The 2nd Round

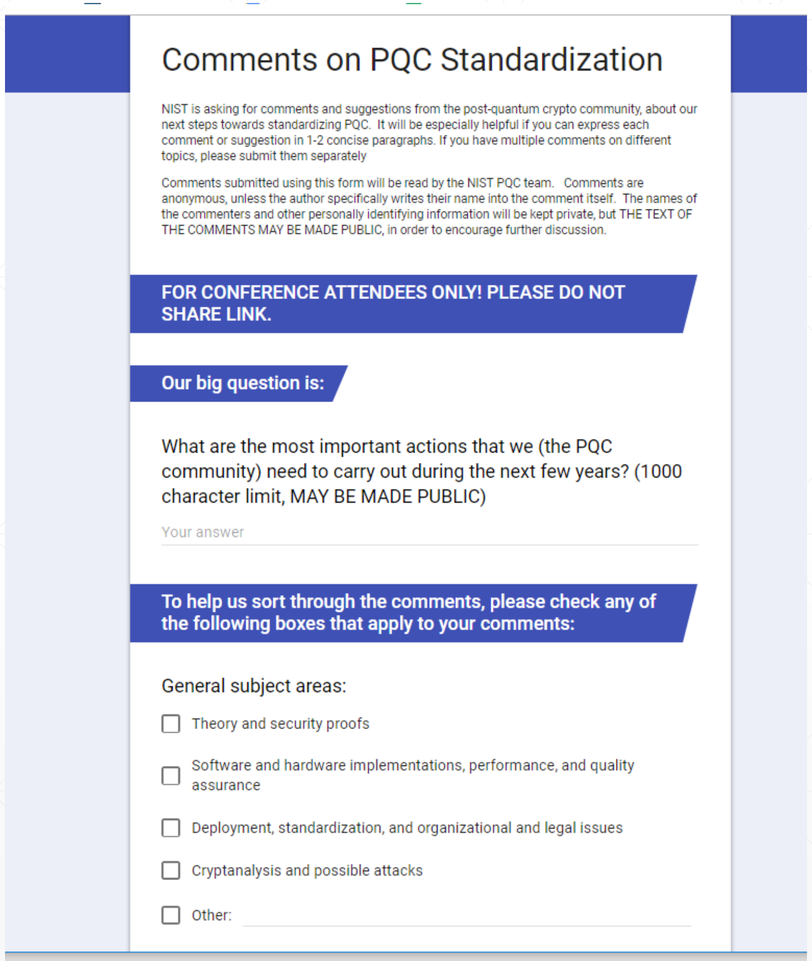
of the NIST PQC Standardization Process

Dustin Moody

National Institute of Standards and Technology
U.S. Department of Commerce

The 2nd NIST PQC Standardization Workshop

- Over 250 people registered
- (almost) All of the Round 2 teams will give an update
- 17 papers to be presented out of 43 submitted
- An Industry Panel later today
- Final session – Next Steps/Open Problems
  - Please answer the questions sent to you / scan the QR code



Comments on PQC Standardization

NIST is asking for comments and suggestions from the post-quantum crypto community, about our next steps towards standardizing PQC. It will be especially helpful if you can express each comment or suggestion in 1-2 concise paragraphs. If you have multiple comments on different topics, please submit them separately

Comments submitted using this form will be read by the NIST PQC team. Comments are anonymous, unless the author specifically writes their name into the comment itself. The names of the commenters and other personally identifying information will be kept private, but THE TEXT OF THE COMMENTS MAY BE MADE PUBLIC, in order to encourage further discussion.

FOR CONFERENCE ATTENDEES ONLY! PLEASE DO NOT SHARE LINK.

Our big question is:

What are the most important actions that we (the PQC community) need to carry out during the next few years? (1000 character limit, MAY BE MADE PUBLIC)

Your answer

To help us sort through the comments, please check any of the following boxes that apply to your comments:

General subject areas:

- [ ] Theory and security proofs
- [ ] Software and hardware implementations, performance, and quality assurance
- [ ] Deployment, standardization, and organizational and legal issues
- [ ] Cryptanalysis and possible attacks
- [ ] Other:

How we got here…

- NIST's public-key crypto standards
  - FIPS 186, The Digital Signature Standard
  - SP 800-56 A/B, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm/Integer Factorization Cryptography

- Quantum computers and Shor's Algorithm

Algorithms for Quantum Computation: Discrete Logarithms and Factoring

Peter W. Shor
AT&T Bell Labs
Room 2D-149
600 Mountain Ave.
Murray Hill, NJ 07974, USA

Abstract

A computer is generally considered to be a universal computational device; i.e., it is believed able to simulate any physical computational device with a cost in computation time of at most a polynomial factor. It is not clear whether this is still true when quantum mechanics is taken into consideration. Several researchers, starting with David Deutsch, have developed models for quantum mechanical computers and have investigated their computational properties. This paper gives Las Vegas algorithms for finding discrete logarithms and factoring integers on a quantum computer that take a number of steps which is polynomial in the input size, e.g., the number of digits of the integer to be factored. These two problems are generally considered hard on a classical computer and have been used as the basis of several proposed cryptosystems. (We thus give the first examples of quantum cryptanalysis.)

