On estimating the lattice security of NTRU

Nick Howgrave-Graham, Jeff Hoffstein, Jill Pipher, William Whyte

NTRU Cryptosystems

Abstract. This report explicitly refutes the analysis behind a recent claim that NTRUEncrypt has a bit security of at most 74 bits. We also sum up some existing literature on NTRU and lattices, in order to help explain what should and what should not be classed as an improved attack against the hard problem underlying NTRUEncrypt. We also show a connection between Schnorr's RSR technique and exhaustively searching the NTRU lattice.

1 Introduction

There have been a number of papers and technical notes [11, 3, 6, 12, 21, 10] explaining the NTRU cryptosystem, and analyzing possible avenues of attack. However there is no all-encompassing paper which allows easy access to this material. It is hoped that this report will help make the material more accessible[1].

As a by-product of this it will hopefully be clearer whether a new lattice idea has the power to impact the security of NTRU, and to what degree. A motivation for this document is the recent posting of [2], in which it is speculated that NTRU has 74 bits of security, rather than the 80 claimed in [12]. In section 4 we explain several ways in which we consider the analysis behind [2] to be flawed.

This report is concerned with the underlying hard problem behind NTRU, and not with the padding scheme or the proof of security, which have been addressed in [22, 7, 9].

In the remainder of this report we first discuss how the practical security of cryptosystems is arrived at (Section 2), we then discuss some recent advances in lattice reduction (Section 3), and in Section 4 we address our issues with the analysis in [2]. In Section 5 we give a preliminary analysis about the use of Random Sampling Reduction [18] (the technique underlying the techniques of [2]) when applied to NTRU lattices.

2 Estimating the practical security of cryptosystems

The supposed hardness of the problem underlying any public key cryptosystem can only be judged by the most effective known attack against it. Moreover, what this attack is may be highly dependent on the parameter generation algorithm of the cryptosystem. For example RSA is weak if (p, q, e) are chosen such that

[1] The reader may also want to refer to the peer review section at `www.ntru.com`

$d = e^{-1} \bmod (p-1)(q-1)$ is particularly small, or if p, q are too close together (see [23, 5]).

Theoretically we normally judge the security of a cryptosystem by the asymptotic behaviour of the most effective known attack as the security parameter tends to infinity.

For example, for most sensible choices of parameter generation algorithm, the most effective known attack against RSA is to factor $N = pq$ using the number field sieve [13, 4]. In an asymptotic sense this makes RSA susceptible to a sub-exponential attack.

For elliptic curve cryptography (ECC) the most effective known attack against a reasonable parameter choice[2] is Pollard's rho method, applied to the elliptic curve group. If this is indeed the best attack against ECC then, since Pollard's rho method essentially square roots the time of an exhaustive search, the time to break ECC remains fully exponential.

In NTRU, with the parameter generation algorithm given in [10], the most effective known attack is a meet-in-the-middle method due to Odlyzko [6]. As with ECC, if this is indeed the best attack against NTRU, then the time to break NTRU remains fully exponential.

In practical cryptography there is a slight deviation from theoretical cryptography in that the asymptotics are not quite as important as the time it takes to break a particular parameter set. For example one may wish to use a parameter set in which the most effective known attack takes approximately 2^{80} operations, or 2^{128} operations.

Of course the most effective known attack, for any of the cryptosystems mentioned above, is subject to change if an inspirational new idea is found. And of course such a new attack may also change the complexity class the cryptosystem is considered to live in.

2.1 Attacking NTRU via lattice reduction

Although the most effective known attack against the standard NTRU parameters [10] is the meet-in-the-middle attack [6], there are other avenues of attack, e.g. it was shown in [11, 3] how a suitably good lattice reduction algorithm could recover the private key.

In the case of the standard NTRU parameters [10], it is estimated that attacks based on lattice reduction [12] are not that far behind the strength implied by the meet-in-the-middle attack [6], so one could imagine a new idea in lattice reduction becoming the most effective known attack against NTRU.

A problem with estimating the running times of lattice reduction algorithms, is that they often behave far better than one can prove. In order to combat this, and try to get a feel for how quickly they work in practice, NTRU have run a series of tests, and extrapolated the data in a conservative manner.

---

[2] There are known bad parameter choices for ECC too, e.g. using anomalous elliptic curves, or using finite fields which allow Weil descent-type methods.

