Gaussian Sampling Precision in Lattice Cryptography

Markku-Juhani O. Saarinen

Centre for Secure Information Technologies (CSIT)
ECIT, Queen's University Belfast, 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 further propose new experimental parameters for practical Gaussian samplers for use in Lattice Cryptography.

Keywords: Post-Quantum Cryptography, Lattice Public Key Cryptography, Gaussian Sampling

1. Introduction

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 [1, 2, 3, 4, 5].

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. We conclude with new, more efficient sampler parameters in Section 5.

Email address: m.saarinen@qub.ac.uk (Markku-Juhani O. Saarinen)

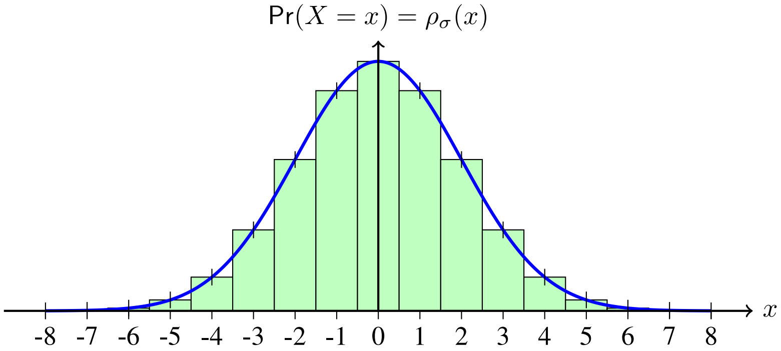

Figure 1: The Discrete Gaussian distribution D_σ (Equation 3) is defined for all $x \in \mathbb{Z}$ and satisfies $\sum_{x=-\infty}^{\infty} \rho_\sigma(x) = 1$. The green discrete bars illustrate the probability mass and the blue line is the corresponding continuous probability density function.

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:

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

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.

Required distance. It has been widely assumed that for cryptographic applications the sampling distance should be roughly the inverse of the security parameter [6]:

> 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. [7, 8, 9, 10]). 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.

Other metrics and related work. Recently, proofs of some Lattice based schemes have been reworked using Rényi distance [11, 12] 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 [10].

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 λ. 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 2 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 [7, 10].

Other Gaussian Sampling Algorithms. High precision non-uniform continuous random sampling is a classic problem [13]. Many of the algorithms of the continuous case also apply to discrete cryptographic applications. Methods such as Inversion Sampling [14], Knuth-Yao Sampling [6, 15], The Ziggurat Method [1, 16, 17, 18], Kahn-Karney Sampling [19], and "Bernoulli" sampling [2] have also been proposed for lattice cryptography. For more (non-cryptographic) methods, see [20].

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.

Tight bounds for distribution identity testing. We quote the following definitions and a recent result (Theorem 1 from [21, 22]) 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.

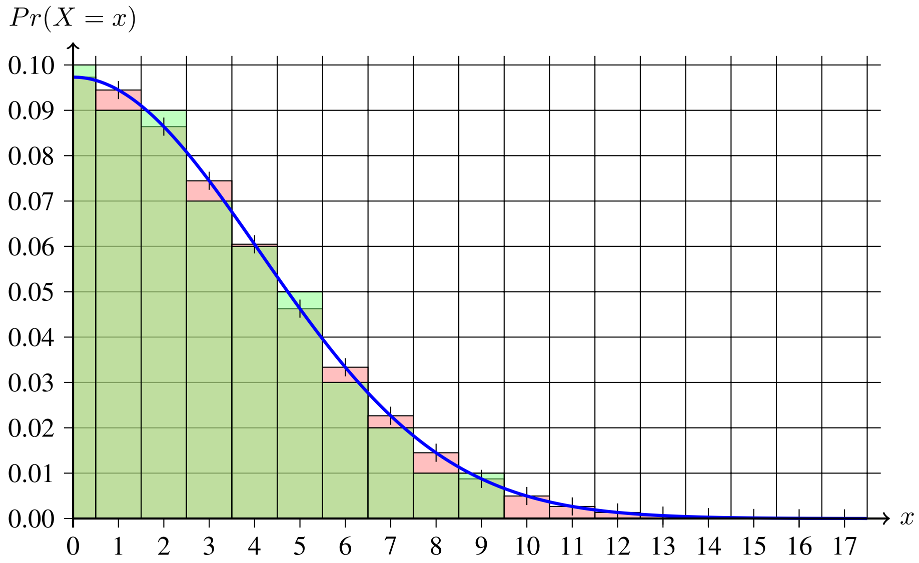

Figure 2: Sampling the discrete Gaussian distribution with $\sigma = 4.1$ and sampling precision 0.01. The red (rounded down) and green (rounded up) areas illustrate the statistical distance (total variation distance) ϵ between the ideal distribution and sampling with the precision of the grid. Only the positive half $x > 0$ of the symmetric distribution is shown. All $|x| \geq 10$ are "tail".

