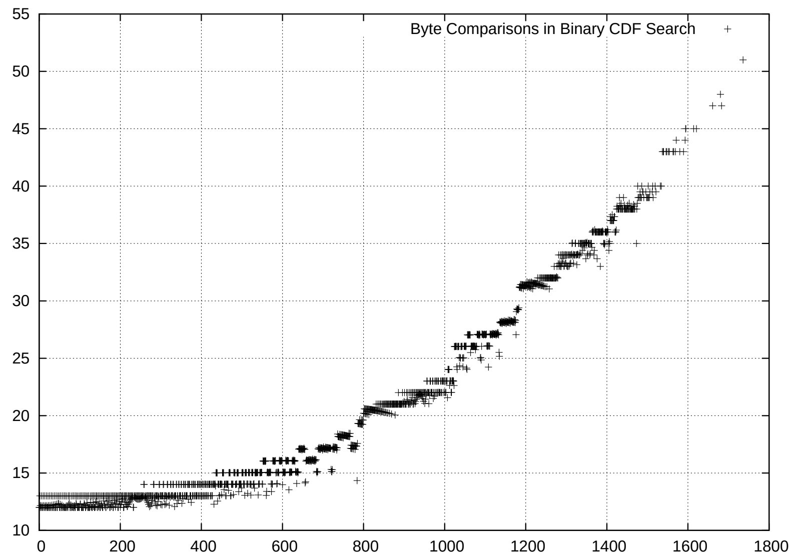

Fig. 4. Average number of byte comparisons (y axis) in a binary search on the Cumulative Distribution Function to find the correct value in the Discrete Gaussian Distribution (absolute value on x axis). For this graph $\sigma = 271.93$, $n = 10^9$. We see that fast runs on the sampler are much more likely to have a norm close to 0 and can be attacked in semi - exhaustive search fashion.

We experimented by generating one billion uniformly random numbers in the interval $[0, 1[$ and tested how many comparisons are required to find the correct discrete "bin" in the distribution. Furthermore, for more narrow intervals (larger $|x|$) more byte comparisons in `memcmp()` type algorithm are needed, further affecting execution time. Figure 4 shows the results. We observe that if the sampling algorithm terminates in a very short time, the norm of the resulting vector is small and be searched more easily.

Constant time Sampling. The simplicity of the binary search algorithm – coupled with our precision bound and tail cutting – allows us to easily find an exact upper bound for its running time. We will simply not terminate the search even when the correct "bin" is found but always run the comparisons to full precision (constant number of steps). The upper bound for number of comparisons is exactly $\lceil \log_2 n \rceil$ where n is the size of the CDT table.

It is possible to implement binary search without any conditional branches. If these are necessary, various conditional cases can be balanced with redundant operations or comparison masks can be used. Memory cache could now become the only source of variation. A constant time sampler was used in [3]. A side-channel resilient hardware sampler design based on Knuth-Yao was described in [24].

6 Conclusions: Experimental Recommendations for Ring-LWE

From the theory of Statistical Identity Testing we know that $\Theta\left(\frac{||p||_{2/3}}{\epsilon^2}\right)$ samples are required to determine if a sampled distribution differs from an ideal one by total variation distance ϵ (and we ignore samples from the distribution "tail" of weight ϵ). Therefore an appropriate selection for sampling precision is $2^{-\frac{s}{2}}$ where s is the desired security level. We conjecture that the ϵ tail has negligible effect on the entropy of secret quantities and the security of Lattice-based cryptosystems of interest, especially signature algorithms. However, this must be evaluated on case-by-base basis, and a two-step recursive application of search algorithm can be used to overcome the problem if desired.

Based on our findings, we propose the following implementation parameters that allow standard, more efficient data types to be used. Here we conservatively claim that the new parameters maintain the original security against all offline attacks if no more than 2^λ private key oracle queries are allowed for any given private key. This is a reasonable assumption as private key queries cannot be parallelised or performed without the consent of the holder of the private key. We further assume that ring polynomials are of relatively small degree n.

