NTRU Prime

(and Kyber, Saber, S-unit Attacks, etc.)

Yi-Kai Liu

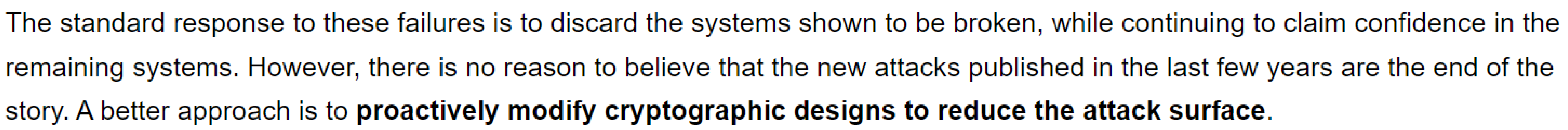
NIST

November 2021

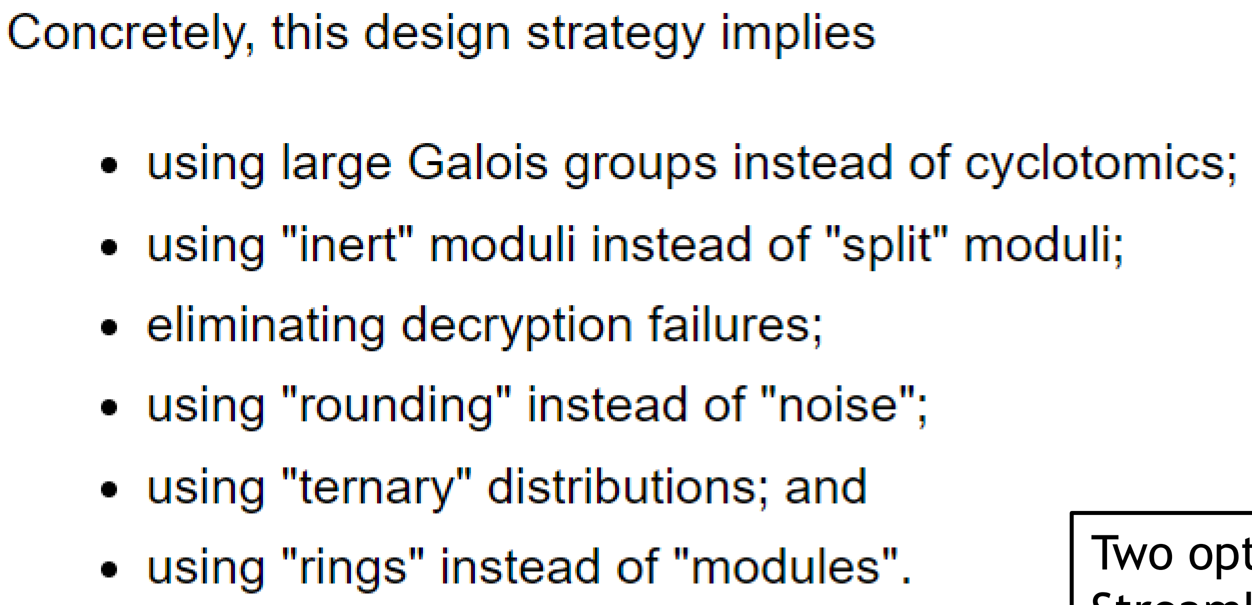
Not for distribution

NTRU Prime – “Design Philosophy”

- https://ntruprime.cr.yp.to/ (version 2021.01.14)



The standard response to these failures is to discard the systems shown to be broken, while continuing to claim confidence in the remaining systems. However, there is no reason to believe that the new attacks published in the last few years are the end of the story. A better approach is to **proactively modify cryptographic designs to reduce the attack surface**.



Concretely, this design strategy implies

- using large Galois groups instead of cyclotomics;
- using "inert" moduli instead of "split" moduli;
- eliminating decryption failures;
- using "rounding" instead of "noise";
- using "ternary" distributions; and
- using "rings" instead of "modules".

Comment:

Some of these design choices can both help and hurt security…

Two options:
Streamlined NTRU Prime
NTRU LPRime

Meta-comment:

Different ways to define and analyze the “attack surface”

Security reductions, security proofs

Taxonomies of possible attacks and counter-measures

NTRU Prime - Security

- Big picture: Choice of the ring: $Z_q[x]$ / $(x^p - x - 1)$

- Contrast with cyclotomic rings, which have more known attacks (but also easier to analyze)
  - [CDW17] Cramer, Ducas and Wesolowski, "Short Stickelberger Class Relations and application to Ideal-SVP," https://eprint.iacr.org/2016/885
  - [DPW19] Ducas, Plancon and Wesolowski, "On the Shortness of Vectors to be found by the Ideal-SVP Quantum Algorithm," https://eprint.iacr.org/2019/234
  - Claim: Quantum algorithm has approx. ratio $\exp(\tilde{O}(\sqrt{n}))$, which is good asymptotically, but doesn't beat LLL/BKZ algorithms on instances that we care about
  - S-unit attacks? We'll discuss this later…

NTRU Prime – “Risks”

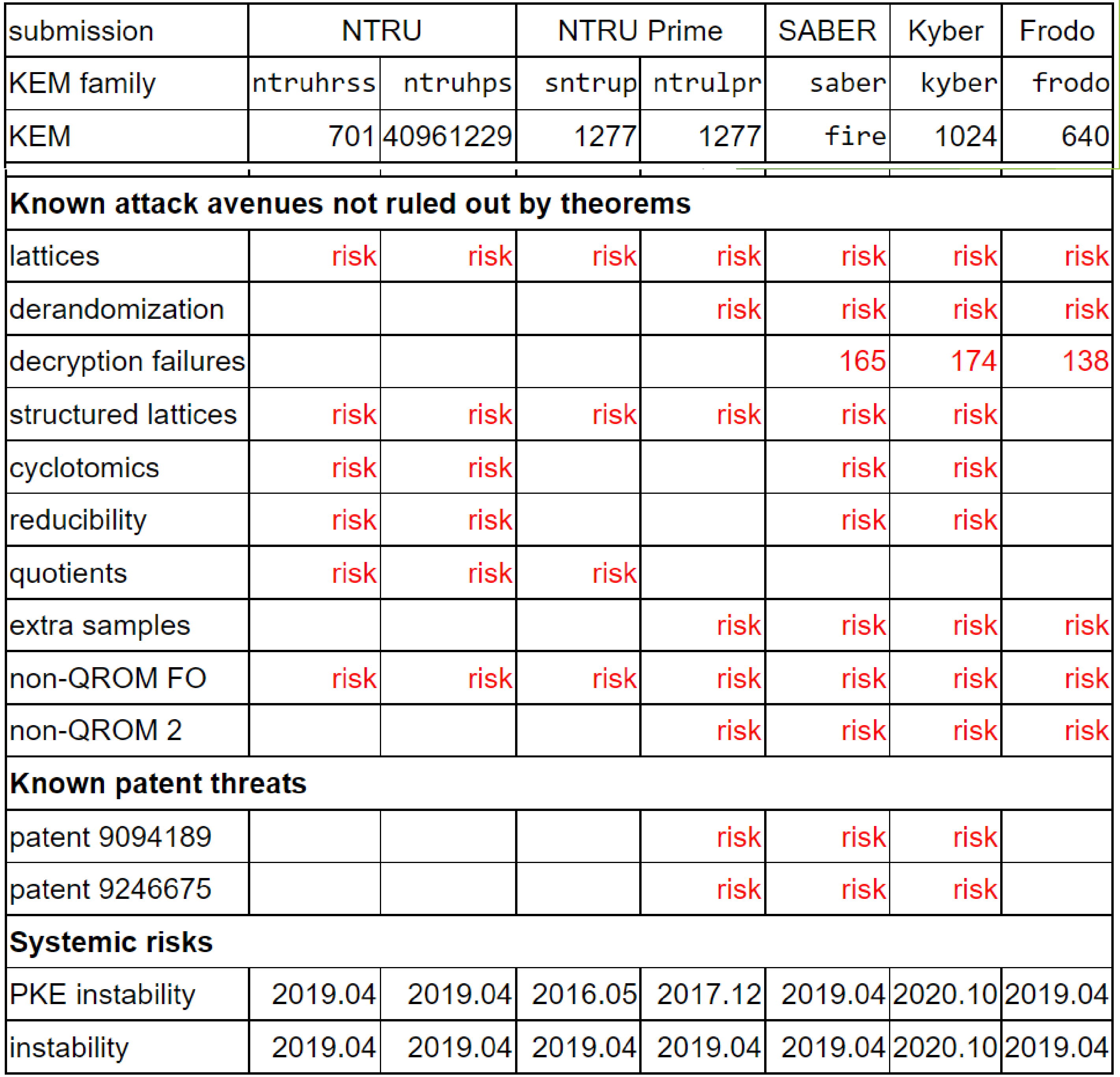
- https://ntruprime.cr.yp.to/warnings.html (version 2021.10.31)

Comments:

Some of these risks are larger than others

Some of these risks can be mitigated easily, some cannot

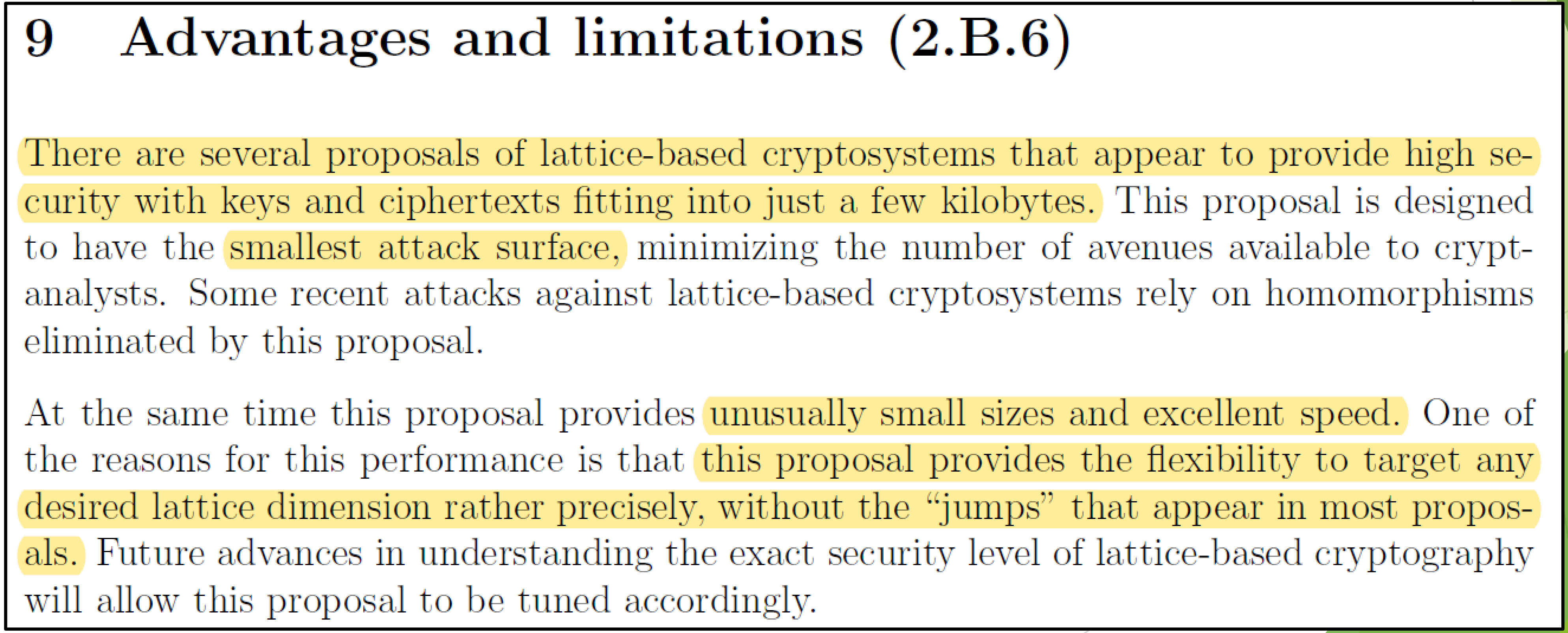
This table does not include all possible risks to PQC



submission	NTRU		NTRU Prime		SABER	Kyber	Frodo
KEM family	ntruhrss	ntruhps	sntrup	ntrulpr	saber	kyber	frodo
KEM	701	40961229	1277	1277	fire	1024	640
Known attack avenues not ruled out by theorems							
lattices	risk	risk	risk	risk	risk	risk	risk
derandomization				risk	risk	risk	risk
decryption failures					165	174	138
structured lattices	risk	risk	risk	risk	risk	risk	
cyclotomics	risk	risk			risk	risk	
reducibility	risk	risk			risk	risk	
quotients	risk	risk	risk				
extra samples				risk	risk	risk	risk
non-QROM FO	risk	risk	risk	risk	risk	risk	risk
non-QROM 2				risk	risk	risk	risk
Known patent threats							
patent 9094189				risk	risk	risk	
patent 9246675				risk	risk	risk	
Systemic risks							
PKE instability	2019.04	2019.04	2016.05	2017.12	2019.04	2020.10	2019.04
instability	2019.04	2019.04	2019.04	2019.04	2019.04	2020.10	2019.04

“The Case for NTRU Prime”

- https://ntruprime.cr.yp.to/nist/ntruprime-20201007.pdf
- Originally claimed advantages in both security and performance
- But, had to add larger/slower parameter sets in rounds 2 and 3
- Recent arguments in favor of NTRU Prime put more emphasis on security



9 Advantages and limitations (2.B.6)

There are several proposals of lattice-based cryptosystems that appear to provide high security with keys and ciphertexts fitting into just a few kilobytes. This proposal is designed to have the smallest attack surface, minimizing the number of avenues available to cryptanalysts. Some recent attacks against lattice-based cryptosystems rely on homomorphisms eliminated by this proposal.

At the same time this proposal provides unusually small sizes and excellent speed. One of the reasons for this performance is that this proposal provides the flexibility to target any desired lattice dimension rather precisely, without the “jumps” that appear in most proposals. Future advances in understanding the exact security level of lattice-based cryptography will allow this proposal to be tuned accordingly.