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 [21, 22]). *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 $\Theta\left(\frac{||p||_{2/3}}{\epsilon^2}\right)$ 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 randomised rounding. Consider a table of λ-precision approximations $T[0, 1, \ldots, 2^\lambda - 1]$:

$$T[i] = \left\lfloor 2^\lambda \frac{S_\sigma(i)}{S_\sigma(\infty)} \right\rfloor . \tag{9}$$

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 randomise it by adding +1 to each entry during initialisation 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 randomised. If the condition of Equation 8 holds and the given random integer j matches all bits of $T[i]$, a randomised 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.

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 1. *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 randomise an another j' and utilise 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 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 5. 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. Conclusions and Experimental Parameters for Lattice Cryptography

From the theory of Statistical Identity Testing we know that $\Theta\big(\frac{||p||_{2/3}}{\epsilon^2}\big)$ 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.

Based on our findings, we propose the following implementation parameters that allow standard, significantly 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**
2^{100}	$\lambda = 50$	$\lvert x\rvert < 8.1\sigma$	IEEE 754 floating point (`double`)
2^{128}	$\lambda = 64$	$\lvert x\rvert < 9.2\sigma$	64-bit fixed point (`uint64_t`)
2^{192}	$\lambda = 96$	$\lvert x\rvert < 11.4\sigma$	IEEE 754 quadruple-precision
2^{256}	$\lambda = 128$	$\lvert x\rvert < 13.2\sigma$	128-bit unsigned integer type

Example. BLISS-I [2, 10] 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

This work was funded by the European Union H2020 SAFEcrypto project (grant no. 644729). The author wishes to thank Máire O'Neill, Ilya Mironov, and Steven Galbraith for their helpful comments.

References

[1] J. Buchmann, D. Cabarcas, F. Göpfert, A. Hülsing, P. Weiden, Discrete ziggurat: A time-memory trade-off for sampling from a gaussian distribution over the integers, in: T. Lange, K. Lauter, P. Lisoněk (Eds.), SAC 2013, Vol. 8282 of LNCS, Springer, 2014, pp. 402–417, extended version available as IACR ePrint 2014/510. `doi:10.1007/978-3-662-43414-7_20`.
URL `https://eprint.iacr.org/2013/510`

[2] L. Ducas, A. Durmus, T. Lepoint, V. Lyubashevsky, Lattice signatures and bimodal gaussians, in: R. Canetti, J. A. Garay (Eds.), CRYPTO 2013, Springer, 2013, pp. 40–56, extended version available as IACR ePrint 2013/383. doi:10.1007/978-3-642-40041-4_3.
URL https://eprint.iacr.org/2013/383

[3] T. Güneysu, T. Oder, T. Pöppelmann, Beyond ECDSA and RSA: Lattice-based digital signatures on constrained devices, in: DAC '14, ACM, 2014, pp. 1–6. doi:10.1145/2593069.2593098.

[4] V. Lyubashevsky, Lattice signatures without trapdoors, in: D. Pointcheval, T. Johansson (Eds.), EUROCRYPT 2012, Vol. 7237 of LNCS, Springer, 2012, pp. 738–755. doi:10.1007/978-3-642-29011-4_43.

[5] S. S. Roy, F. Vercauteren, I. Verbauwhede, High precision discrete gaussian sampling on FPGAs, in: T. Lange, K. Lauter, P. Lisoněk (Eds.), SAC 2013, Vol. 8282 of LNCS, Springer, 2014, pp. 383–401. doi:10.1007/978-3-662-43414-7_19.

[6] N. C. Dwarakanath, S. D. Galbraith, Sampling from discrete gaussians for lattice-based cryptography on a constrained device, Applicable Algebra in Engineering, Communication and Computing 25 (3) (2014) 159–180. doi:10.1007/s00200-014-0218-3.