1 Introduction

Since the discovery of quantum mechanics, people have found the behavior of the laws of probability in quantum mechanics counterintuitive. Because of this behavior, quantum mechanical phenomena behave quite differently than the phenomena of classical physics that we are used to. Feynman seems to have been the first to ask what effect this has on computation [13, 14]. He gave arguments as [1, 2]. Although he did not ask whether quantum mechanics conferred extra power to computation, he did show that a Turing machine could be simulated by the reversible unitary evolution of a quantum process, which is a necessary prerequisite for quantum computation. Deutsch [9, 10] was the first to give an explicit model of quantum computation. He defined both quantum Turing machines and quantum circuits and investigated some of their properties.

The next part of this paper discusses how quantum computation relates to classical complexity classes. We will thus first give a brief intuitive discussion of complexity classes for those readers who do not have this background. There are generally two resources which limit the ability of computers to solve large problems: time and space (i.e., memory). The field of analysis of algorithms considers the asymptotic demands that algorithms make for these resources as a function of the problem size. Theoretical computer scientists generally classify algorithms as efficient when the number of steps of the algorithms grows as a polynomial in the size of the input. The class of problems which can be solved by efficient algorithms is known as P. This classification has several nice properties. For one thing, it does a reasonable job of reflecting the performance of algorithms in practice (although an algorithm whose running time is the tenth power of the input size, say, is not truly efficient). For another, this classification is nice theoretically, as different reasonable machine models produce the same class P. We will see this behavior reappear in quantum computation, where different models for

How we got here…

- 2006 – 1st PQCrypto conference in Leuven, Belgium
- 2009 – NIST PQC survey Quantum Resistant Public Key Cryptography: A Survey [Perlner, Cooper]
- 2012 – NIST begins PQC project
- Apr 2015 – NIST Workshop on Cybersecurity in a Post-Quantum World
- Aug 2015 – NSA announcement
- Feb 2016 – NIST Report on PQC (NISTIR 8105)
- Feb 2016 – NIST announcement of “competition-like process” at PQCrypto in Japan
- Dec 2016 – Final requirements and evaluation criteria published
- Nov 2017 – Deadline for Submissions
- Dec 2017 – Round 1 begins – 69 candidates accepted as “complete and proper”
- Apr 2018 – 1st NIST PQC Standardization Workshop
- Jan 2019 – Round 2 candidates announced
- Aug 2019 – 2nd NIST PQC Standardization Workshop

The “Competition”

- Scope:
  - Digital Signatures
    - EUF-CMA up to 2^{64} signature queries
  - Public-key Encryption / Key-Encapsulation Mechanisms (KEMs)
    - IND-CCA up to 2^{64} decryption/decapsulation queries
    - IND-CPA option
- Open and transparent process
- Unlike previous AES and SHA-3 competitions, there will not be a single “winner”

Evaluation Criteria

- Security – against both classical and quantum attacks

Level	Security Description
I	At least as hard to break as AES128 (exhaustive key search)
II	At least as hard to break as SHA256 (collision search)
III	At least as hard to break as AES192 (exhaustive key search)
IV	At least as hard to break as SHA384 (collision search)
V	At least as hard to break as AES256 (exhaustive key search)

  - NIST asked submitters to focus on levels 1,2, and 3. (Levels 4 and 5 are for very high security)
- Performance – measured on various classical platforms
- Other properties: Drop-in replacements, Perfect forward secrecy, Resistance to side-channel attacks, Simplicity and flexibility, Misuse resistance, etc.

The 1st Round Candidates

- 82 submissions received.
- 69 accepted as “complete and proper”   (5 withdrew)

	Signatures	KEM/Encryption	Overall
Lattice-based	5	21	26
Code-based	2	17	19
Multi-variate	7	2	9
Symmetric-based	3		3
Other	2	5	7
Total	**19**	**45**	**64**

The ~~1st Round~~ 2nd Round Candidates

	Signatures		KEM/Encryption		Overall	
Lattice-based	~~5~~	3	~~21~~	9	~~26~~	12
Code-based	~~2~~	0	~~17~~	7	~~19~~	7
Multi-variate	~~7~~	4	~~2~~	0	~~9~~	4
Symmetric-based	~~3~~	2			~~3~~	2
Other	~~2~~	0	~~5~~	1	~~7~~	1
Total	~~19~~	**9**	~~45~~	**17**	~~64~~	**26**

Overview of the 1st Round

- Began Dec 2017 – 1st Round Candidates published
- Resources:
  - Internal and external cryptanalysis
    - 21 of the 69 schemes had been broken/attacked by April
  - The 1st NIST PQC Standardization Workshop
  - Research publications
  - Performance benchmarks
  - Official comments
  - The pqc-forum mailing list

NIST’s Process

- Dec 2017 – Check submissions for completeness
- Jan to Sep 2018 – Detailed internal presentations on submissions
- Apr 2018 – 1st Workshop – submitter’s presentations
- Sep to Nov 2018 – Review and make preliminary decisions
  - Compare similar type schemes to each other
- Dec 2018 – Final decision and start report (NISTIR 8240)
  - Very hard decisions
  - NISTIR 8240 – Status Report on the 1st Round of the NIST PQC Standardization Process
    - Report focused on the reasons for moving on
- Announced 2nd Round candidates – Jan 30, 2019