NTRU Prime – Parameter Sets

- Estimates of security strength have changed quite a bit, see table
  - Why? Mistakes, confusion about NIST security categories, recent research progress
- We should probably do a more detailed comparison of security/performance with NTRU, Kyber, Saber…

Parameter set	Round 1	Round 2	Round 3
sntrup653		Category 2	Category 1
sntrup761 (sntrup4591761)	Category 5	Category 3	Category 2
sntrup857		Category 4	Category 3 or 2*
sntrup953			Category 4 or 3*
sntrup1013			Category 4
sntrup1277			Category 5

https://ntruprime.cr.yp.to/nist.html (version 2020.10.31)

* See "Bulletproofing strategies #1 and #2," reverse-engineering the NIST security categories, https://ntruprime.cr.yp.to/nist/ntruprime-20201007.pdf

“The Case for NTRU Prime”

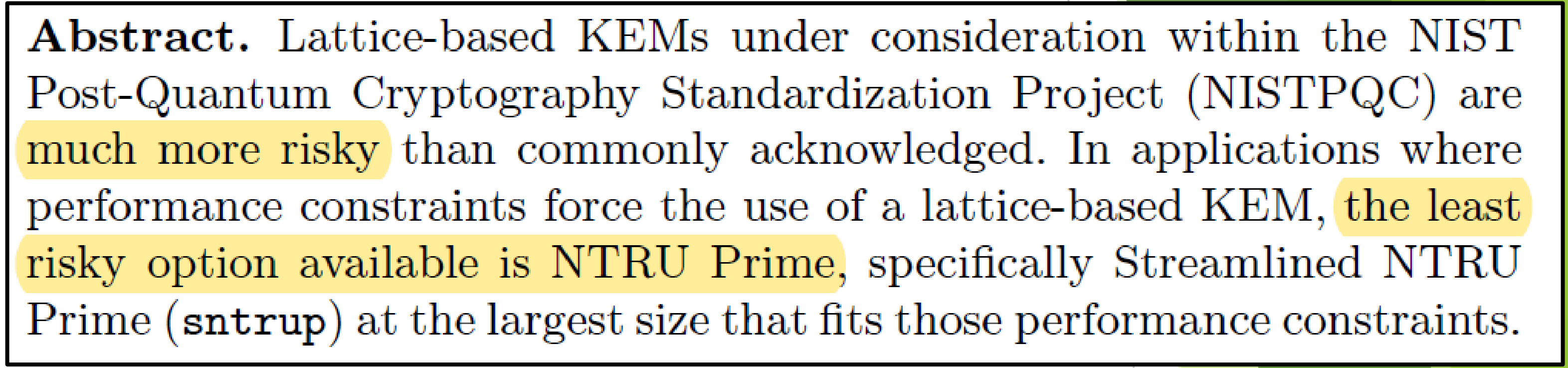
- https://ntruprime.cr.yp.to/latticerisks-20211031.pdf (version 2021.10.31)

Comments:

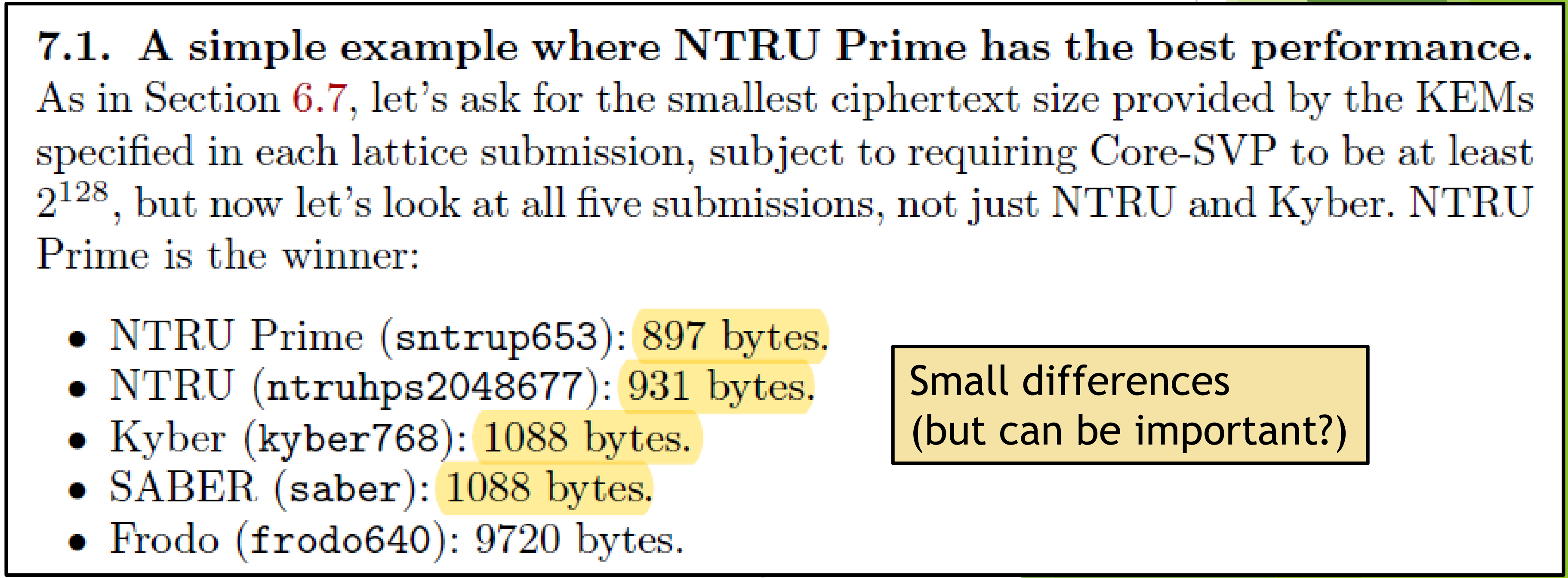
The case for NTRU Prime is based on security, not performance

Most of the evidence is negative (i.e., against Kyber and Saber, rather than for NTRU Prime)

If we standardize Kyber or Saber, we should address these criticisms in our report



Abstract. Lattice-based KEMs under consideration within the NIST Post-Quantum Cryptography Standardization Project (NISTPQC) are much more risky than commonly acknowledged. In applications where performance constraints force the use of a lattice-based KEM, the least risky option available is NTRU Prime, specifically Streamlined NTRU Prime (`sntrup`) at the largest size that fits those performance constraints.



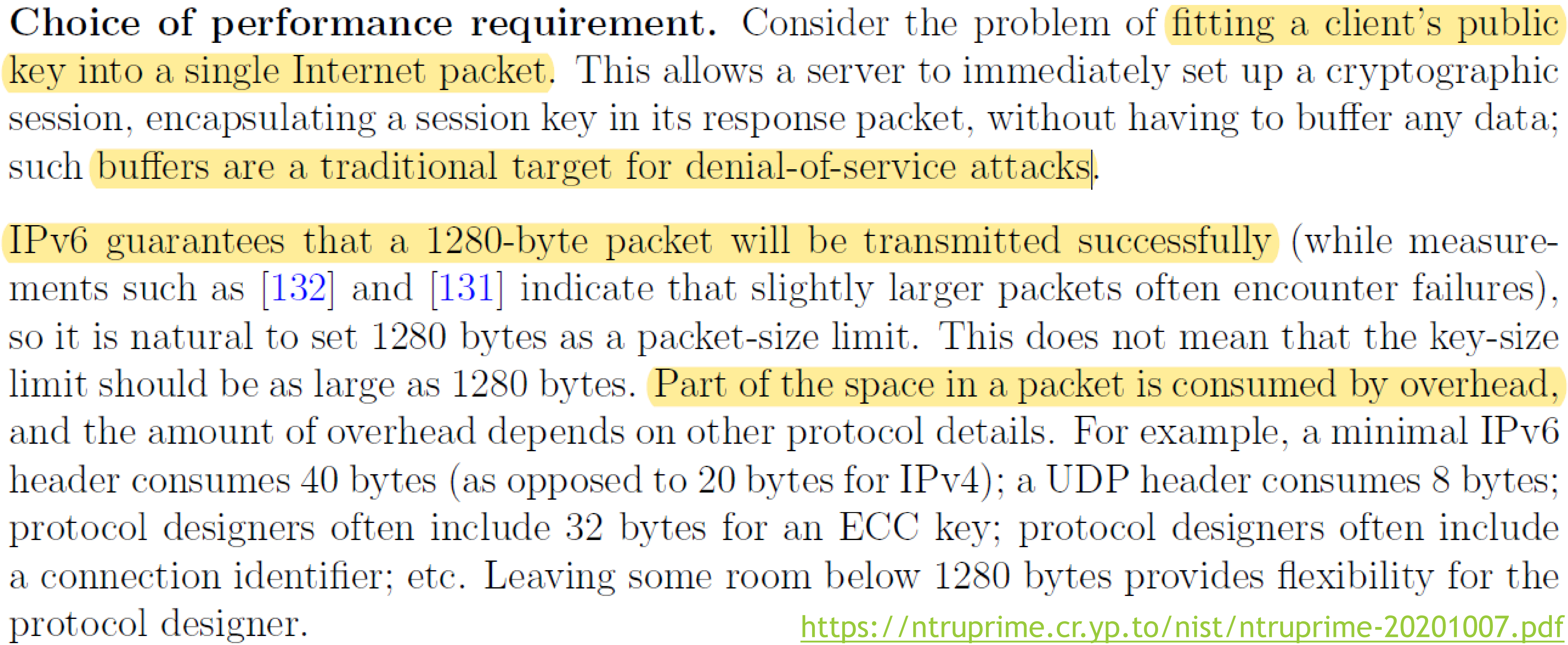
7.1. A simple example where NTRU Prime has the best performance. As in Section 6.7, let's ask for the smallest ciphertext size provided by the KEMs specified in each lattice submission, subject to requiring Core-SVP to be at least 2^{128}, but now let's look at all five submissions, not just NTRU and Kyber. NTRU Prime is the winner:

- NTRU Prime (`sntrup653`): 897 bytes.
- NTRU (`ntruhps2048677`): 931 bytes.
- Kyber (`kyber768`): 1088 bytes.
- SABER (`saber`): 1088 bytes.
- Frodo (`frodo640`): 9720 bytes.

Small differences
(but can be important?)

“Fitting a public key into a single packet”

Question: How important is this?

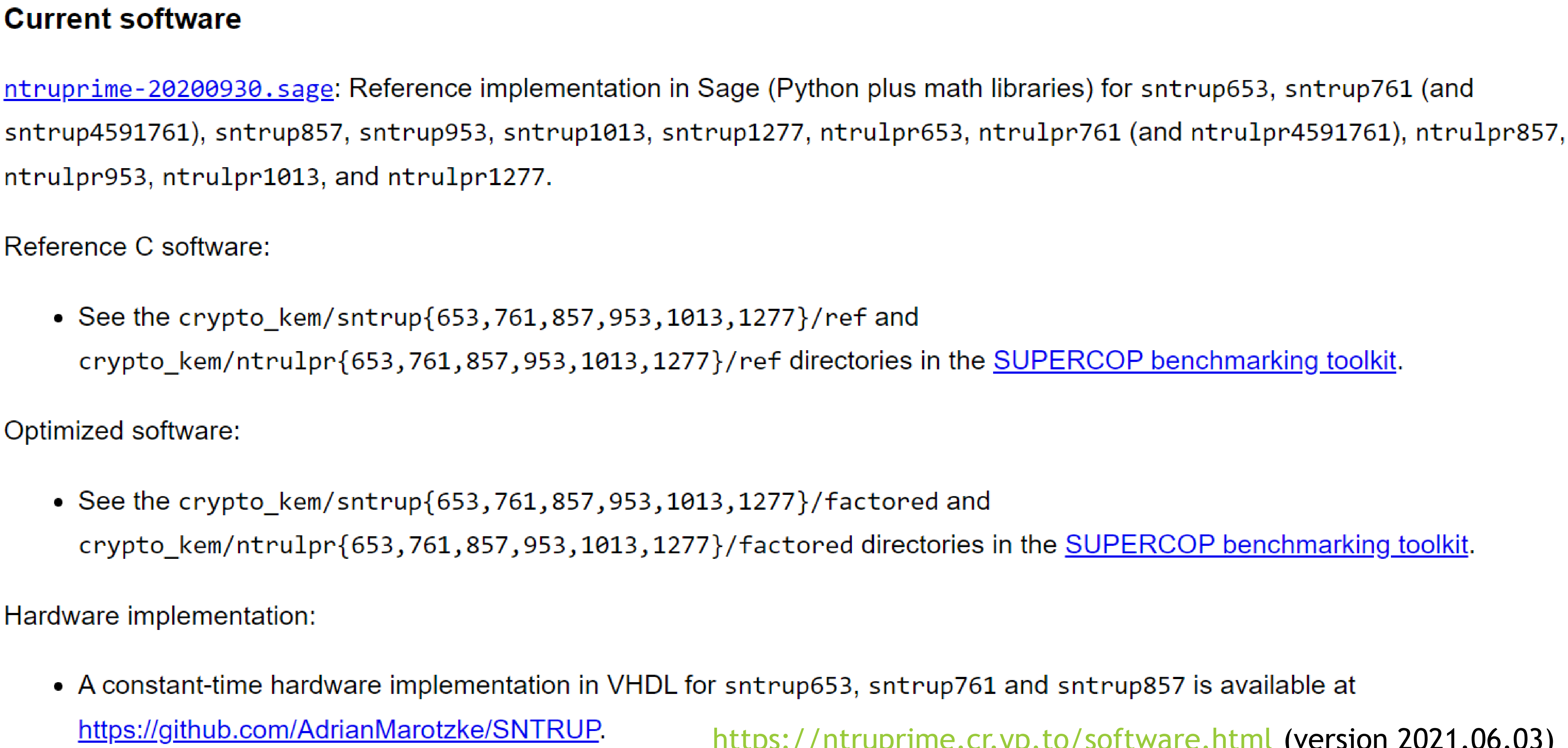


Choice of performance requirement. Consider the problem of fitting a client’s public key into a single Internet packet. This allows a server to immediately set up a cryptographic session, encapsulating a session key in its response packet, without having to buffer any data; such buffers are a traditional target for denial-of-service attacks.