[7] J. W. Bos, C. Costello, M. Naehrig, D. 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, IEEE Computer Society, 2015, pp. 553–570, extended version available as IACR ePrint 2014/599. doi:10.1109/SP.2015.40.
URL https://eprint.iacr.org/2014/599

[8] R. de Clercq, S. S. Roy, F. Vercauteren, I. Verbauwhede, Efficient software implementation of Ring-LWE encryption, in: DATE 2015, IEEE, 2015, pp. 339–344. doi:10.7873/DATE.2015.0378.

[9] Z. Liu, H. Seo, S. S. Roy, J. Großschädl, H. Kim, I. Verbauwhede, Efficient Ring-LWE encryption on 8-bit AVR processors, in: T. Güneysu, H. Handschuh (Eds.), CHES 2015, Vol. 9293 of LNCS, Springer, 2015, pp. 663–682. doi:10.1007/978-3-662-48324-4_33.

[10] T. Pöppelmann, L. Ducas, T. Güneysu, Enhanced lattice-based signatures on reconfigurable hardware, in: L. Batina, M. Robshaw (Eds.), CHES 2014, Vol. 8731 of LNCS, Springer, 2014, pp. 353–370, extended version available as IACR ePrint 2014/254. doi:10.1007/978-3-662-44709-3_20.
URL https://eprint.iacr.org/2014/254

[11] S. Bai, A. Langlois, T. Lepoint, D. Stehlé, R. 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

[12] K. Takashima, A. Takayasu, Tighter security for efficient lattice cryptography via the Rényi divergence of optimized orders, in: M.-H. Au, A. Miyaji (Eds.), ProvSec 2014, Vol. 9451 of LNCS, Springer, 2015, pp. 412–431. doi:10.1007/978-3-319-26059-4_23.

[13] J. F. Monahan, Accuracy in random number generation, Mathematics of Computation 45 (172) (1985) 559–568. doi:10.1090/S0025-5718-1985-0804945-X.

[14] C. Peikert, An efficient and parallel gaussian sampler for lattices, in: T. Rabin (Ed.), CRYPTO 2010, Vol. 6223 of LNCS, Springer, 2010, pp. 80–97. doi:10.1007/978-3-642-14623-7_5.

[15] D. E. Knuth, A. C. Yao, The complexity of nonuniform random number generation, in: J. F. Traub (Ed.), Algorithms and Complexity: New Directions and Recent Results, Academic Press, New York, 1976, pp. 357–428.

[16] H. Edrees, B. Cheung, M. Sandora, D. B. Nummey, D. Stefan, Hardware-optimized ziggurat algorithm for high-speed gaussian random number generators, in: T. P. Plaks (Ed.), ERSA 2009, CSREA Press, 2009, pp. 254–260.
URL http://sprocom.cooper.edu/sprocom2/pubs/conference/ecsns2009ersa.pdf

[17] G. Marsaglia, W. W. Tsang, A fast, easily implemented method for sampling from decreasing or symmetric unimodal density functions, SIAM Journal on Scientific and Statistical Computing 5 (2) (1984) 349–359. doi:10.1137/0905026.

[18] G. Marsaglia, W. W. Tsang, The ziggurat method for generating random variables, Journal of Statistical Software 5 (8) (2000) 1–7.
URL http://www.jstatsoft.org/v05/i08

[19] C. F. F. Karney, Sampling exactly from the normal distribution, preprint arXiv:1303.6257, Version 2 (2014).
URL http://arxiv.org/abs/1303.6257

[20] D. B. Thomas, W. Luk, P. H. W. Leong, J. D. Villasenor, Gaussian random number generators, ACM Computing Surveys 39 (4). doi:10.1145/1287620.1287622.

[21] G. Valiant, P. Valiant, Instance-by-instance optimal identity testing, Electronic Colloquium on Computational Complexity (ECCC) 20 (2013) 111.
URL http://eccc.hpi-web.de/report/2013/111

[22] G. Valiant, P. Valiant, An automatic inequality prover and instance optimal identity testing, in: FOCS 2014, IEEE Computer Society, 2014, pp. 51–60, full version available as http://theory.stanford.edu/~valiant/papers/instanceOptFull.pdf. doi:10.1109/FOCS.2014.14.

Document version 20151207094300