Security	Precision	Tailcut	Possible data type
Up to 2^{100}	$\lambda = 50$	$\lvert x\rvert < 8.1\sigma$	IEEE 754 floating point (`double`)
Up to 2^{128}	$\lambda = 64$	$\lvert x\rvert < 9.2\sigma$	64-bit fixed p. integer (`uint64_t`)
Up to 2^{192}	$\lambda = 96$	$\lvert x\rvert < 11.4\sigma$	IEEE 754 quadruple-precision
Up to 2^{256}	$\lambda = 128$	$\lvert x\rvert < 13.2\sigma$	128-bit unsigned integer type

We further recommend using constant-time samplers for all algorithms which are used in online protocols, since non-constant time samplers are easily exploitable with timing attacks.

Example. BLISS-I [7,23] with $\sigma = 215.75$ and claimed 128-bit security can equivalently use $\lambda = 64$ and a CDT table of size $n = 2048$ entries (9.5σ) in constant-time binary search. The total size of the CDT table is therefore 16kB in this case and 12 simple comparisons are required to produce each sample in constant time (if we ignore memory cache variation).

Acknowledgments

The author wishes to thank Máire O'Neill, Ilya Mironov, and Steven Galbraith for their helpful comments.

References

1. National Security Agency. NSA suite B cryptography: Cryptography today, August 2015. URL: `https://www.nsa.gov/ia/programs/suiteb_cryptography/`.

2. Shi Bai, Adeline Langlois, Tancrède Lepoint, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: using the Rényi divergence rather than the statistical distance. In *ASIACRYPT '15 (To Appear)*. Springer, 2015. URL: https://eprint.iacr.org/2015/483.
3. Joppe W. Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 553–570. IEEE Computer Society, 2015. Extended version available as IACR ePrint 2014/599. URL: https://eprint.iacr.org/2014/599, doi:10.1109/SP.2015.40.
4. Johannes Buchmann, Daniel Cabarcas, Florian Göpfert, Andreas Hülsing, and Patrick Weiden. Discrete ziggurat: A time-memory trade-off for sampling from a gaussian distribution over the integers. In Tanja Lange, Kristin Lauter, and Petr Lisoněk, editors, *SAC 2013*, volume 8282 of *LNCS*, pages 402–417. Springer, 2014. Extended version available as IACR ePrint 2014/510. URL: https://eprint.iacr.org/2013/510, doi:10.1007/978-3-662-43414-7_20.
5. Peter Campbell, Michael Groves, and Dan Shepherd. Soliloquy: A cautionary tale. ETSI 2nd Quantum-Safe Crypto Workshop in partnership with the IQC, October 2014. URL: https://docbox.etsi.org/Workshop/2014/201410_CRYPTO/S07_Systems_and_Attacks/S07_Groves_Annex.pdf.
6. Ruan de Clercq, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. Efficient software implementation of Ring-LWE encryption. In *DATE 2015*, pages 339–344. IEEE, 2015. doi:10.7873/DATE.2015.0378.
7. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013*, pages 40–56. Springer, 2013. Extended version available as IACR ePrint 2013/383. URL: https://eprint.iacr.org/2013/383, doi:10.1007/978-3-642-40041-4_3.
8. Léo Ducas and Phong Q. Nguyen. Faster gaussian lattice sampling using lazy floating-point arithmetic. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 415–432. Springer, 2012. doi:10.1007/978-3-642-34961-4_26.
9. Nagarjun C. Dwarakanath and Steven D. Galbraith. Sampling from discrete gaussians for lattice-based cryptography on a constrained device. *Applicable Algebra in Engineering, Communication and Computing*, 25(3):159–180, June 2014. doi:10.1007/s00200-014-0218-3.
10. Hassan Edrees, Brian Cheung, McCullen Sandora, David B. Nummey, and Deian Stefan. Hardware-optimized ziggurat algorithm for high-speed gaussian random number generators. In Toomas P. Plaks, editor, *ERSA 2009*, pages 254–260. CSREA Press, 2009. URL: http://sprocom.cooper.edu/sprocom2/pubs/conference/ecsns2009ersa.pdf.
11. Tim Güneysu, Tobias Oder, and Thomas Pöppelmann. Beyond ECDSA and RSA: Lattice-based digital signatures on constrained devices. In *DAC '14*, pages 1–6. ACM, 2014. doi:10.1145/2593069.2593098.
12. Charles F. F. Karney. Sampling exactly from the normal distribution, 2014. Preprint arXiv:1303.6257, Version 2. URL: http://arxiv.org/abs/1303.6257.
13. Donald E. Knuth and Andrew C. Yao. The complexity of nonuniform random number generation. In Joseph F. Traub, editor, *Algorithms and Complexity: New Directions and Recent Results*, pages 357–428, New York, 1976. Academic Press.
14. Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *CRYPTO '96*, volume 1109 of *LNCS*, pages 104–113. Springer, 1996. doi:10.1007/3-540-68697-5_9.