IPv6 guarantees that a 1280-byte packet will be transmitted successfully (while measurements such as [132] and [131] indicate that slightly larger packets often encounter failures), so it is natural to set 1280 bytes as a packet-size limit. This does not mean that the key-size limit should be as large as 1280 bytes. Part of the space in a packet is consumed by overhead, and the amount of overhead depends on other protocol details. For example, a minimal IPv6 header consumes 40 bytes (as opposed to 20 bytes for IPv4); a UDP header consumes 8 bytes; protocol designers often include 32 bytes for an ECC key; protocol designers often include a connection identifier; etc. Leaving some room below 1280 bytes provides flexibility for the protocol designer.

https://ntruprime.cr.yp.to/nist/ntruprime-20201007.pdf

NTRU Prime - Implementations



Current software

ntruprime-20200930.sage: Reference implementation in Sage (Python plus math libraries) for sntrup653, sntrup761 (and sntrup4591761), sntrup857, sntrup953, sntrup1013, sntrup1277, ntrulpr653, ntrulpr761 (and ntrulpr4591761), ntrulpr857, ntrulpr953, ntrulpr1013, and ntrulpr1277.

Reference C software:

- See the crypto_kem/sntrup{653,761,857,953,1013,1277}/ref and crypto_kem/ntrulpr{653,761,857,953,1013,1277}/ref directories in the SUPERCOP benchmarking toolkit.

Optimized software:

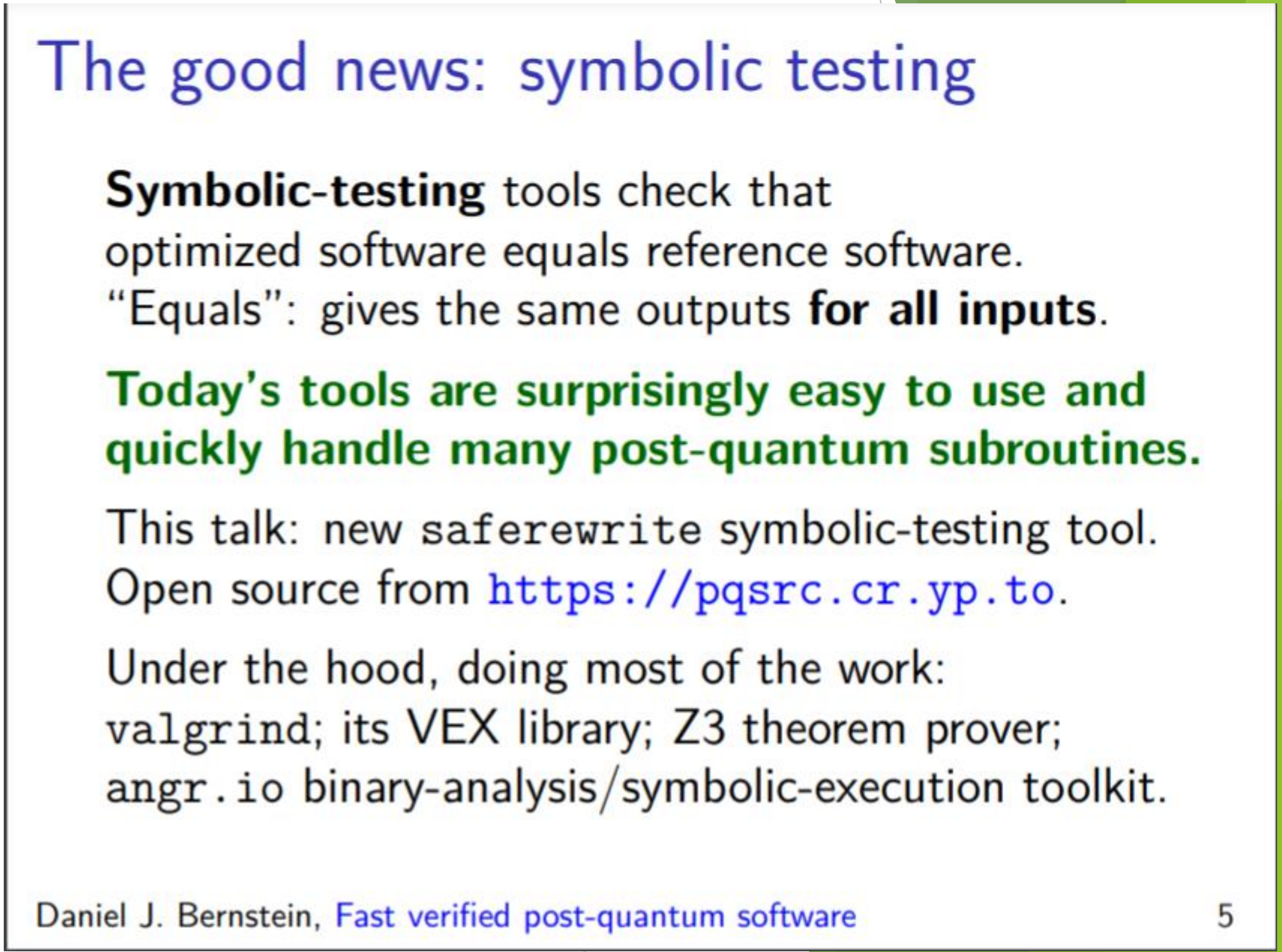
- See the crypto_kem/sntrup{653,761,857,953,1013,1277}/factored and crypto_kem/ntrulpr{653,761,857,953,1013,1277}/factored directories in the SUPERCOP benchmarking toolkit.

Hardware implementation:

- A constant-time hardware implementation in VHDL for sntrup653, sntrup761 and sntrup857 is available at https://github.com/AdrianMarotzke/SNTRUP.

https://ntruprime.cr.yp.to/software.html (version 2021.06.03) https://cr.yp.to/talks/2021.09.03/slides-djb-20210903-saferewrite-4x3.pdf

Software verification



The good news: symbolic testing

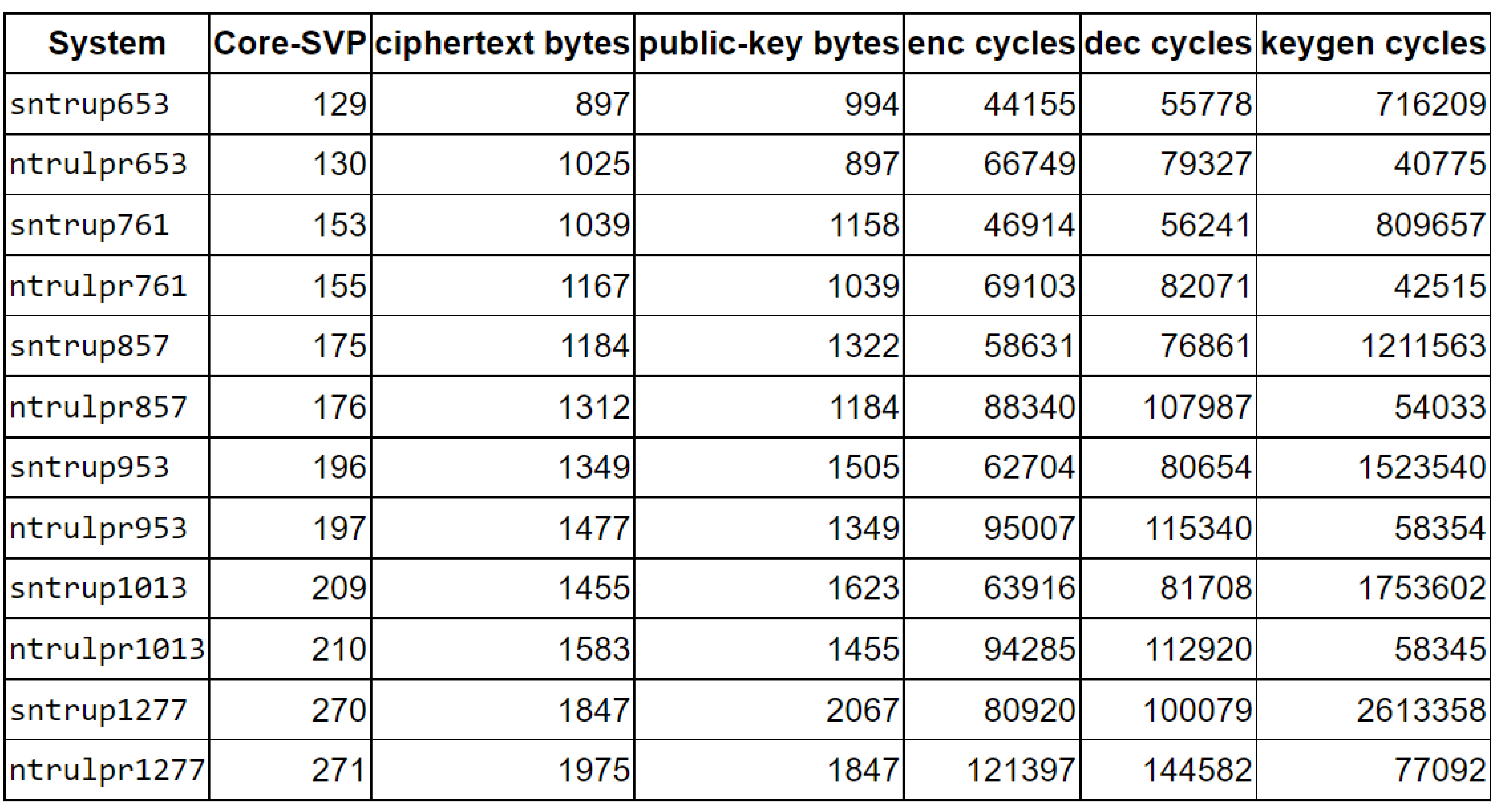
Symbolic-testing tools check that optimized software equals reference software. "Equals": gives the same outputs **for all inputs**.

Today's tools are surprisingly easy to use and quickly handle many post-quantum subroutines.

This talk: new `saferewrite` symbolic-testing tool. Open source from `https://pqsrc.cr.yp.to`.

Under the hood, doing most of the work: `valgrind`; its VEX library; Z3 theorem prover; `angr.io` binary-analysis/symbolic-execution toolkit.

NTRU Prime - Performance

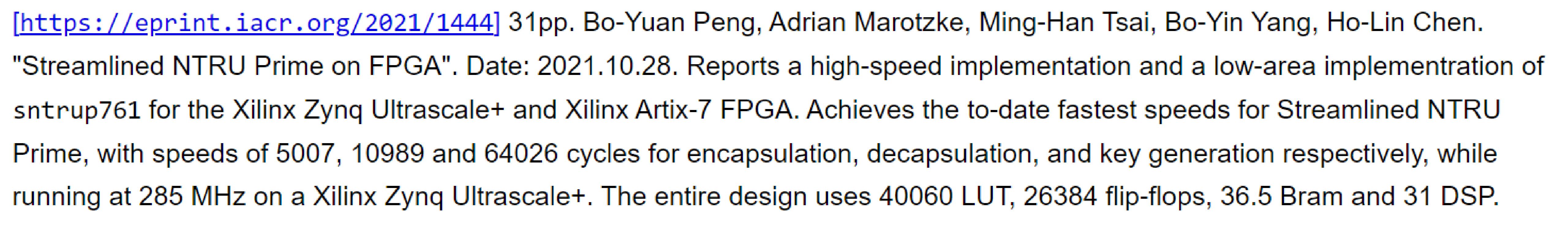


System	Core-SVP	ciphertext bytes	public-key bytes	enc cycles	dec cycles	keygen cycles
sntrup653	129	897	994	44155	55778	716209
ntrulpr653	130	1025	897	66749	79327	40775
sntrup761	153	1039	1158	46914	56241	809657
ntrulpr761	155	1167	1039	69103	82071	42515
sntrup857	175	1184	1322	58631	76861	1211563
ntrulpr857	176	1312	1184	88340	107987	54033
sntrup953	196	1349	1505	62704	80654	1523540
ntrulpr953	197	1477	1349	95007	115340	58354
sntrup1013	209	1455	1623	63916	81708	1753602
ntrulpr1013	210	1583	1455	94285	112920	58345
sntrup1277	270	1847	2067	80920	100079	2613358
ntrulpr1277	271	1975	1847	121397	144582	77092

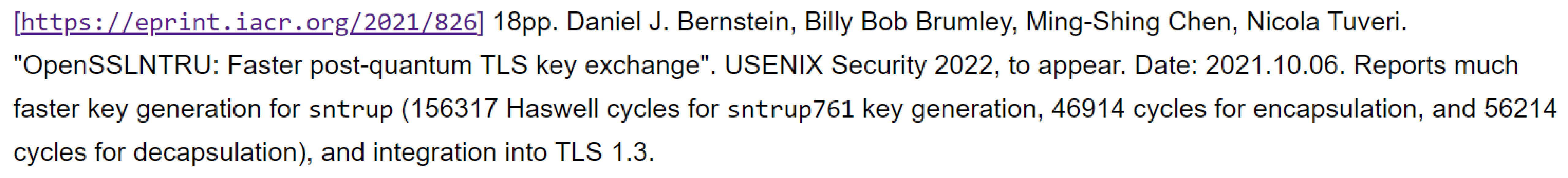
https://ntruprime.cr.yp.to/speed.html (version 2021.06.04)

NTRU Prime - Performance

- Strategies to improve performance: FPGAs, batch key generation



[https://eprint.iacr.org/2021/1444] 31pp. Bo-Yuan Peng, Adrian Marotzke, Ming-Han Tsai, Bo-Yin Yang, Ho-Lin Chen. "Streamlined NTRU Prime on FPGA". Date: 2021.10.28. Reports a high-speed implementation and a low-area implementration of `sntrup761` for the Xilinx Zynq Ultrascale+ and Xilinx Artix-7 FPGA. Achieves the to-date fastest speeds for Streamlined NTRU Prime, with speeds of 5007, 10989 and 64026 cycles for encapsulation, decapsulation, and key generation respectively, while running at 285 MHz on a Xilinx Zynq Ultrascale+. The entire design uses 40060 LUT, 26384 flip-flops, 36.5 Bram and 31 DSP.