Apples and Oranges

Encryption/KEMs						Signatures		
Crystals-Kyber	Lattice	MLWE	Big Quake	Codes	Goppa	CRYSTALS-Dilithium	Lattice	Fiat-Shamir
KINDI	Lattice	MLWE	Classic McEliece	Codes	Goppa	qTesla	Lattice	Fiat-Shamir
Saber	Lattice	MLWR	NTS-KEM	Codes	Goppa	Falcon	Lattice	Hash then sign
FrodoKEM	Lattice	LWE	BIKE	Codes	short Hamming	pqNTRUSign	Lattice	Hash then sign
Lotus	Lattice	LWE	HQC	Codes	short Hamming			
Lizard	Lattice	LWE/RLWE	LEDAkem	Codes	short Hamming	Gravity-SPHINCS	Symm	Hash
Emblem/R.emblem	Lattice	LWE/RLWE	LEDApkc	Codes	short Hamming	SPHINCS+	Symm	Hash
KCL	Lattice	LWE/RLWE/LWR	QC-MDPC KEM	Codes	short Hamming	Picnic	Symm	ZKP
Round 2	Lattice	LWR/RLWR	LAKE	Codes	low rank			
Hila5	Lattice	RLWE	LOCKER	Codes	low rank	GeMMS	MultVar	HFE
Ding's key exchange	Lattice	RLWE	Ouroboros-R	Codes	low rank	Gui	MultVar	HFE
LAC	Lattice	RLWE	RQC	Codes	low rank	HiMQ-3	MultVar	UOV
Lima	Lattice	RLWE				LUOV	MultVar	UOV
NewHope	Lattice	RLWE				Rainbow	MultVar	UOV
Three Bears	Lattice	IMLWE	SIKE	Isogeny	Isogeny	MQDSS	MultVar	Fiat-Shamir
Mersenne-756839	Lattice	ILWE						
Titanium	Lattice	MP-LWE						
Ramstake	Lattice	LWE like						
Odd Manhattan	Lattice	Generic						
NTRU Encrypt	Lattice	NTRU						
NTRU-HRSS-KEM	Lattice	NTRU						
NTRUprime	Lattice	NTRU						

Biting the Bullet

Crystals-Kyber	Lattice	MLWE
KINDI	Lattice	MLWE
Saber	Lattice	MLWR
FrodoKEM	Lattice	LWE
Lotus	Lattice	LWE
Lizard	Lattice	LWE/RLWE
Emblem/R.emblem	Lattice	LWE/RLWE
KCL	Lattice	LWE/RLWE/LWR
Round 2	Lattice	LWR/RLWR
Hila5	Lattice	RLWE
Ding's key exchange	Lattice	RLWE
LAC	Lattice	RLWE
Lima	Lattice	RLWE
NewHope	Lattice	RLWE
Three Bears	Lattice	IMLWE
Mersenne-756839	Lattice	ILWE
Titanium	Lattice	MP-LWE
Ramstake	Lattice	LWE like
Odd Manhattan	Lattice	Generic
NTRU Encrypt	Lattice	NTRU
NTRU-HRSS-KEM	Lattice	NTRU
NTRUprime	Lattice	NTRU

→

Crystals-Kyber	Lattice	MLWE
Saber	Lattice	MLWR
FrodoKEM	Lattice	LWE
Round 5	Lattice	LWR/RLWR
LAC	Lattice	RLWE
NewHope	Lattice	RLWE
Three Bears	Lattice	IMLWE
NTRU	Lattice	NTRU
NTRUprime	Lattice	NTRU

Big Quake	Codes	Goppa
Classic McEliece	Codes	Goppa
NTS-KEM	Codes	Goppa
BIKE	Codes	short Hamming
HQC	Codes	short Hamming
LEDAkem	Codes	short Hamming
LEDApkc	Codes	short Hamming
QC-MDPC KEM	Codes	short Hamming
LAKE	Codes	low rank
LOCKER	Codes	low rank
Ouroboros-R	Codes	low rank
RQC	Codes	low rank
SIKE	Isogeny	Isogeny

→

Classic McEliece	Codes	Goppa
NTS-KEM	Codes	Goppa
BIKE	Codes	short Hamming
HQC	Codes	short Hamming
LEDAcrypt	Codes	short Hamming
Rollo	Codes	low rank
RQC	Codes	low rank
SIKE	Isogeny	Isogeny

Signatures		
CRYSTALS-Dilithium	Lattice	Fiat-Shamir
qTesla	Lattice	Fiat-Shamir
Falcon	Lattice	Hash then sign
pqNTRUSign	Lattice	Hash then sign
Gravity-SPHINCS	Symm	Hash
SPHINCS+	Symm	Hash
Picnic	Symm	ZKP
GeMMS	MultVar	HFE
Gui	MultVar	HFE
HiMQ-3	MultVar	UOV
LUOV	MultVar	UOV
Rainbow	MultVar	UOV
MQDSS	MultVar	Fiat-Shamir

→

Signatures		
CRYSTALS-Dilithium	Lattice	Fiat-Shamir
qTesla	Lattice	Fiat-Shamir
Falcon	Lattice	Hash then sign
SPHINCS+	Symm	Hash
Picnic	Symm	ZKP
GeMMS	MultVar	HFE
LUOV	MultVar	UOV
Rainbow	MultVar	UOV
MQDSS	MultVar	Fiat-Shamir

The Second Round (and beyond)