The exact details of the data set and extrapolation technique can be found in [12], but the general principal is summed up below. The phenomenon observed was that taking logarithms of the running times still gave a graph with a slight upward concavity. In order to get a conservative straight line fit, the slope of the graph was estimated at the high end, and security was calculated by projecting this tangent line.

Admittedly this is not a particularly rigorous way to estimate security, but accurate mathematical models for lattice reduction times are not presently in existence, and this does seem to be conservative. To add credence to this claim, this analysis has not been contradicted by the cryptographic community in the years since its inception.

The particular lattice reduction strategy that was used to get the data points, was one of continually increasing the block size of Schnorr's BKZ technique [16, 17] until the private key was recovered. This was done using the implementation of BKZ in the NTL Library [20].

3 Recent advances in lattice reduction

There have been several new ideas in lattice reduction algorithms recently [1, 18, 15, 19]. The result in [1] was particularly well received since it introduces a randomized sieving algorithm which solves SVP in time $2^{O(n)}$ in an n-dimensional rational lattice; a large improvement, in asymptotic terms, over the previously best known result of $2^{O(n \log n)}$.

Another new and interesting idea is suggested in [18], which mixes BKZ reduction, and an exhaustive search. By a suitable choice of parameters the paper claims to 4'th root the provable reduction times necessary to solve SVP within a given approximation factor, where this is compared to Koy's primal-dual[3] strategy [14].

More recently, faster versions of LLL reduction [8] have been proposed in [15, 19].

As acknowledged in Section 2, any new idea has the potential to become the best known algorithm against a given cryptosystem. To show that a new lattice approach is indeed the best known attack against NTRU, as specified in [10], the idea would have to beat the meet-in-the-middle approach [6] by some margin. To show this, the would-be authors might be able to rigorously prove the running time (perhaps under some realistic heuristics), or they might have to resort to an extrapolation technique, similar to that which NTRU uses to derive security.

Either way could be convincing, but it is thought that to make a convincing argument as to the (in)security of NTRU, one (or both) of these approaches should be followed.

One point to note, when using the practical extrapolation technique, is that if the data set has an upward concavity, as witnessed by the NTRU analysis, then it is prudent to test examples which are as large as is reasonably possible to

[3] This result is in fact unpublished, though the citation holds some slides describing the technique.

obtain. This is because by testing only small examples, and using the end data points to extrapolate a line, one would be in danger of underestimating the true security.

4 On practical lattice basis sampling reduction

Recently in [2] the authors applied ideas similar to that of Schnorr in [18] to NTRU. However they do not use either of the techniques recommended in Section 3 to estimate the strength of their attack. Instead they seem to use the following argument:

1. NTRU have estimated[4] the running time for a particular parameter set, dependent on N, to be $10^{0.1095N-12.6401}$.
2. NTRU used a strategy of increasing the BKZ blocksize until a success was found.
3. Schnorr's RSR technique [18] predicts asymptotics that 4'th root the running times, compared to this approach of increasing the BKZ blocksize.
4. RSR does not seem to work well with NTRU lattices, but it is claimed the new modification, SR, does work reasonably well.
5. It is not clear what the asymptotic improvement of SR is though, so a guess of a factor of 3/4 in the exponent is made, as opposed to the 1/4 predicted by RSR.
6. Thus SR probably has an asymptotic running time of

   $$10^{(3/4)(0.1095N-12.6401)+c} = 10^{0.082N+c'}$$

   for some constants c, c'.

This is the reasoning that gives the figure of 74 bit security, when $N = 251$, as opposed to the 80-bit security predicted in [12].

There are several fallacies in this argument. The first (at step 3) is to assert that Schnorr's algorithm asymptotically 4'th roots the BKZ reduction times, for a given approximation factor. Schnorr does not make this claim in [18], and the only comparison involves a table between Schnorr's $2k$-reduction and Koy's primal-dual reduction. Indeed since the asymptotics that Schnorr predicts are super-exponential, yet the asymptotics NTRU states are purely exponential, there is clearly no advantage in an asymptotic sense to using RSR[5]. We show the impact of naively applying Schnorr's RSR technique to NTRU in Section 5.1.

If RSR were to 4'th root the practical times achieved by BKZ then the above argument might be plausible, but there is no evidence of this.

The second problem with [2] is that there is no justification for using the factor of 3/4 at step 5, other than it lies between 1/4 and 1. If indeed such a

[4] In fact, these figures are based on [11]; more up-to-date figures can be found in [12, 10].