[https://eprint.iacr.org/2021/826] 18pp. Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Nicola Tuveri. "OpenSSLNTRU: Faster post-quantum TLS key exchange". USENIX Security 2022, to appear. Date: 2021.10.06. Reports much faster key generation for `sntrup` (156317 Haswell cycles for `sntrup761` key generation, 46914 cycles for encapsulation, and 56214 cycles for decapsulation), and integration into TLS 1.3.

https://ntruprime.cr.yp.to/papers.html (version 2021.10.28)

NTRU Prime – “Official Comments”

- NTRU Prime team: Complaint that NIST has not specified a metric for estimating costs of attacks, leading to incorrect estimates of security strength for Kyber, NTRU Prime, etc.
- NTRU Prime team: Complaint that NIST evaluations are biased against NTRU Prime
- D. Bernstein: Announcement of “saferewrite” tool for software verification
- D. Bernstein: Complaint of misconduct by D. Apon (17 pages)
- C. Peikert: Complaint that NTRU Prime FAQ makes misleading statements about patents
- D. Bernstein: Complaint that NIST has been discouraging public discussion of patent issues
- D. Moody: NIST position regarding bad behavior on the PQC Forum
- Followed by more bad behavior on the PQC Forum
- NTRU Prime team: Announcement on “Risks of lattice KEMs” (99 pages)

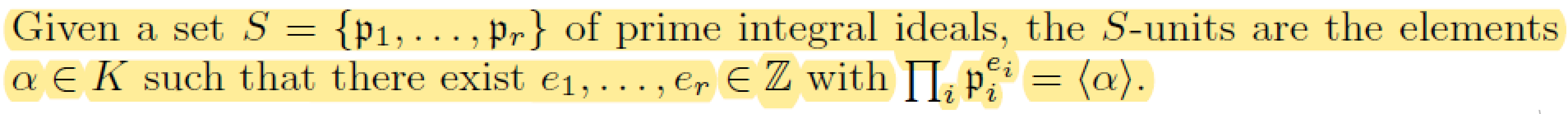
Let’s take a short break…

This Photo by Unknown Author is licensed under CC BY-SA

S-unit Attacks

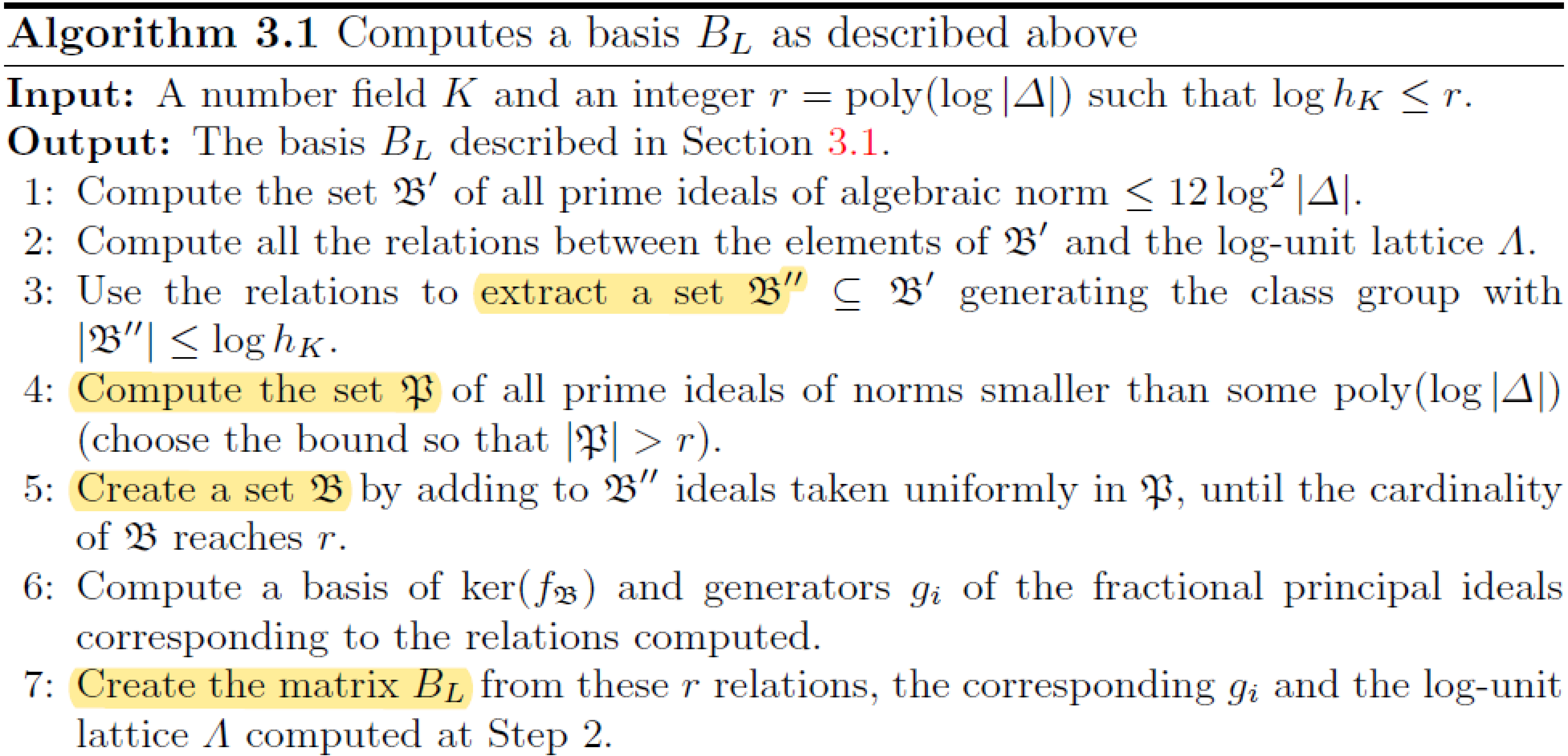
- Folklore? (see, e.g., emails by Dan Bernstein, circa 2016)
  - Given an ideal I, find an element g of I, then find an S-unit u such that gu is short
- Rigorous analysis by Pellet-Mary et al ([PHS19]: https://eprint.iacr.org/2019/215)
  - Using the log-S-unit lattice; some details seem awkward or sub-optimal
- Improved by Bernard and Roux-Langlois ([BR20]: https://eprint.iacr.org/2020/1081)
  - Nicer variant of log-S-unit lattice; numerics suggest it finds shorter vectors; easier to read
  - Further improvements by Bernard et al ([BLNR21]: https://eprint.iacr.org/2021/1384)
- Dan Bernstein (2021): Conjecture that S-unit attacks can do much better (https://cr.yp.to/talks/2021.08.20/slides-djb-20210820-sunitattacks-4x3.pdf)
  - Limited evidence for this conjecture; hard to see asymptotic scaling from numerics
  - Preprint ([BL21]: https://eprint.iacr.org/2021/1428) claims that the analysis by [PHS19], applying the Gaussian heuristic to the log-unit lattice, is not accurate
  - Also thinks the [BR20] algorithm is better than [PHS19]; thinks they should have cited him

[PHS19] version of the log-S-unit lattice



Given a set $S = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_r\}$ of prime integral ideals, the S-units are the elements $\alpha \in K$ such that there exist $e_1, \ldots, e_r \in \mathbb{Z}$ with $\prod_i \mathfrak{p}_i^{e_i} = \langle \alpha \rangle$.

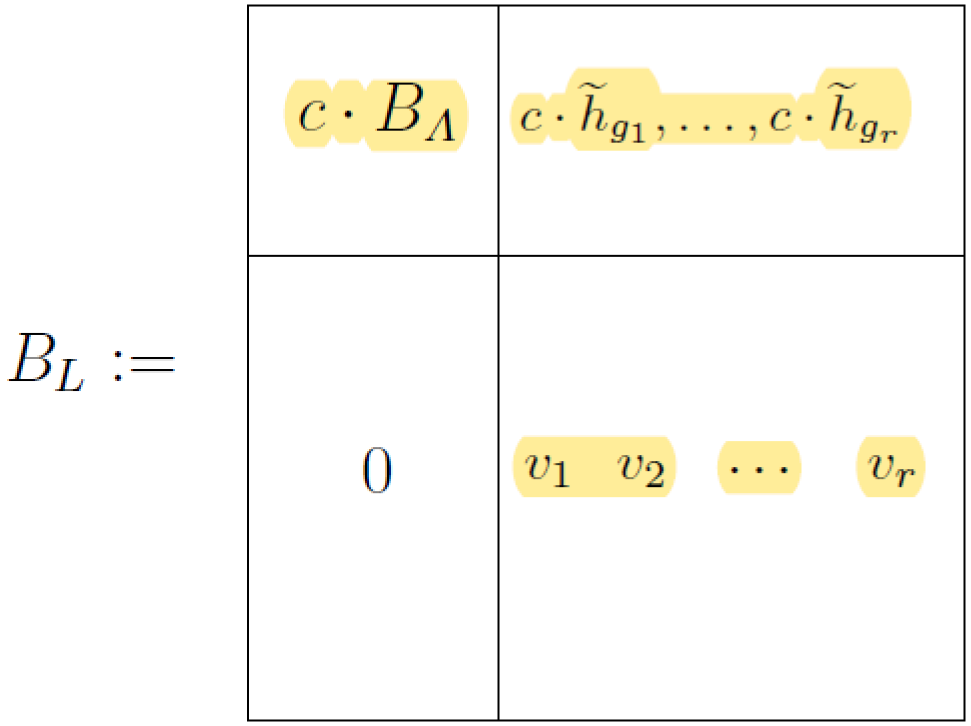
Idea: α is a unit modulo the p_i



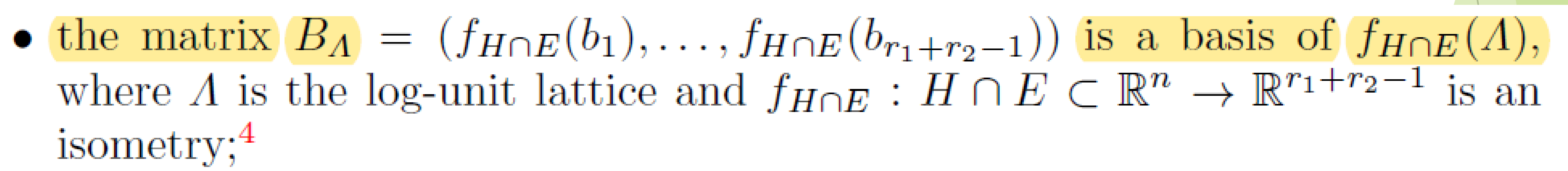
```
Algorithm 3.1 Computes a basis $B_L$ as described above
Input: A number field $K$ and an integer $r = \mathrm{poly}(\log|\Delta|)$ such that $\log h_K \leq r$.
Output: The basis $B_L$ described in Section 3.1.
1: Compute the set $\mathfrak{B}'$ of all prime ideals of algebraic norm $\leq 12\log^2|\Delta|$.
2: Compute all the relations between the elements of $\mathfrak{B}'$ and the log-unit lattice $\Lambda$.
3: Use the relations to extract a set $\mathfrak{B}'' \subseteq \mathfrak{B}'$ generating the class group with
   $|\mathfrak{B}''| \leq \log h_K$.
4: Compute the set $\mathfrak{P}$ of all prime ideals of norms smaller than some $\mathrm{poly}(\log|\Delta|)$
   (choose the bound so that $|\mathfrak{P}| > r$).
5: Create a set $\mathfrak{B}$ by adding to $\mathfrak{B}''$ ideals taken uniformly in $\mathfrak{P}$, until the cardinality
   of $\mathfrak{B}$ reaches $r$.
6: Compute a basis of $\ker(f_{\mathfrak{B}})$ and generators $g_i$ of the fractional principal ideals
   corresponding to the relations computed.
7: Create the matrix $B_L$ from these $r$ relations, the corresponding $g_i$ and the log-unit
   lattice $\Lambda$ computed at Step 2.
```

Note: this is P, not B

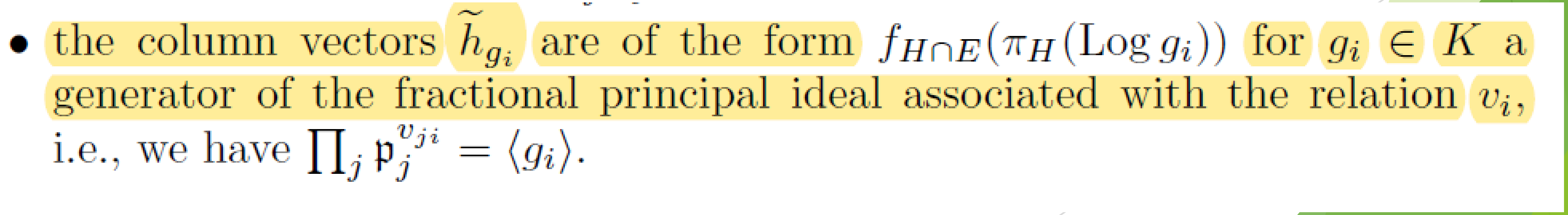
[PHS19] version of the log-S-unit lattice



$$B_L := \left(\begin{array}{|c|c|} \hline c \cdot B_\Lambda & c \cdot \widetilde{h}_{g_1}, \dots, c \cdot \widetilde{h}_{g_r} \\ \hline 0 & v_1 \quad v_2 \quad \cdots \quad v_r \\ \hline \end{array}\right)$$