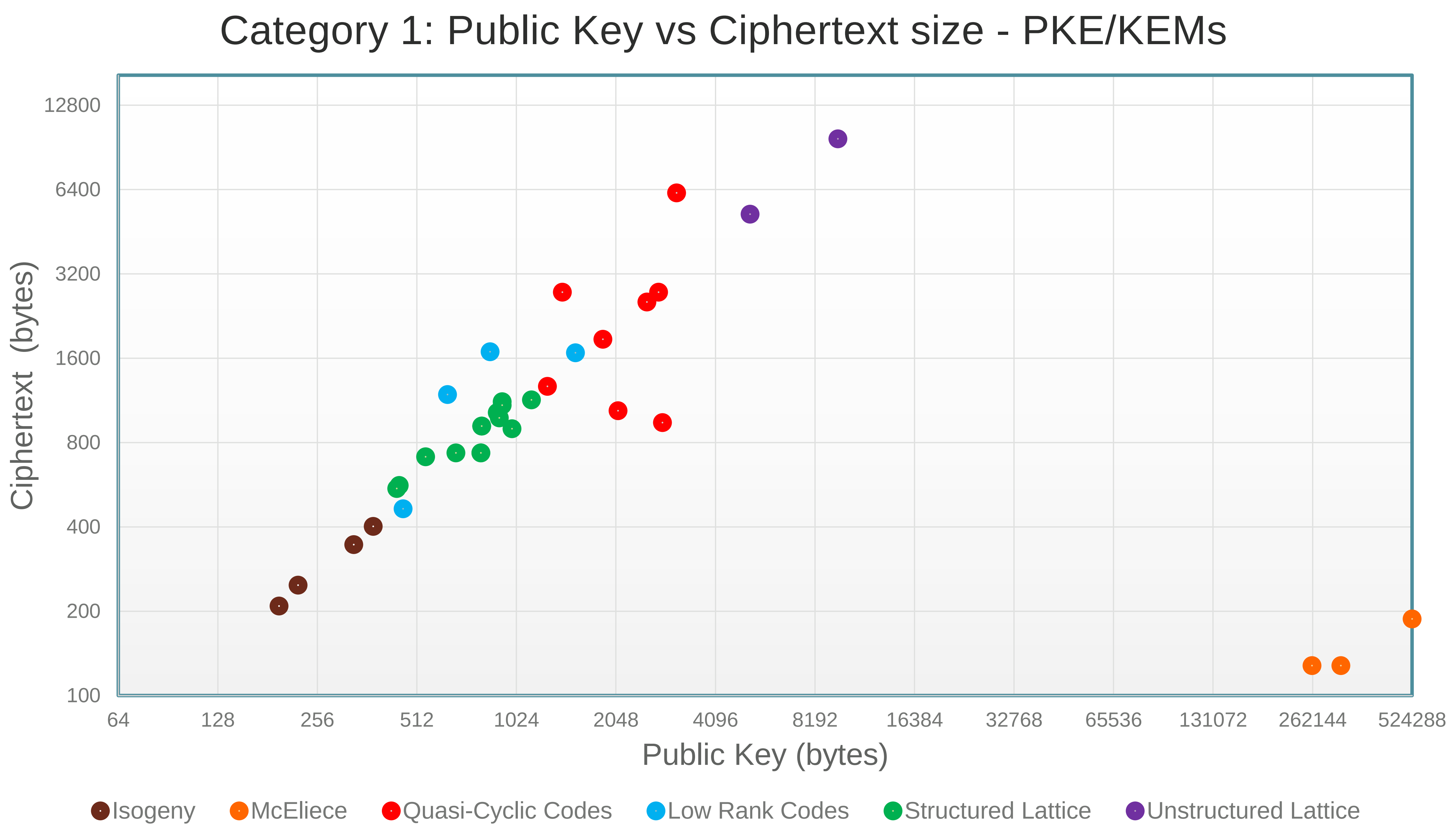
- NIST is still open to mergers
- Only need new IP statements if new team members have joined, or if IP status has changed
  - Later on in process, IP concerns may play a larger role in our decisions
- The 2nd Round will take 12-18 months, after which we expect to have a 3rd Round
- Overall timeline: we still expect draft standards around 2022ish
  - (but reserve the right to change this!)