15. Zhe Liu, Hwajeong Seo, Sujoy Sinha Roy, Johann Großschädl, Howon Kim, and Ingrid Verbauwhede. Efficient Ring-LWE encryption on 8-bit AVR processors. In Tim Güneysu and Helena Handschuh, editors, *CHES 2015*, volume 9293 of *LNCS*, pages 663–682. Springer, 2015. `doi:10.1007/978-3-662-48324-4_33`.
16. Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, 2012. `doi:10.1007/978-3-642-29011-4_43`.
17. George Marsaglia and Wai Wan Tsang. A fast, easily implemented method for sampling from decreasing or symmetric unimodal density functions. *SIAM Journal on Scientific and Statistical Computing*, 5(2):349–359, 1984. `doi:10.1137/0905026`.
18. George Marsaglia and Wai Wan Tsang. The ziggurat method for generating random variables. *Journal of Statistical Software*, 5(8):1–7, October 2000. URL: `http://www.jstatsoft.org/v05/i08`.
19. John F. Monahan. Accuracy in random number generation. *Mathematics of Computation*, 45(172):559–568, October 1985. `doi:10.1090/S0025-5718-1985-0804945-X`.
20. Committee on National Security Systems. Use of public standards for the secure sharing of information among national security systems. CNSS Advisory Memorandum, Information Assurance 02-15, July 2015.
21. Liam Paninski. A coincidence-based test for uniformity given very sparsely-sampled discrete data. *IEEE Transactions on Information Theory*, 54:4750–4755, October 2008. `doi:10.1109/TIT.2008.928987`.
22. Chris Peikert. An efficient and parallel gaussian sampler for lattices. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 80–97. Springer, 2010. `doi:10.1007/978-3-642-14623-7_5`.
23. Thomas Pöppelmann, Léo Ducas, and Tim Güneysu. Enhanced lattice-based signatures on reconfigurable hardware. In Lejla Batina and Matthew Robshaw, editors, *CHES 2014*, volume 8731 of *LNCS*, pages 353–370. Springer, 2014. Extended version available as IACR ePrint 2014/254. URL: `https://eprint.iacr.org/2014/254`, `doi:10.1007/978-3-662-44709-3_20`.
24. Sujoy Sinha Roy, Oscar Reparaz, Frederik Vercauteren, and Ingrid Verbauwhede. Compact and side channel secure discrete gaussian sampling. IACR ePrint 2014/591, 2014. URL: `https://eprint.iacr.org/2014/591`.
25. Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. High precision discrete gaussian sampling on FPGAs. In Tanja Lange, Kristin Lauter, and Petr Lisoněk, editors, *SAC 2013*, volume 8282 of *LNCS*, pages 383–401. Springer, 2014. `doi:10.1007/978-3-662-43414-7_19`.
26. David B. Thomas, Wayne Luk, Philip H. W. Leong, and John D. Villasenor. Gaussian random number generators. *ACM Computing Surveys*, 39(4), November 2007. `doi:10.1145/1287620.1287622`.
27. Gregory Valiant and Paul Valiant. Instance-by-instance optimal identity testing. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:111, 2013. URL: `http://eccc.hpi-web.de/report/2013/111`.
28. Gregory Valiant and Paul Valiant. An automatic inequality prover and instance optimal identity testing. In *FOCS 2014*, pages 51–60. IEEE Computer Society, 2014. Full version available as `http://theory.stanford.edu/~valiant/papers/instanceOptFull.pdf`. `doi:10.1109/FOCS.2014.14`.

Document version 20151117102300.