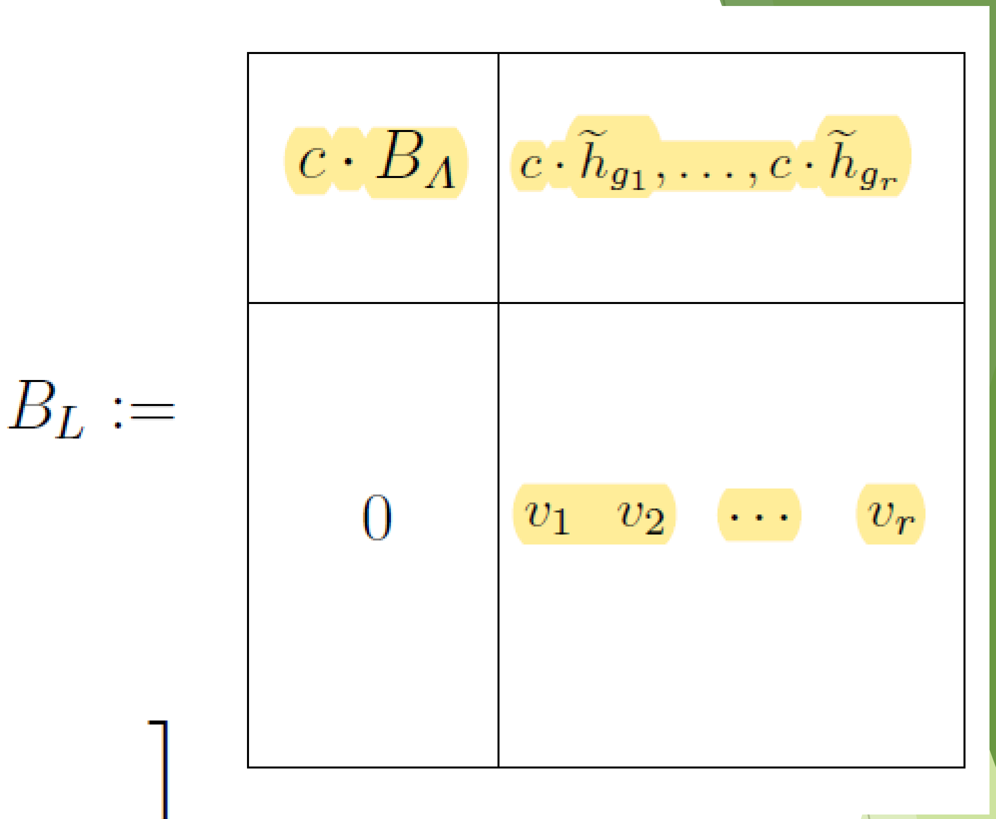


- the matrix $B_\Lambda = (f_{H \cap E}(b_1), \dots, f_{H \cap E}(b_{r_1+r_2-1}))$ is a basis of $f_{H \cap E}(\Lambda)$, where Λ is the log-unit lattice and $f_{H \cap E} : H \cap E \subset \mathbb{R}^n \to \mathbb{R}^{r_1+r_2-1}$ is an isometry;[4]

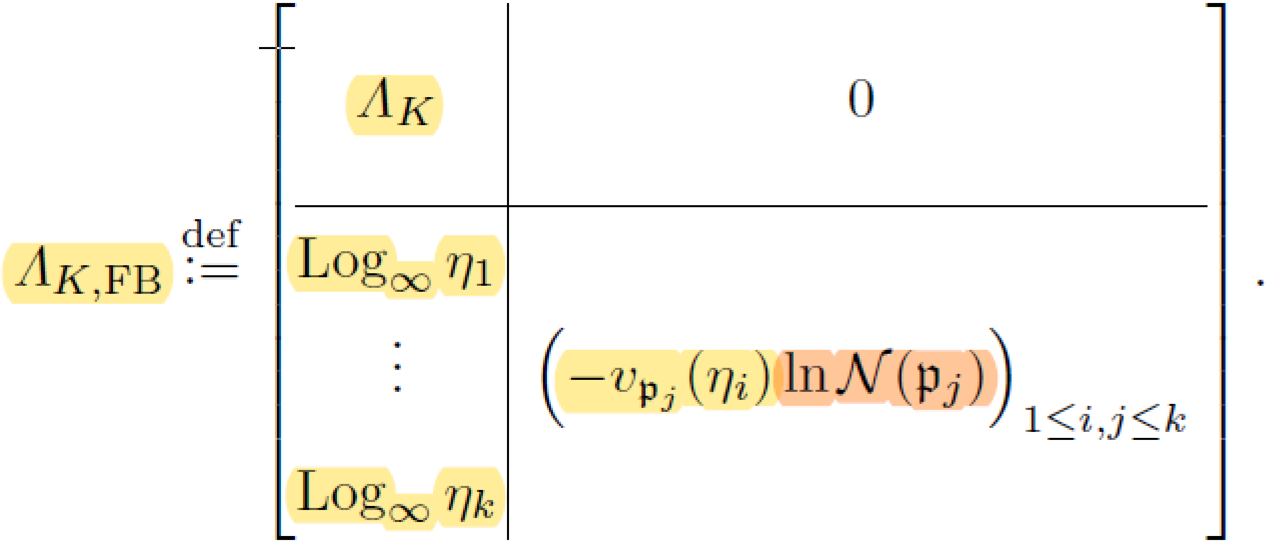


- the column vectors $\widetilde{h}_{g_i}$ are of the form $f_{H \cap E}(\pi_H(\operatorname{Log} g_i))$ for $g_i \in K$ a generator of the fractional principal ideal associated with the relation v_i, i.e., we have $\prod_j \mathfrak{p}_j^{v_{ji}} = \langle g_i \rangle$.

[PHS19] and [BR20] versions of the log-S-unit lattice

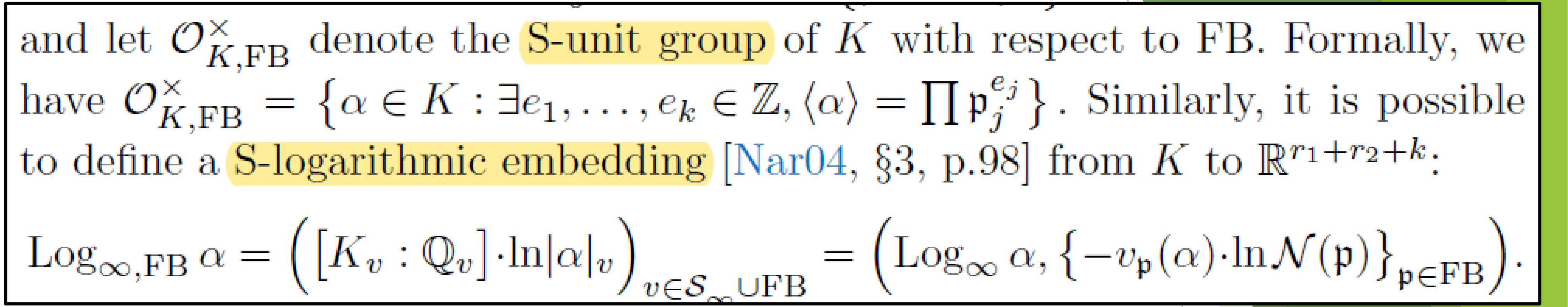


$$B_L := \begin{array}{|c|c|} \hline c \cdot B_\Lambda & c \cdot \widetilde{h}_{g_1}, \dots, c \cdot \widetilde{h}_{g_r} \\ \hline 0 & v_1 \quad v_2 \quad \cdots \quad v_r \\ \hline \end{array}$$



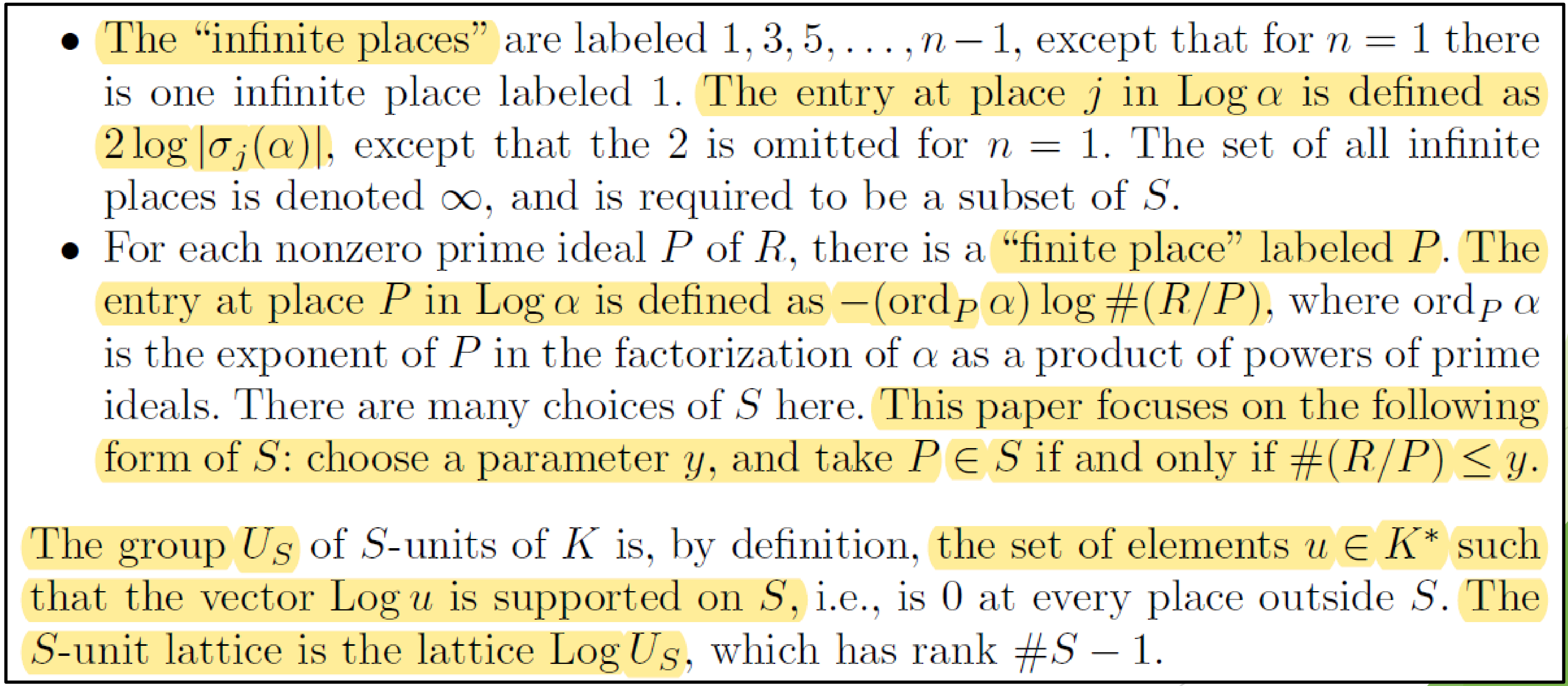
$$\Lambda_{K,\mathrm{FB}} \overset{\text{def}}{:=} \left[\begin{array}{c|c} \Lambda_K & 0 \\ \hline \mathrm{Log}_\infty \eta_1 & \\ \vdots & \left(-v_{\mathfrak{p}_j}(\eta_i) \ln \mathcal{N}(\mathfrak{p}_j)\right)_{1 \le i,j \le k} \\ \mathrm{Log}_\infty \eta_k & \end{array}\right].$$

[BR20] and [BL21] versions of the log map



and let $\mathcal{O}_{K,\mathrm{FB}}^{\times}$ denote the S-unit group of K with respect to FB. Formally, we have $\mathcal{O}_{K,\mathrm{FB}}^{\times} = \{\alpha \in K : \exists e_1, \ldots, e_k \in \mathbb{Z}, \langle\alpha\rangle = \prod \mathfrak{p}_j^{e_j}\}$. Similarly, it is possible to define a S-logarithmic embedding [Nar04, §3, p.98] from K to $\mathbb{R}^{r_1+r_2+k}$:

$$\mathrm{Log}_{\infty,\mathrm{FB}}\, \alpha = \Big([K_v : \mathbb{Q}_v]\cdot \ln|\alpha|_v\Big)_{v\in\mathcal{S}_\infty\cup\mathrm{FB}} = \Big(\mathrm{Log}_\infty\, \alpha, \{-v_{\mathfrak{p}}(\alpha)\cdot \ln \mathcal{N}(\mathfrak{p})\}_{\mathfrak{p}\in\mathrm{FB}}\Big).$$

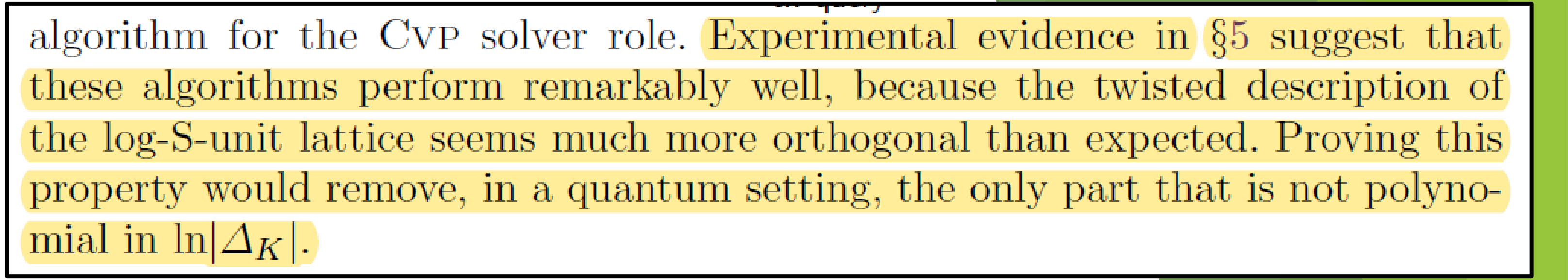


- The "infinite places" are labeled $1, 3, 5, \ldots, n-1$, except that for $n = 1$ there is one infinite place labeled 1. The entry at place j in $\mathrm{Log}\,\alpha$ is defined as $2\log|\sigma_j(\alpha)|$, except that the 2 is omitted for $n = 1$. The set of all infinite places is denoted ∞, and is required to be a subset of S.
- For each nonzero prime ideal P of R, there is a "finite place" labeled P. The entry at place P in $\mathrm{Log}\,\alpha$ is defined as $-(\mathrm{ord}_P\,\alpha)\log\#(R/P)$, where $\mathrm{ord}_P\,\alpha$ is the exponent of P in the factorization of α as a product of powers of prime ideals. There are many choices of S here. This paper focuses on the following form of S: choose a parameter y, and take $P \in S$ if and only if $\#(R/P) \leq y$.

The group U_S of S-units of K is, by definition, the set of elements $u \in K^*$ such that the vector $\mathrm{Log}\,u$ is supported on S, i.e., is 0 at every place outside S. The S-unit lattice is the lattice $\mathrm{Log}\,U_S$, which has rank $\#S - 1$.

Note: #(R/P) = infinity? Should be the algebraic norm N(P)?

Numerical results from [BR20]



algorithm for the CVP solver role. Experimental evidence in §5 suggest that these algorithms perform remarkably well, because the twisted description of the log-S-unit lattice seems much more orthogonal than expected. Proving this property would remove, in a quantum setting, the only part that is not polynomial in $\ln|\Delta_K|$.