[5] Both [1] and a naive exhaustive search are merely exponential, and therefore beat RSR in an asymptotic sense.

constant exists for average case running time, it should either be justified in a mathematical sense, or it should be worked out experimentally.

A third problem with [2] is in the whole idea of applying an asymptotic result to a formula gotten by testing data. If one had strong confidence that the formula of $10^{0.1095N-12.6401}$ was exact, i.e. with no conservatism, then it might be realistic to assume the asymptotics had kicked in by $N = 251$, and make the above argument. However the data gotten by NTRU has a definite upward concavity. If this upward concavity continues[6] it could counteract any constant like $3/4$ in the exponent, or even the $1/4$ predicted by Schnorr in [18].

The conclusion is that in the absence of a more rigorous mathematical model, there is no substitute for running the experiments for the technique suggested in [2], and extrapolating the data. As stated numerous times before, we cannot apriori say that such ideas will not yield an improved attack against NTRU; we can only say that the argument given in [2] is not rigorous.

5 RSR and the NTRU lattice

5.1 A naive use of RSR

It is instructive to see what happens if one simply takes the asymptotic bounds for sampling reduction given in [18] and applies them directly to obtain estimates of the running times for sampling reduction of the parameter sets given in [10].

The performance of a lattice reduction algorithm is expressed in terms of its running time to achieve a given approximation factor, or α. where

$$\alpha = \frac{\text{Length of shortest vector found}}{\text{Length of shortest vector in lattice}} .$$

The α obtained in [18] is $\alpha_{\mathrm{RSR}}(n, k) = (k/6)^{n/(2k)}$, where n is the dimension of the lattice and k is a freely chosen parameter. The corresponding (fourth rooted) running time is then given as $n^3(k/6)^{k/4}$, which is superexponential in k as noted above. The required α to solve an NTRU lattice would be $\alpha = q/\sqrt{2||f|| \cdot ||g||}$. Here we are making the assumption that an attacker has broken a parameter set simply by finding a vector shorter than a q vector, in line with the assumptions of [2].

The parameters q, df, N are those provided in [10] for a given bit security level s. Also $dg = N/2$ and the dimension n is $n = 2N$. The column k gives the least k necessary to achieve the appropriate α and the column T gives the corresponding log running time as read off from the formula $T = \log_2(n^3(k/6)^{k/4}$. This log running time T needs to be shifted by a constant factor translating running time to bits, but even the most conservative choice for the constant factor puts it far in excess of the meet in the middle security level s.

[6] NTRU acknowledges that if the upward concavity continues, then it is merely an indication that this particular BKZ-based attack is less effective asymptotically.

s	N	df	q	α	k	T
80	251	48	197	19.8	339	520
112	347	66	269	23.1	485	797
128	397	74	307	24.7	562	949
160	491	91	367	26.6	715	1263
192	587	108	439	29.1	865	1581
256	787	140	587	33.8	1180	2280

Table 1. Estimated $\log_2$ running time based on asymptotic formula.

5.2 A connection with exhaustively searching the NTRU lattice

We briefly explain that although the analysis in Section 5.1 is what one gets when naively applying the asymptotics predicted by Schnorr to the NTRU lattice, one can achieve far better asymptotics with RSR when one does no BKZ reduction whatsoever.

Consider the NTRU lattice as

$$\begin{pmatrix} q & 0 \\ h & 1 \end{pmatrix}$$

where $h = g/f \bmod q$, and with small vectors equal to all the N rotations of a binary vector (g, f). If one misses out on the BKZ step of RSR, and then applies the exhaustive search part of RSR to the last N vectors, then essentially one is exhaustively searching through the space of all possible 2^N binary vectors f. When any rotation of the f vector is tried, the technique will find the corresponding g (which will be surprisingly small), and hence recover the private key.

Thus it can be argued that RSR applied to NTRU lattices has complexity at most 2^N. It is easy to see that Schnorr's technique can in fact be speeded up for the NTRU lattice, if f only has d_f ones. However even with this observation, the approach is still slower[7] than the meet-in-the-middle attack on the NTRU lattice.

Intuitively it is hard to believe that the BKZ part of RSR does not help at all. Indeed NTRU is performing an internal analysis to assess the impact of such approaches. We do not include the results of such an analysis here.

6 Conclusions

We have explained how the NTRU estimates for lattice security were gotten, and suggest that, barring a nice mathematical model, this is a reasonable way to do it. Having said this, we still hope that the mathematical modelling of lattice reduction techniques will improve.

[7] It takes roughly the square of the time.