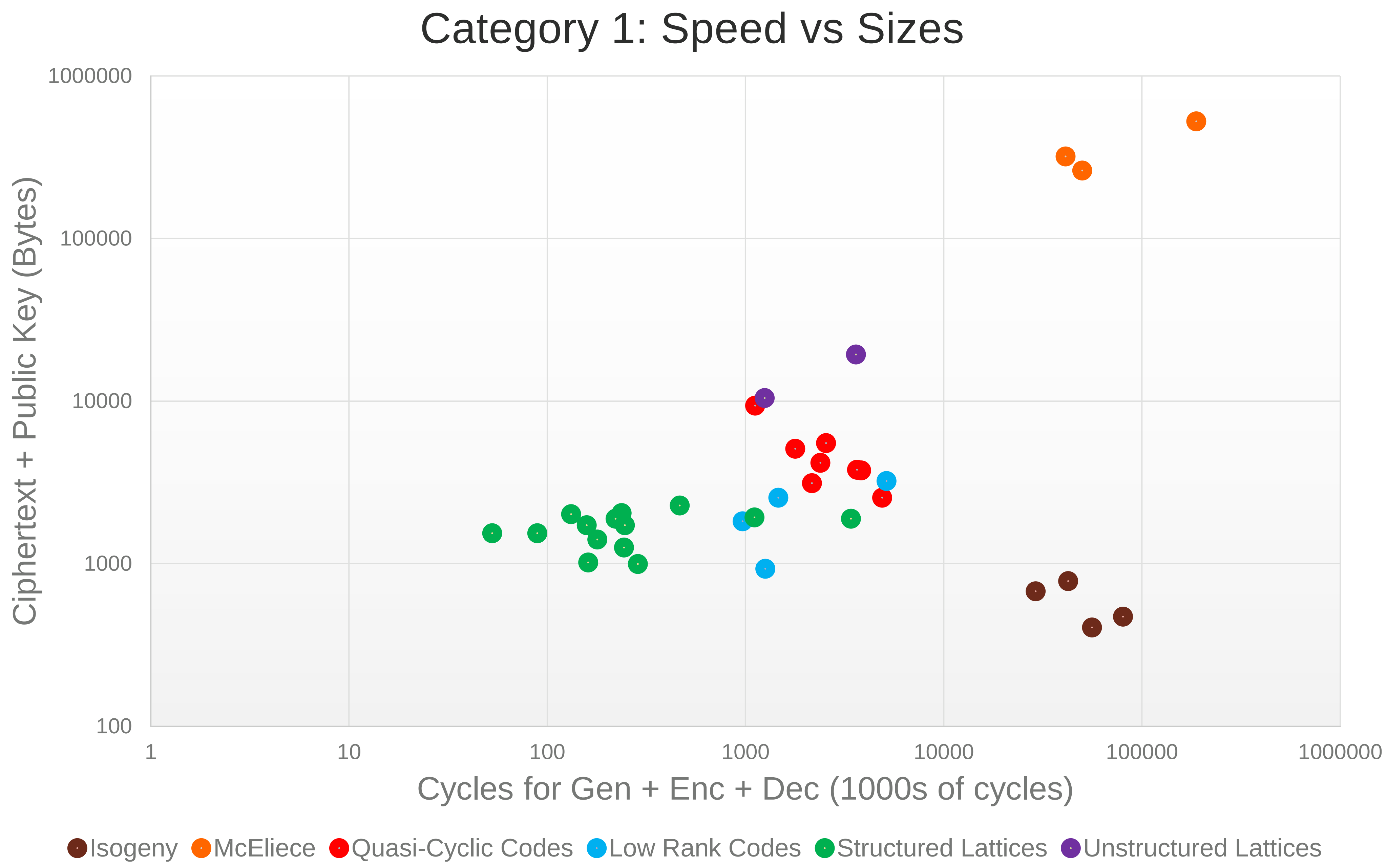
Performance

- We have internal numbers, based on implementations sent to us
  - We strongly prefer code that is constant time

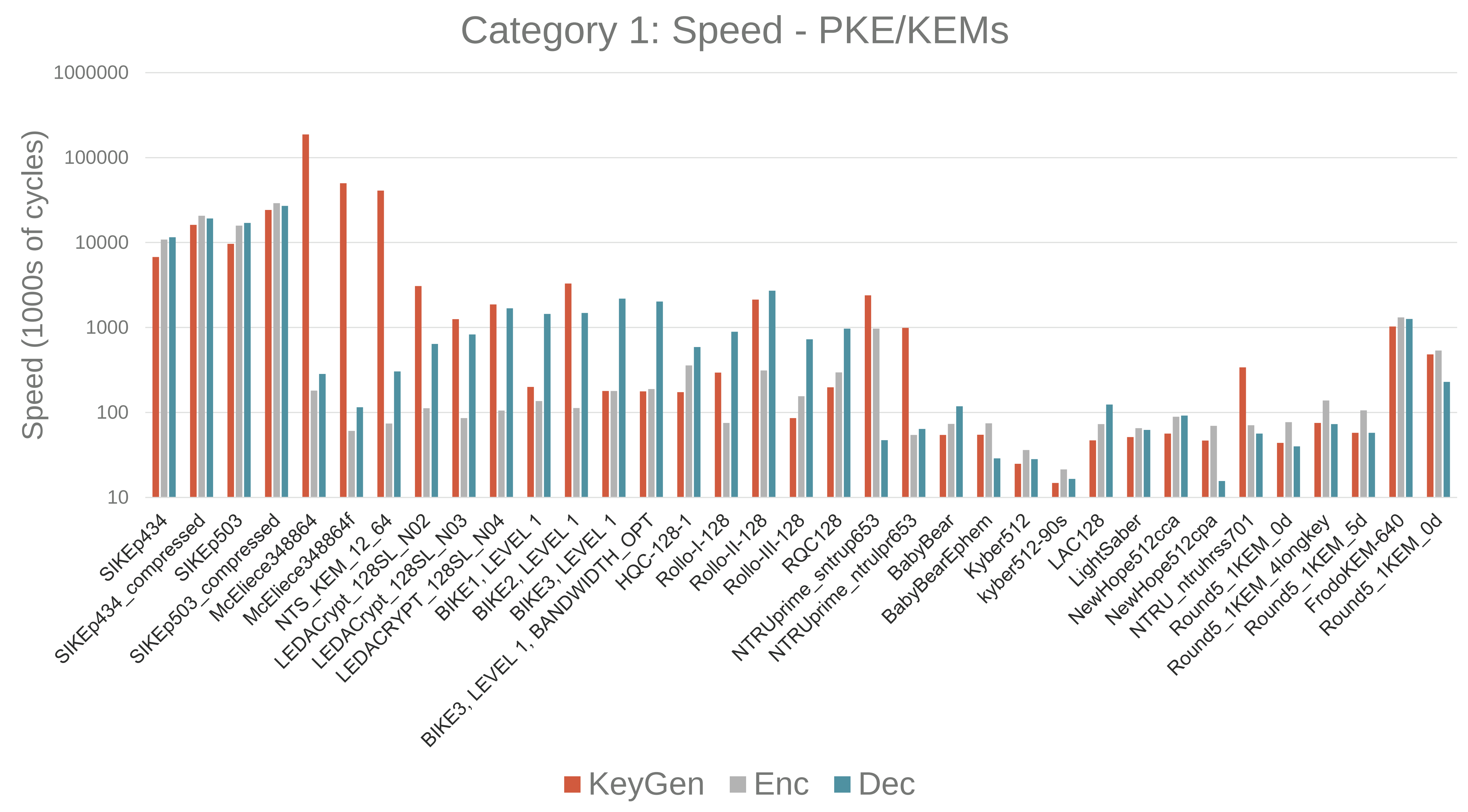
- Performance will play a larger role in the 2nd Round
  - We encourage benchmarking on a variety of platforms
  - We are looking for mature schemes – beyond just proof of concept