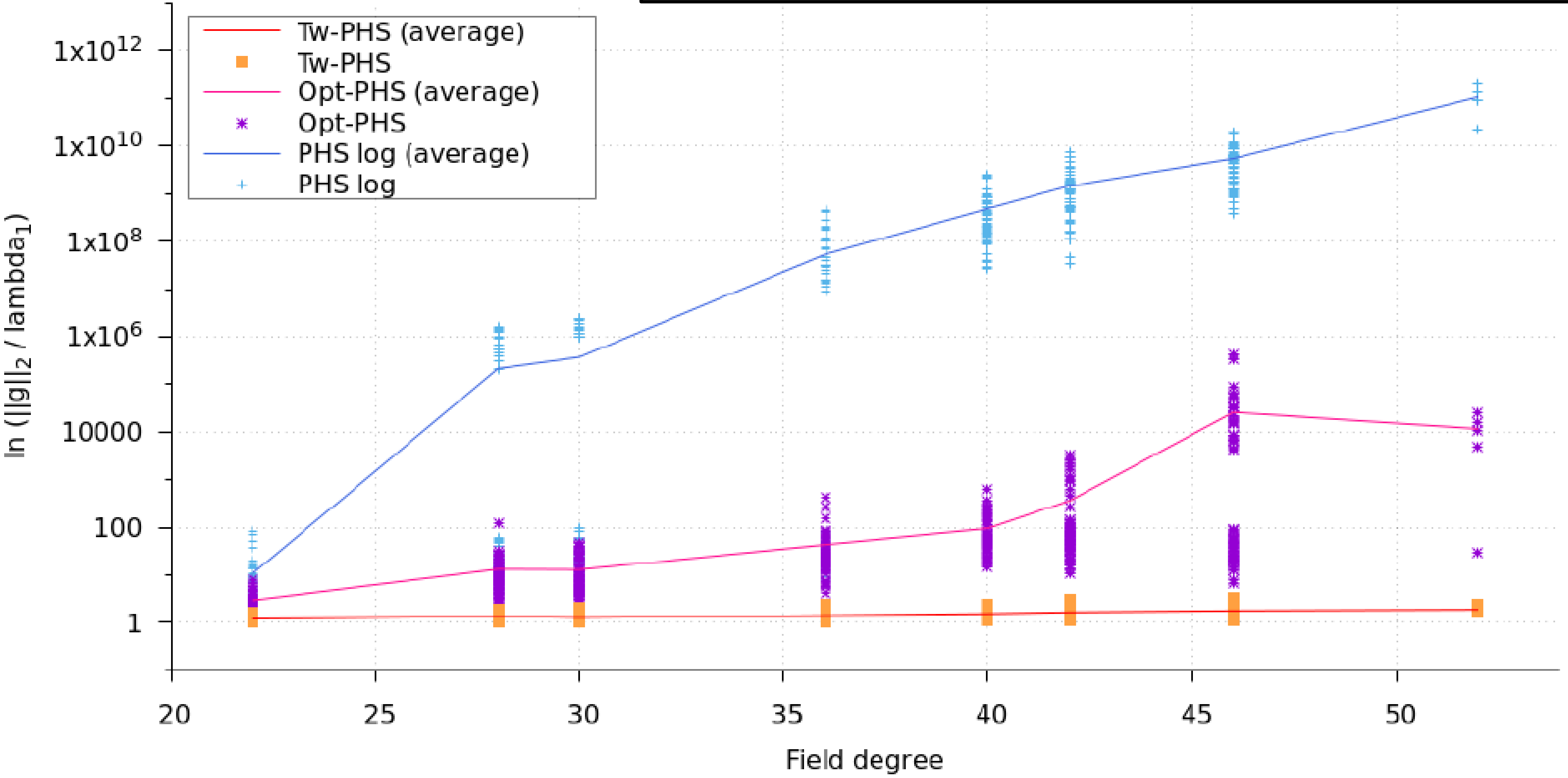


Fig. 1.1 – Approximation factors reached by Tw-PHS, Opt-PHS and PHS fo cyclotomic fields of conductors 23, 29, 31, 37, 41, 43, 47 and 53 (in log scale).

Questions:

What is the asymptotic scaling of this attack?

Does this affect our concrete security estimates?

For comparison, LLL has approx. ratio ~ 1.022^n

LLL wins, until n gets large

Numerical results from [BR20]

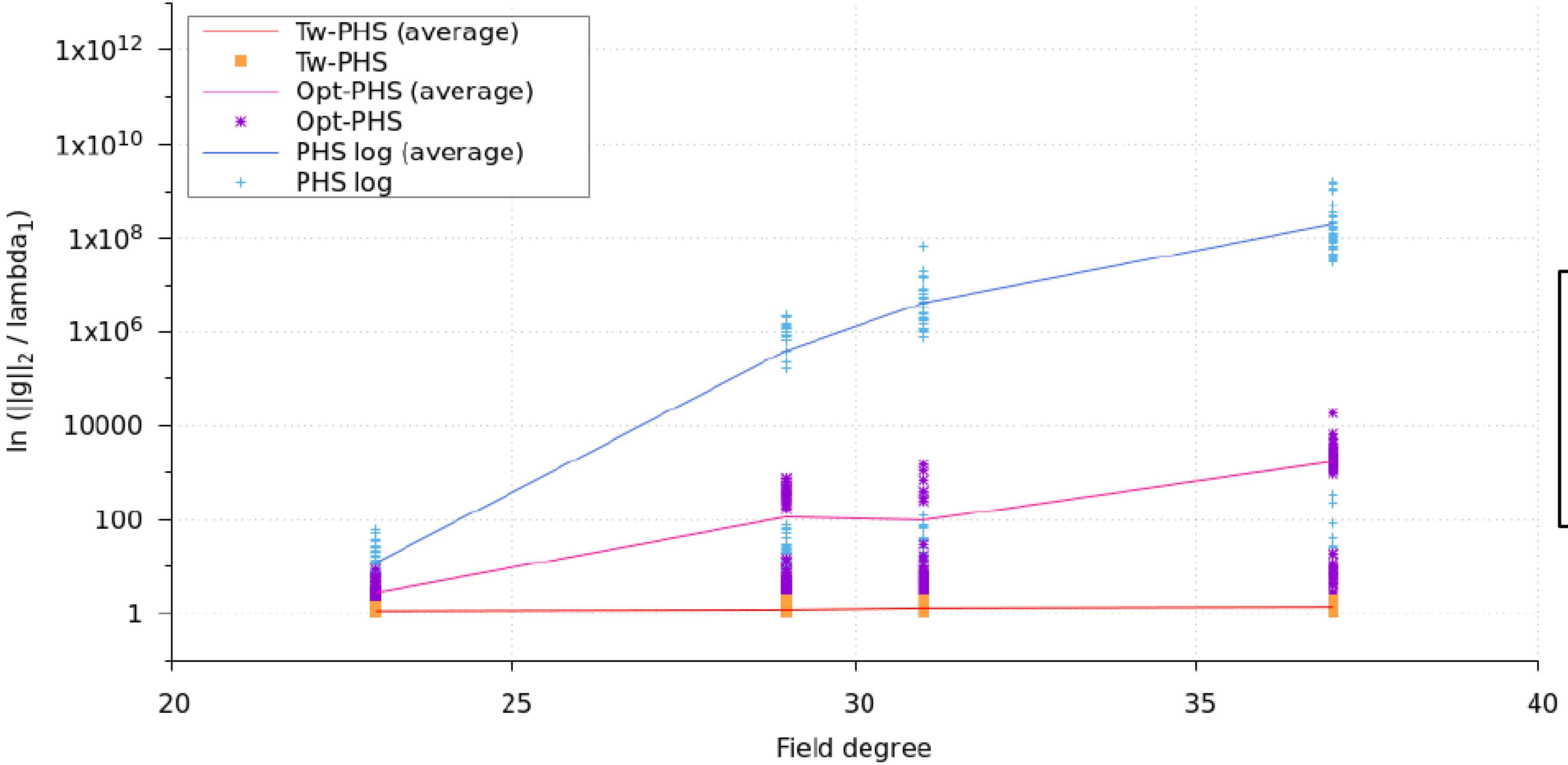


Fig. 5.4 – Approximation factors reached by Tw-PHS, Opt-PHS and PHS for NTRU Prime fields of degrees 23, 29, 31 and 37 (in log scale).

Question:

Any difference b/w cyclotomic and NTRU Prime fields?

Numerical results from [BLNR21]

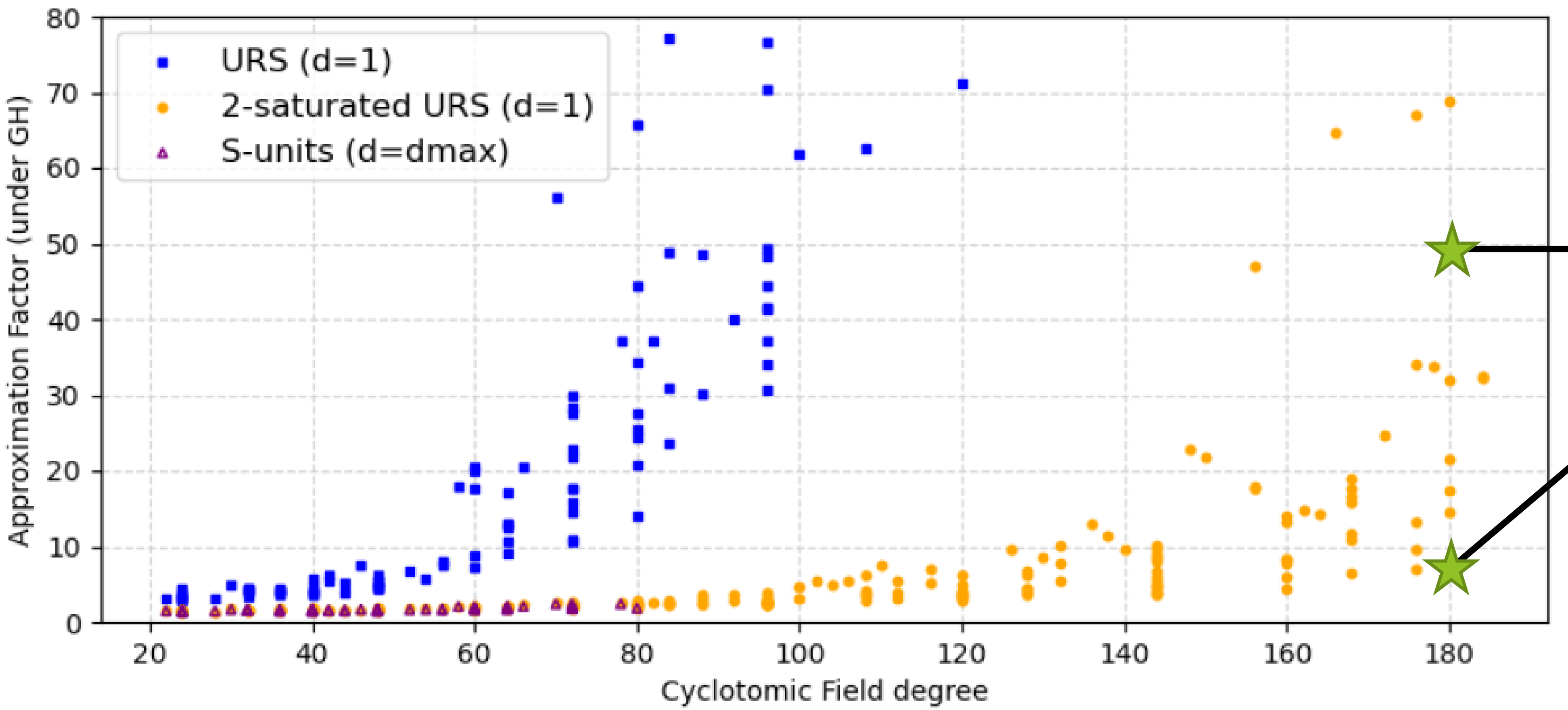


FIG. 1.1 – Approximation factors, estimated with Gaussian Heuristic, reached by Tw-PHS for cyclotomic fields of degree $\varphi(m) < 190$ with $h_m^+ = 1$ on lattices L_{urs}, L_{sat} and L_{su} (when available).

Blue and gold dots are weaker versions of the Twisted-PHS algorithm

LLL has approx. ratio ~ 1.022^n

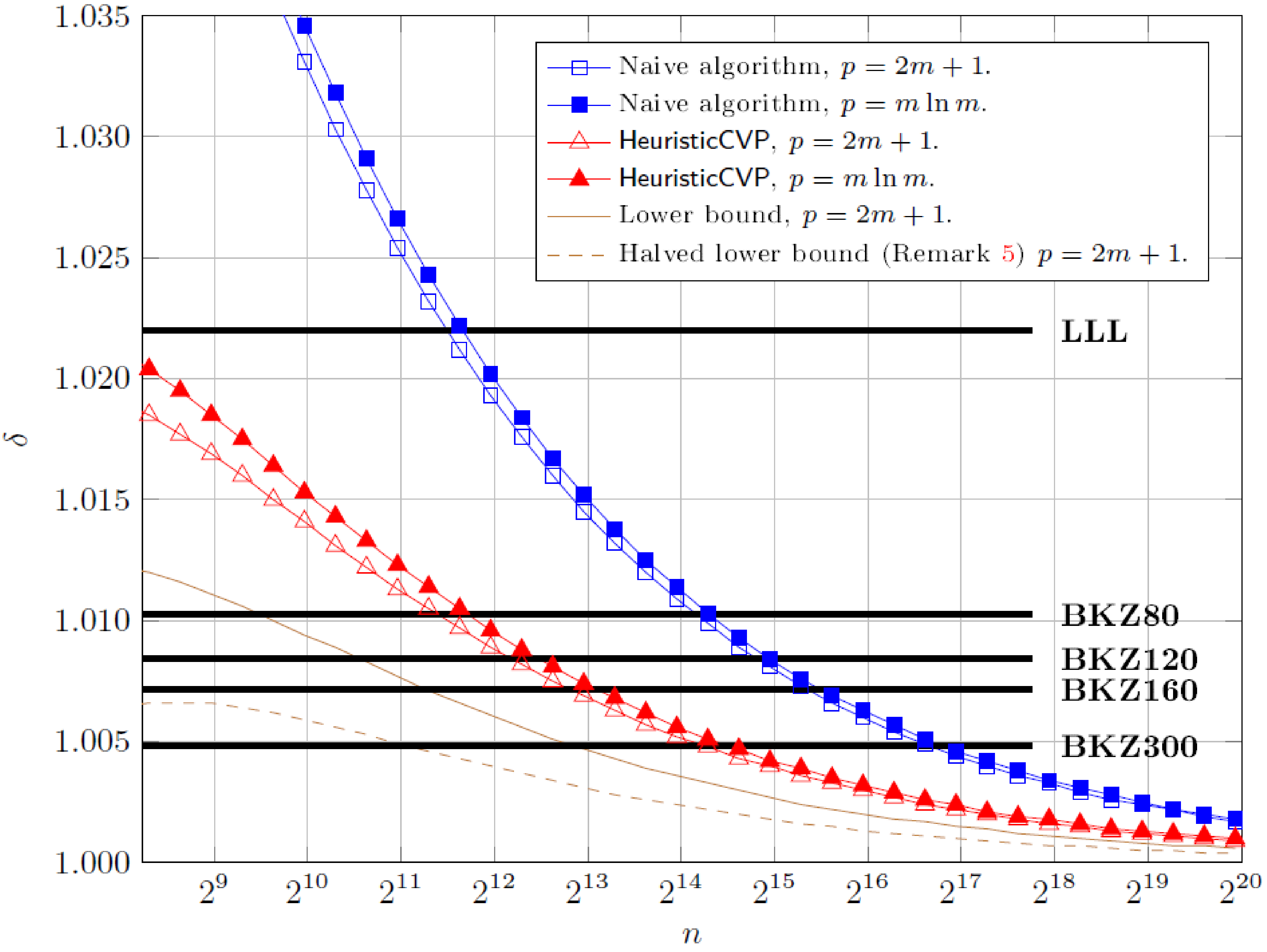
BKZ80 has approx. ratio ~ 1.01^n

Purple dots are the full version of Twisted-PHS

For comparison: older q. algorithm for Ideal-SVP (using units, not S-units) [DPW19]

- Root Hermite factor delta = eta^(1/n) ~ (approx. ratio)^(1/n)
- Q. algorithm has delta = exp(O˜(1/√n)) → 1 + O˜(1/√n)
- Graph revised in Aug. 2021, fixing mistake found by D. Bernstein
- Where does Twisted-PHS lie on this graph?