We have discussed the potential impact of RSR to NTRU lattices, and have shown that one must take considerable care when stating the impact of asymptotic results. We are also analyzing the impact of RSR internally.

We do not agree with the principle of mixing worst case asymptotic results and experimental data, as done in [2].

References

1. M. Ajtai, R. Kumar, D. Sivakumar *Sampling short lattice vectors and the closest lattice vector problem*, Proc. 17th IEEE Conference on Computational Complexity (CCC), pp. 53–57
2. J. Buchmann, C. Ludwig *Cryptology ePrint Archive Report 2005/072: Practical Lattice Basis Sampling Reduction*
3. D. Coppersmith, A. Shamir, *Lattice attacks on NTRU*, Proceedings of EUROCRYPT 97.
4. D. Coppersmith, *Modifications to the Number Field Sieve*, J. Cryptology 6, pp. 169–180, 1993.
5. D. Coppersmith, *Finding a small root of a bivariate integer equation; factoring with high bits known*, Proceedings of EUROCRYPT '96.
6. N. Howgrave-Graham, J. H. Silverman, W. Whyte, *NTRU Cryptosystems Technical Report #004, Version 2: A Meet-In-The-Middle Attack on an NTRU Private Key*, `www.ntru.com`
7. N. Howgrave-Graham, P. Nguyen, D. Pointcheval, J. Proos, J. H. Silverman, A. Singer, W. Whyte *The Impact of Decryption Failures on the Security of NTRU Encryption*, Proceedings of CRYPTO 2003.
8. A. K. Lenstra, H. W. Lenstra, L. Lovasz *Factoring Polynomials with Rational Coefficients*, Mathematische Annalen, vol. 261, n. 4, 1982, pp. 515–534
9. N. Howgrave-Graham, J. H. Silverman, A. Singer, W. Whyte *NAEP: Provable Security in the Presence of Decryption Failures*, `www.ntru.com`
10. N. Howgrave-Graham, J. H. Silverman, A. Singer, W. Whyte *Choosing Parameter Sets for NTRUEncrypt with NAEP and SVES-3*, Proceedings of CT-RSA 2005.
11. J. Hoffstein, J. Pipher, J. H. Silverman, *NTRU: A Ring-Based Public Key Cryptosystem*, Proceedings of ANTS III, pp. 267-288, LNCS 1423, Springer-Verlag, June 1998.
12. J. Hoffstein, J. H. Silverman, W. Whyte, *NTRU Cryptosystems Technical Report #012, Version 2: Estimated Breaking Times for NTRU Lattices*, `www.ntru.com`
13. A. K. Lenstra, H. W. Lenstra Jr., M. S. Manasse, J. M. Pollard, *The number field sieve*, Proc. 22nd ACM Symp. Theory of Computing (1990), pp. 564–572.
14. H. Koy, *Primale/duale Segment-Reduktion von Gitterbasen*, Available from `http://www.mi.informatik.uni-frankfurt.de/research/papers.html`, 2004.
15. P. .Q. Nguyen, D. Stehlé *Floating-Point LLL Revisited*, Proc. of EUROCRYPT '05. Available from `http://www.di.ens.fr/ pnguyen/pub.html`
16. C. P. Schnorr *A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms*, Theoretical Computer Science, Vol. 53, pp. 201–224, 1987.
17. C. P. Schnorr, M. Euchner *Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems*, Mathematical Programming, Vol. 66, pp. 181–191, 1994.
18. C. P. Schnorr *Lattice Reduction by Random Sampling and Birthday Methods*, STACS 2003, LNCS 2607, Springer-Verlag, pp. 145–156

19. C. P. Schnorr *Fast LLL-Type Lattice Reduction*, `http://www.mi.informatik.uni-frankfurt.de/research/papers.html`
20. V. Shoup, *NTL: A library for doing Number Theory*, `http://www.shoup.net`
21. J. H. Silverman *NTRU Cryptosystems Technical Report #013, Version 2: Dimension-Reduced Lattices, Zero-Forced Lattices, and the NTRU Public Key Cryptosystem*, `www.ntru.com`
22. J. H. Silverman, W. Whyte *NTRU Cryptosystems Technical Report #018, Estimating Decryption Failure Probabilities for NTRUEncrypt*, `www.ntru.com`
23. M. J. Wiener, *Cryptanalysis of Short RSA Secret Exponents*, IEEE Transactions on Information Theory, vol. 36, no. 3, 1990, pp. 553-558.