- Implementations can always be updated
  - We won’t change the implementations on our Round 2 webpage
  - Teams should feel free to advertise results on the pqc-forum, and on their own websites

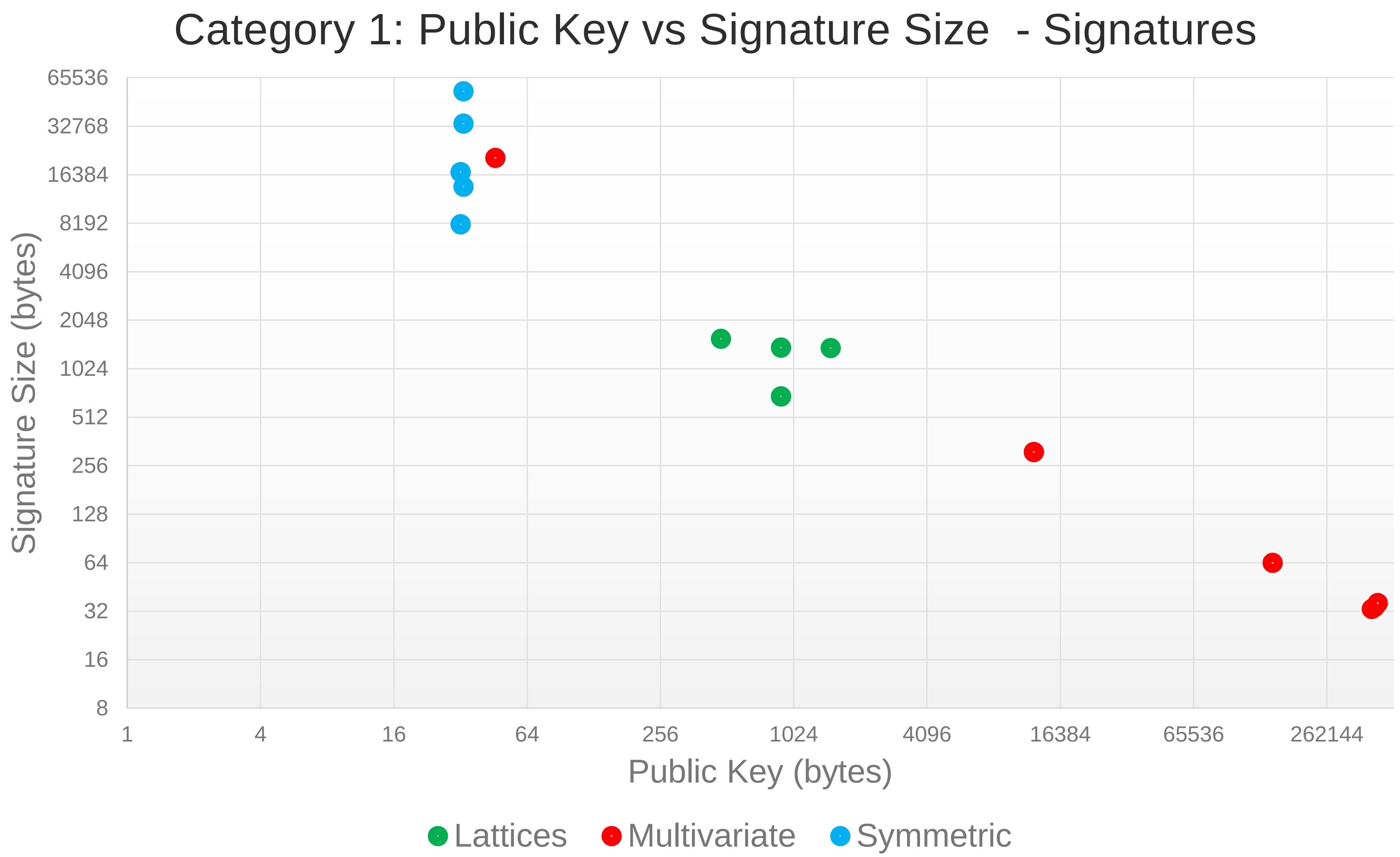

Category 1: Public Key vs Ciphertext size - PKE/KEMs
Ciphertext (bytes)
12800
6400
3200
1600
800
400
200
100
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
Public Key (bytes)
Isogeny
McEliece
Quasi-Cyclic Codes
Low Rank Codes
Structured Lattice
Unstructured Lattice