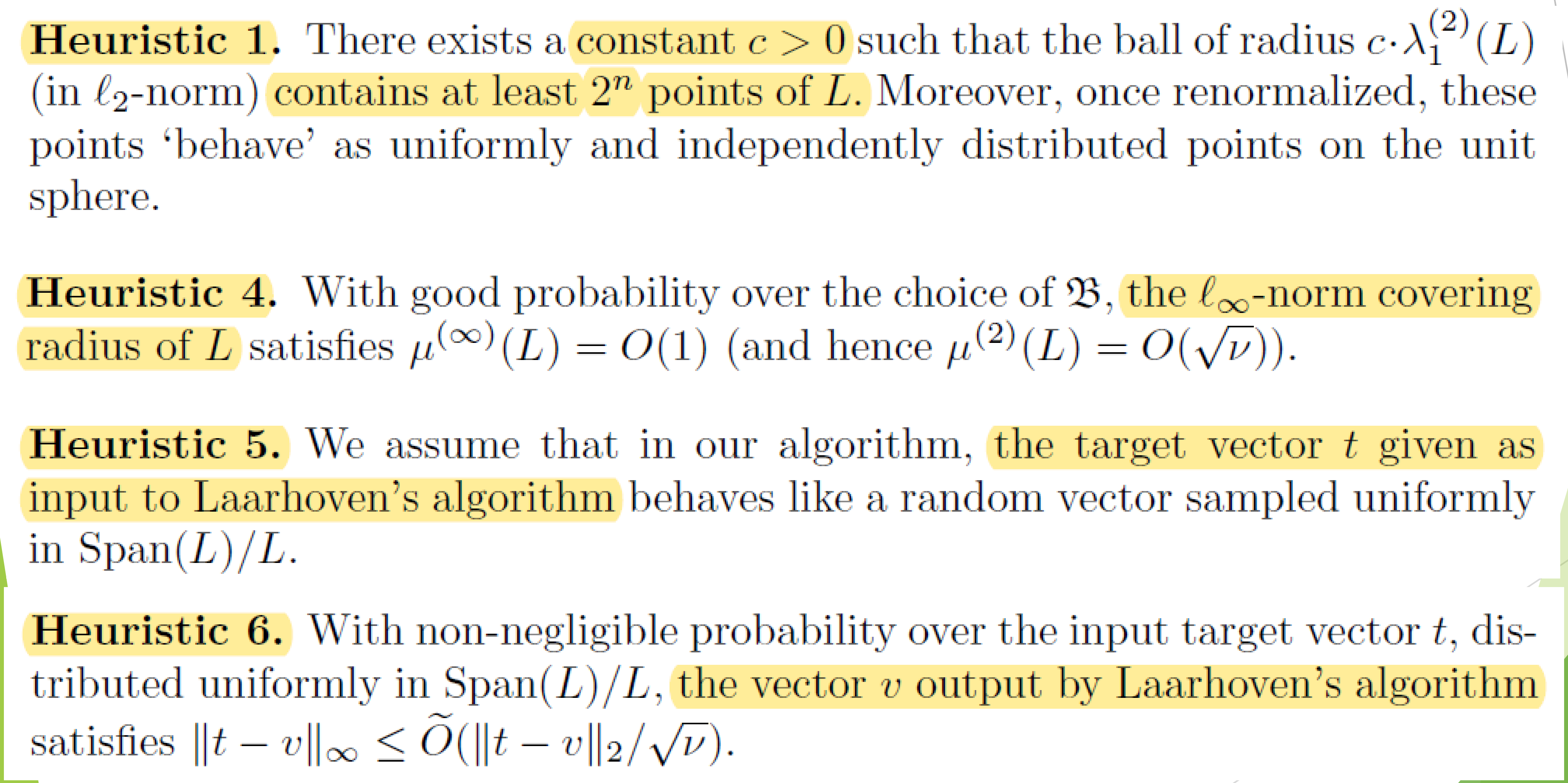
Fig. 5: Quality of Quantum Ideal-SVP vs. LLL and BKZ.



What is the asymptotic scaling of these S-unit attacks?

- [PHS19] wanted to show an upper-bound (i.e., given subexponential computing resources, S-unit attacks succeed)
  - Made some heuristic assumptions, including Gaussian heuristic
- Subsequent arguments: how tight is this upper-bound? (i.e., could S-unit attacks perform better than what [PHS19] showed?)
- Folklore: for random lattices, the Gaussian heuristic is pretty close to reality
  - Caveat: log-unit lattice has some special features that don’t look random
  - Caveat: Gaussian heuristic can be used to predict many properties of a lattice, and some of these predictions are more robust than others

Assumptions used in [PHS19]



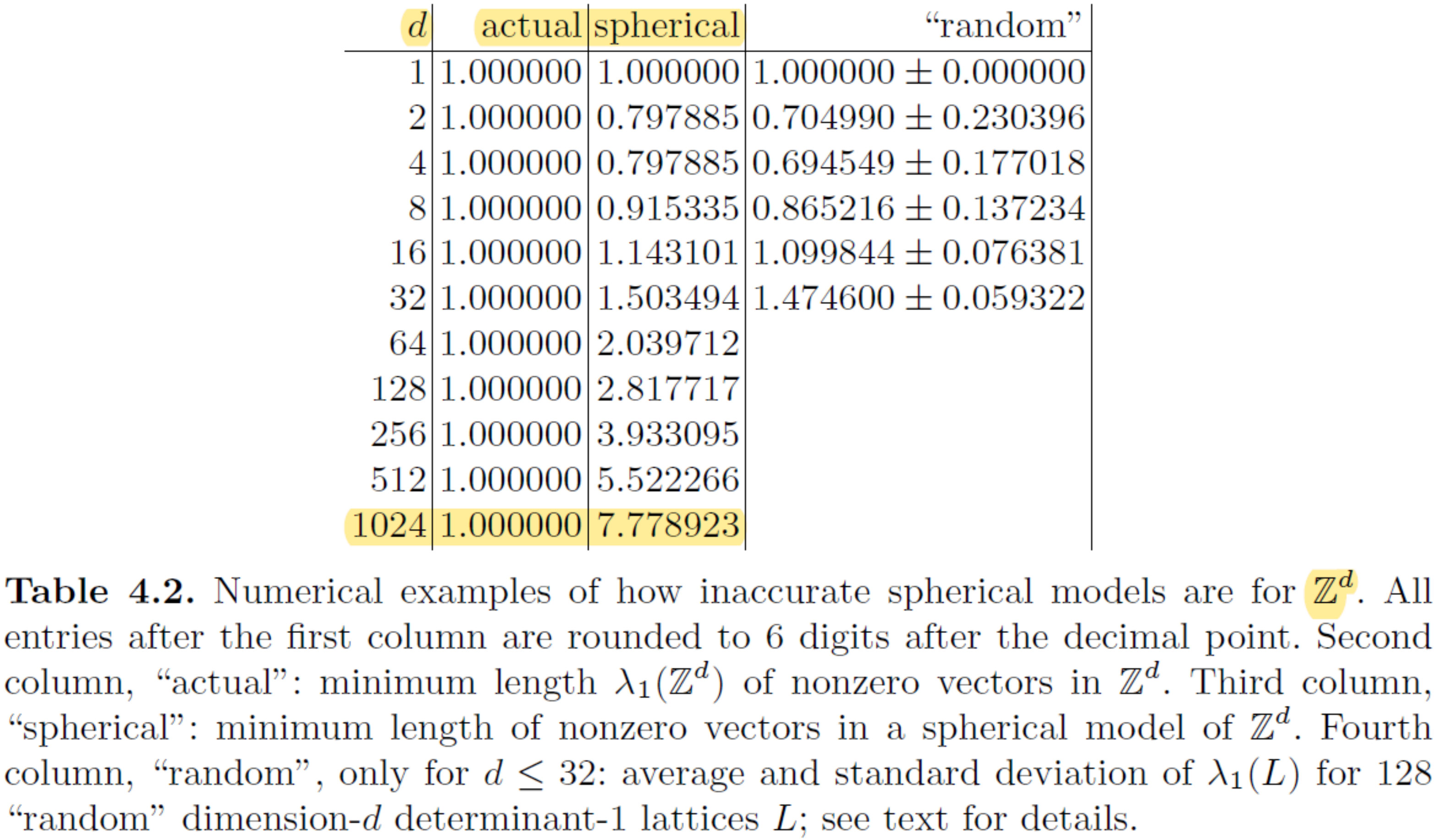
Heuristic 1. There exists a constant $c > 0$ such that the ball of radius $c \cdot \lambda_1^{(2)}(L)$ (in ℓ_2-norm) contains at least 2^n points of L. Moreover, once renormalized, these points 'behave' as uniformly and independently distributed points on the unit sphere.

Heuristic 4. With good probability over the choice of $\mathfrak{B}$, the ℓ_∞-norm covering radius of L satisfies $\mu^{(\infty)}(L) = O(1)$ (and hence $\mu^{(2)}(L) = O(\sqrt{\nu})$).

Heuristic 5. We assume that in our algorithm, the target vector t given as input to Laarhoven's algorithm behaves like a random vector sampled uniformly in $\mathrm{Span}(L)/L$.

Heuristic 6. With non-negligible probability over the input target vector t, distributed uniformly in $\mathrm{Span}(L)/L$, the vector v output by Laarhoven's algorithm satisfies $\|t - v\|_\infty \leq \widetilde{O}(\|t - v\|_2/\sqrt{\nu})$.

[BL21]: Examples of how the Gaussian heuristic fails on log-S-unit lattices

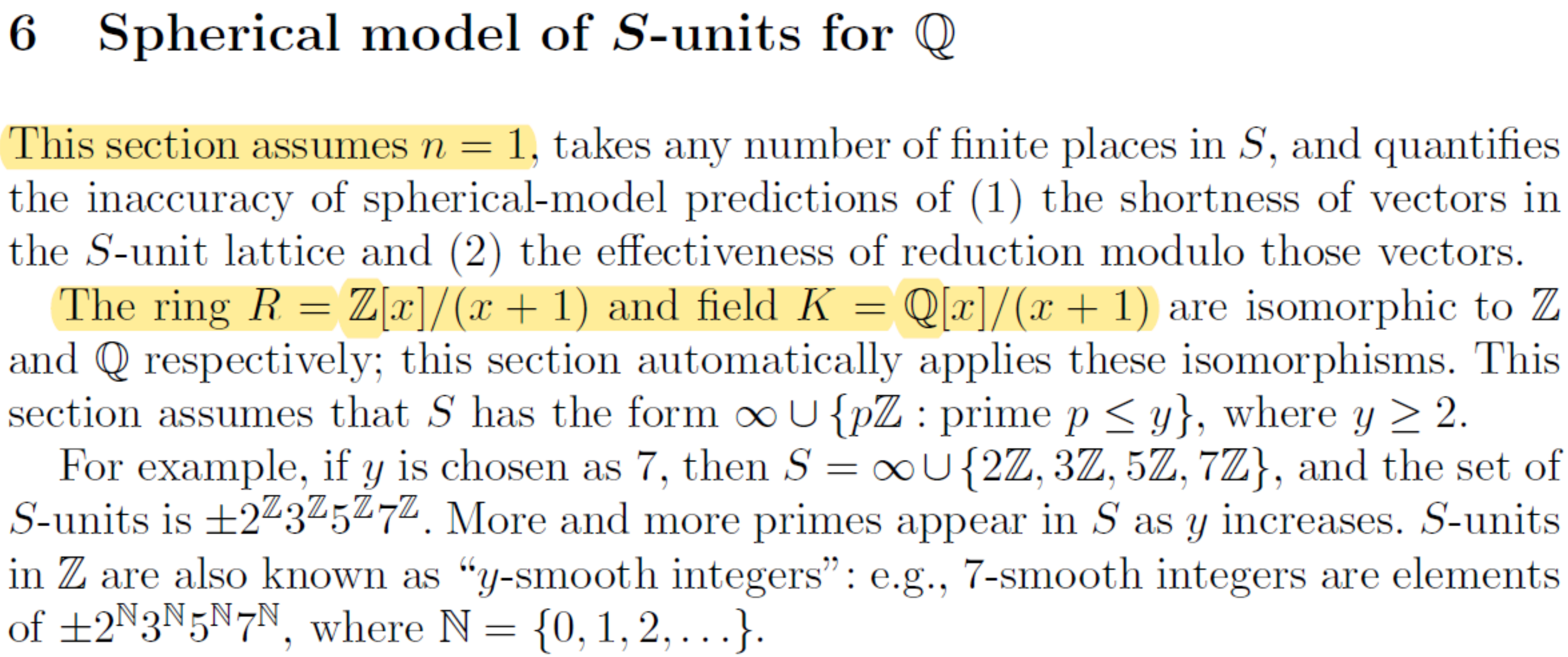


d	actual	spherical	"random"
1	1.000000	1.000000	1.000000 ± 0.000000
2	1.000000	0.797885	0.704990 ± 0.230396
4	1.000000	0.797885	0.694549 ± 0.177018
8	1.000000	0.915335	0.865216 ± 0.137234
16	1.000000	1.143101	1.099844 ± 0.076381
32	1.000000	1.503494	1.474600 ± 0.059322
64	1.000000	2.039712	
128	1.000000	2.817717	
256	1.000000	3.933095	
512	1.000000	5.522266	
1024	1.000000	7.778923	

Table 4.2. Numerical examples of how inaccurate spherical models are for $\mathbb{Z}^d$. All entries after the first column are rounded to 6 digits after the decimal point. Second column, "actual": minimum length $\lambda_1(\mathbb{Z}^d)$ of nonzero vectors in $\mathbb{Z}^d$. Third column, "spherical": minimum length of nonzero vectors in a spherical model of $\mathbb{Z}^d$. Fourth column, "random", only for $d \leq 32$: average and standard deviation of $\lambda_1(L)$ for 128 "random" dimension-d determinant-1 lattices L; see text for details.