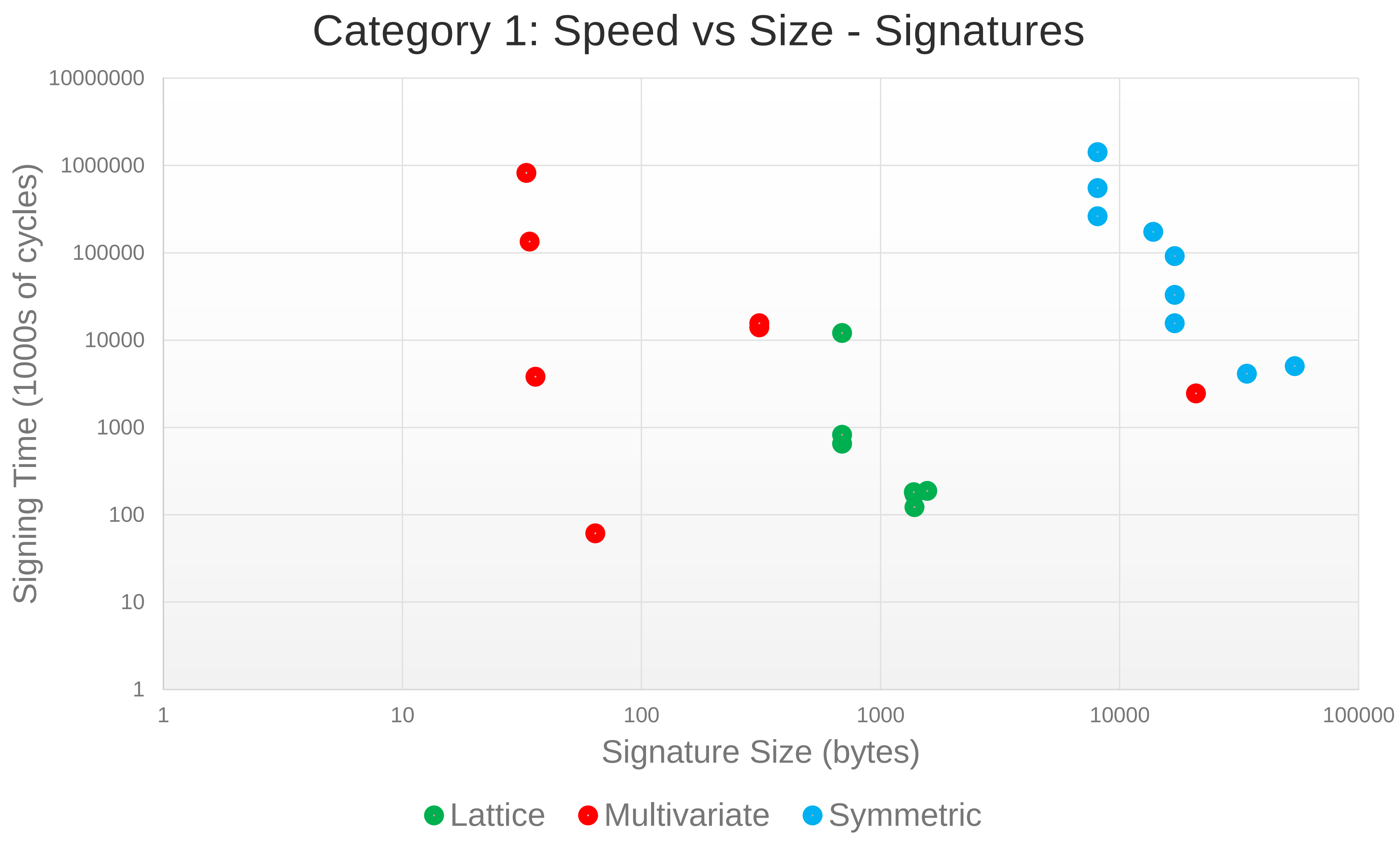
Category 1: Speed vs Sizes
1000000
100000
10000
1000
100
Ciphertext + Public Key (Bytes)
1
10
100
1000
10000
100000
1000000
Cycles for Gen + Enc + Dec (1000s of cycles)
Isogeny
McEliece
Quasi-Cyclic Codes
Low Rank Codes
Structured Lattices
Unstructured Lattices