Question:
Do any of the examples in [BL21] disprove any of the assumptions made in [PHS19]?

[BL21]: Examples of how the Gaussian heuristic fails on log-S-unit lattices



6 Spherical model of S-units for $\mathbb{Q}$

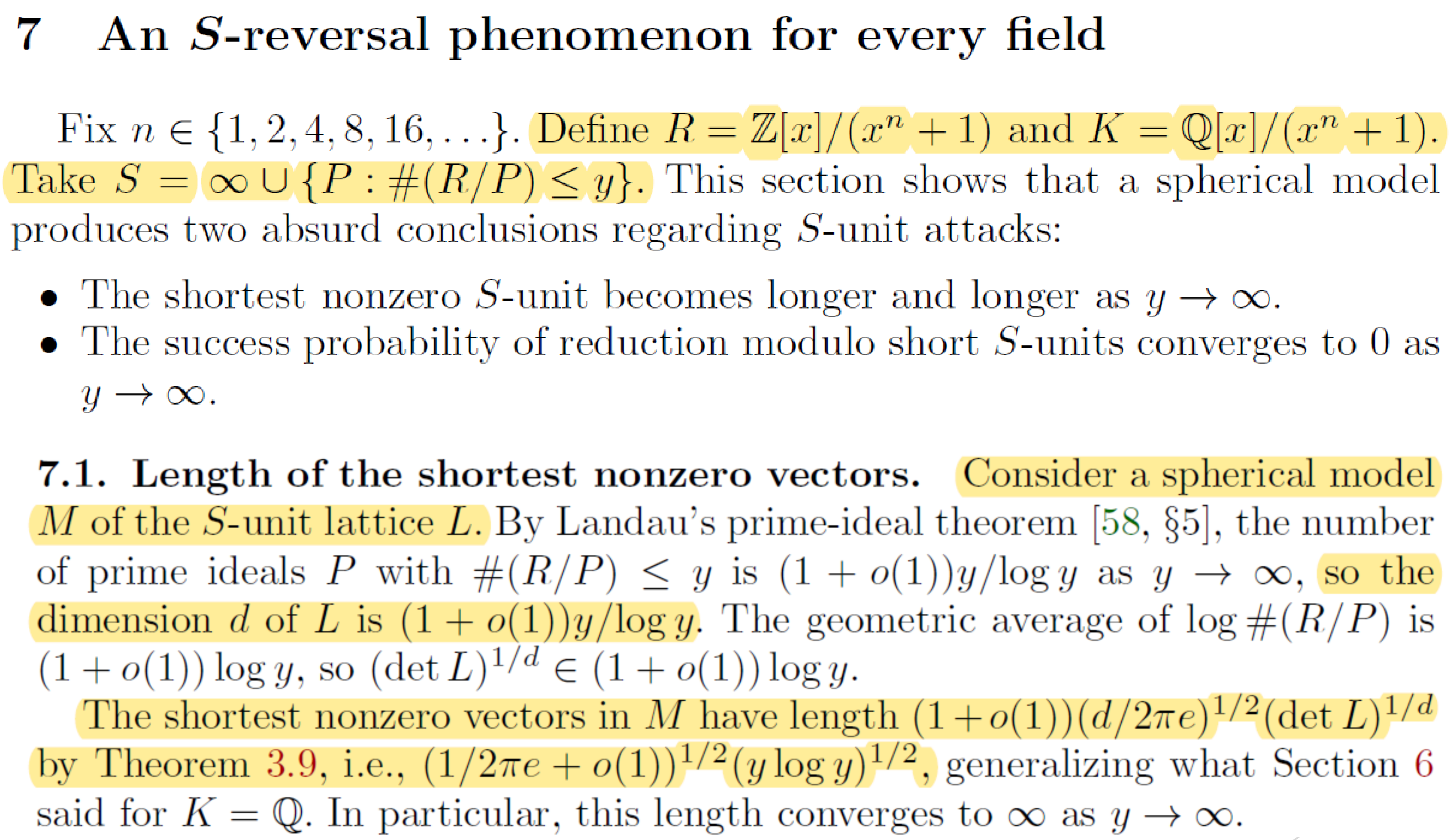
This section assumes $n = 1$, takes any number of finite places in S, and quantifies the inaccuracy of spherical-model predictions of (1) the shortness of vectors in the S-unit lattice and (2) the effectiveness of reduction modulo those vectors.

The ring $R = \mathbb{Z}[x]/(x + 1)$ and field $K = \mathbb{Q}[x]/(x + 1)$ are isomorphic to $\mathbb{Z}$ and $\mathbb{Q}$ respectively; this section automatically applies these isomorphisms. This section assumes that S has the form $\infty \cup \{p\mathbb{Z} : \text{prime } p \leq y\}$, where $y \geq 2$.

For example, if y is chosen as 7, then $S = \infty \cup \{2\mathbb{Z}, 3\mathbb{Z}, 5\mathbb{Z}, 7\mathbb{Z}\}$, and the set of S-units is $\pm 2^{\mathbb{Z}} 3^{\mathbb{Z}} 5^{\mathbb{Z}} 7^{\mathbb{Z}}$. More and more primes appear in S as y increases. S-units in $\mathbb{Z}$ are also known as "y-smooth integers": e.g., 7-smooth integers are elements of $\pm 2^{\mathbb{N}} 3^{\mathbb{N}} 5^{\mathbb{N}} 7^{\mathbb{N}}$, where $\mathbb{N} = \{0, 1, 2, \ldots\}$.

Question:
Do any of the examples in [BL21] disprove any of the assumptions made in [PHS19]?

[BL21]: Examples of how the Gaussian heuristic fails on log-S-unit lattices



7 An S-reversal phenomenon for every field

Fix $n \in \{1, 2, 4, 8, 16, \ldots\}$. Define $R = \mathbb{Z}[x]/(x^n + 1)$ and $K = \mathbb{Q}[x]/(x^n + 1)$. Take $S = \infty \cup \{P : \#(R/P) \leq y\}$. This section shows that a spherical model produces two absurd conclusions regarding S-unit attacks:

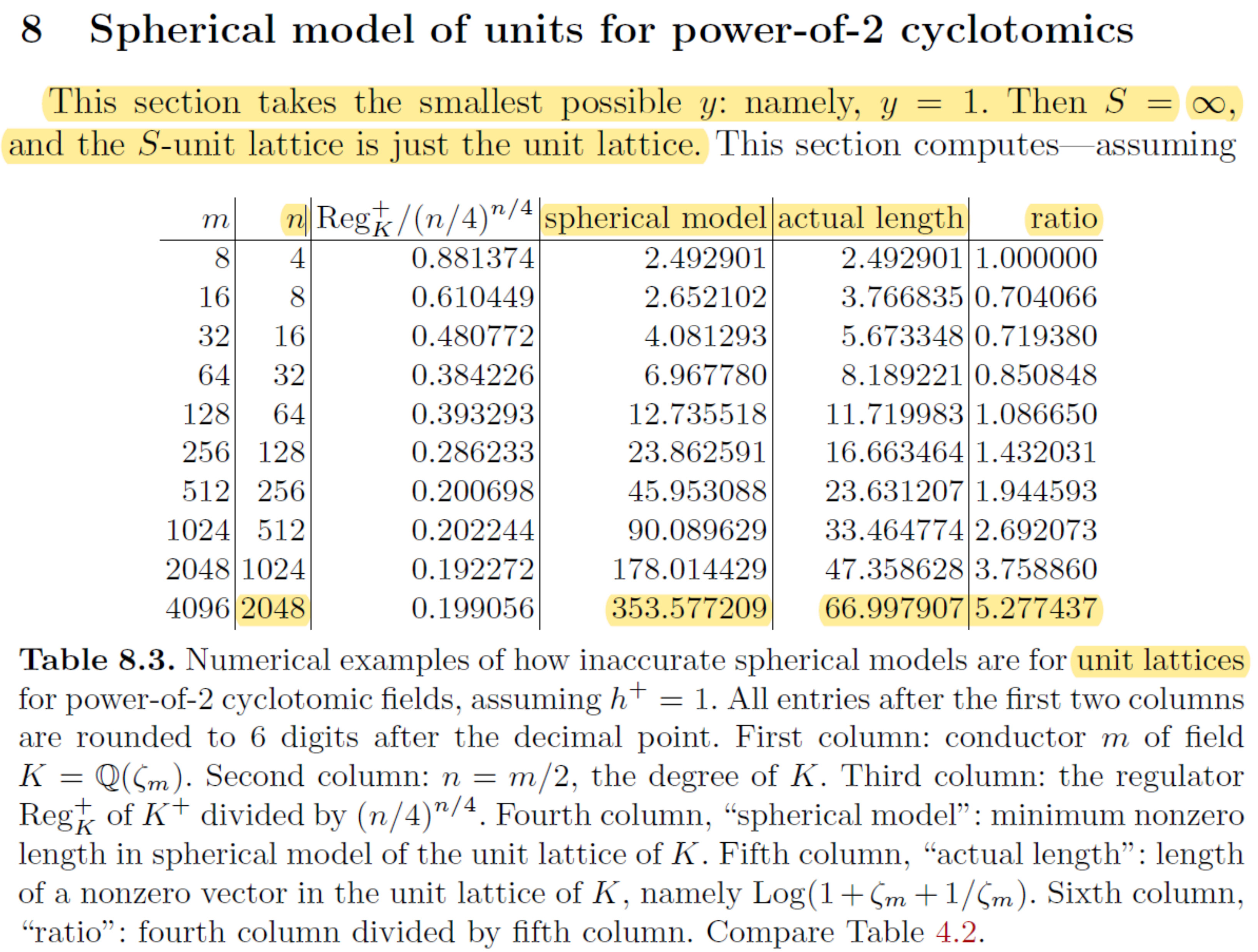
- The shortest nonzero S-unit becomes longer and longer as $y \to \infty$.
- The success probability of reduction modulo short S-units converges to 0 as $y \to \infty$.

7.1. Length of the shortest nonzero vectors. Consider a spherical model M of the S-unit lattice L. By Landau's prime-ideal theorem [58, §5], the number of prime ideals P with $\#(R/P) \leq y$ is $(1 + o(1))y/\log y$ as $y \to \infty$, so the dimension d of L is $(1 + o(1))y/\log y$. The geometric average of $\log \#(R/P)$ is $(1 + o(1)) \log y$, so $(\det L)^{1/d} \in (1 + o(1)) \log y$.

The shortest nonzero vectors in M have length $(1 + o(1))(d/2\pi e)^{1/2}(\det L)^{1/d}$ by Theorem 3.9, i.e., $(1/2\pi e + o(1))^{1/2}(y \log y)^{1/2}$, generalizing what Section 6 said for $K = \mathbb{Q}$. In particular, this length converges to ∞ as $y \to \infty$.

Question:
Do any of the examples in [BL21] disprove any of the assumptions made in [PHS19]?

[BL21]: Examples of how the Gaussian heuristic fails on log-S-unit lattices



8 Spherical model of units for power-of-2 cyclotomics

This section takes the smallest possible y: namely, $y = 1$. Then $S = \infty$, and the S-unit lattice is just the unit lattice. This section computes—assuming

m	n	$\mathrm{Reg}_K^+/(n/4)^{n/4}$	spherical model	actual length	ratio
8	4	0.881374	2.492901	2.492901	1.000000
16	8	0.610449	2.652102	3.766835	0.704066
32	16	0.480772	4.081293	5.673348	0.719380
64	32	0.384226	6.967780	8.189221	0.850848
128	64	0.393293	12.735518	11.719983	1.086650
256	128	0.286233	23.862591	16.663464	1.432031
512	256	0.200698	45.953088	23.631207	1.944593
1024	512	0.202244	90.089629	33.464774	2.692073
2048	1024	0.192272	178.014429	47.358628	3.758860
4096	2048	0.199056	353.577209	66.997907	5.277437

Table 8.3. Numerical examples of how inaccurate spherical models are for unit lattices for power-of-2 cyclotomic fields, assuming $h^+ = 1$. All entries after the first two columns are rounded to 6 digits after the decimal point. First column: conductor m of field $K = \mathbb{Q}(\zeta_m)$. Second column: $n = m/2$, the degree of K. Third column: the regulator Reg_K^+ of K^+ divided by $(n/4)^{n/4}$. Fourth column, "spherical model": minimum nonzero length in spherical model of the unit lattice of K. Fifth column, "actual length": length of a nonzero vector in the unit lattice of K, namely $\mathrm{Log}(1+\zeta_m+1/\zeta_m)$. Sixth column, "ratio": fourth column divided by fifth column. Compare Table 4.2.

Question:
Do any of the examples in [BL21] disprove any of the assumptions made in [PHS19]?

S-unit Attacks

- Take-away message: we are starting to understand these attacks?
  - Bernard and Roux-Langlois ([BR20]: https://eprint.iacr.org/2020/1081)
  - Bernard et al ([BLNR21]: https://eprint.iacr.org/2021/1384)