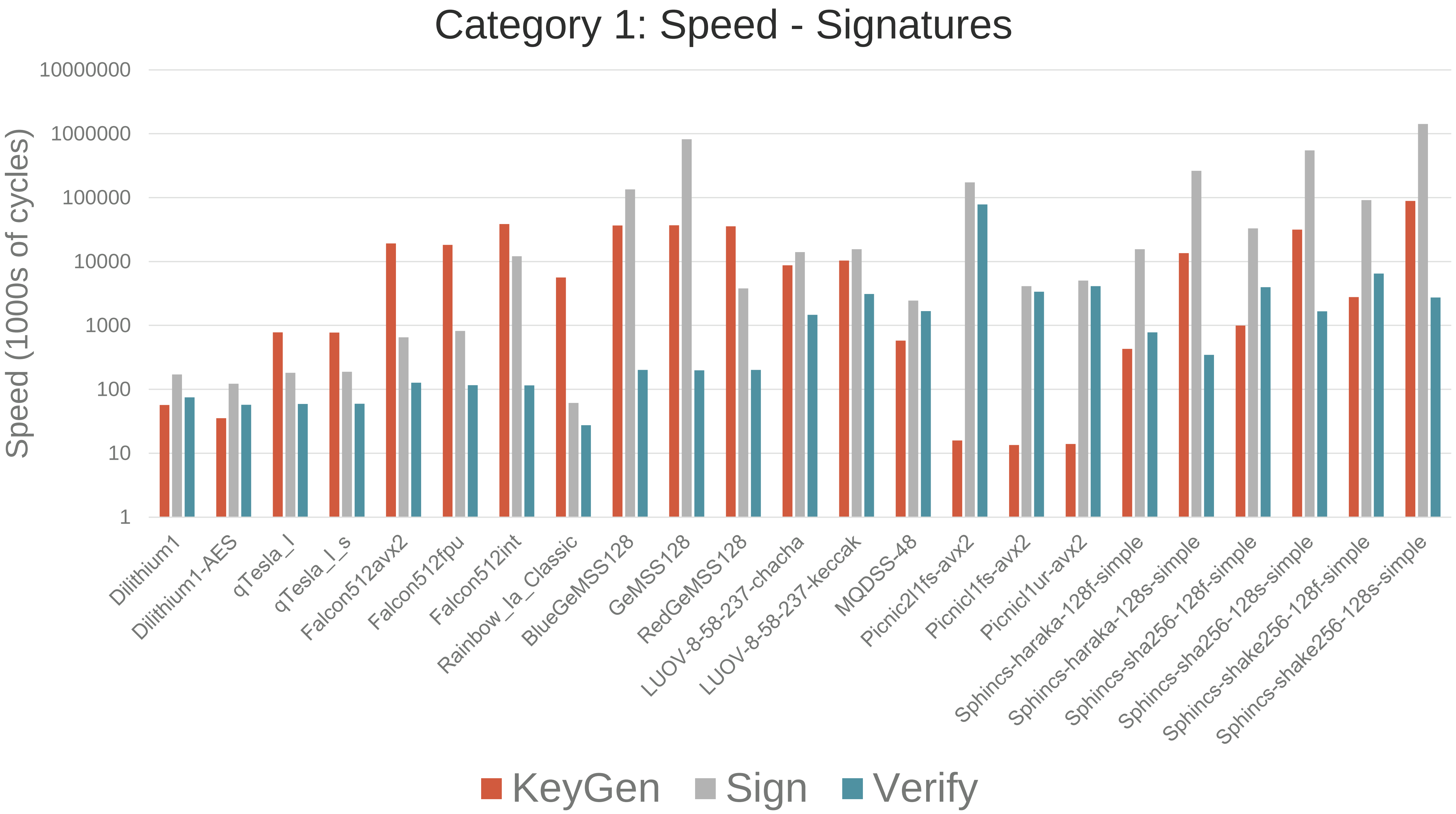


Category 1: Speed - PKE/KEMs
Speed (1000s of cycles)
1000000
100000
10000
1000
100
10
SIKEp434
SIKEp434_compressed
SIKEp503
SIKEp503_compressed
McEliece348864
McEliece348864f
NTS_KEM_12_64
LEDACrypt_128SL_N02
LEDACrypt_128SL_N03
LEDACRYPT_128SL_N04
BIKE1, LEVEL 1
BIKE2, LEVEL 1
BIKE3, LEVEL 1
BIKE3, LEVEL 1, BANDWIDTH_OPT
HQC-128-1
Rollo-I-128
Rollo-II-128
Rollo-III-128
RQC128
NTRUprime_sntrup653
NTRUprime_ntrulpr653
BabyBear
BabyBearEphem
Kyber512
kyber512-90s
LAC128
LightSaber
NewHope512cca
NewHope512cpa
NTRU_ntruhrss701
Round5_1KEM_0d
Round5_1KEM_4longkey
Round5_1KEM_5d
FrodoKEM-640
Round5_1KEM_0d
KeyGen
Enc
Dec



Category 1: Public Key vs Signature Size - Signatures
Signature Size (bytes)
65536
32768
16384
8192
4096
2048
1024
512
256
128
64
32
16
8
1
4
16
64
256
1024
4096
16384
65536
262144
Public Key (bytes)
Lattices
Multivariate
Symmetric


Category 1: Speed vs Size - Signatures
Signing Time (1000s of cycles)
10000000
1000000
100000
10000
1000
100
10
1
1
10
100
1000
10000
100000
Signature Size (bytes)
Lattice
Multivariate
Symmetric


Category 1: Speed - Signatures
Speed (1000s of cycles)
10000000
1000000
100000
10000
1000
100
10
1
Dilithium1
Dilithium1-AES
qTesla_I
qTesla_I_s
Falcon512avx2
Falcon512fpu
Falcon512int
Rainbow_Ia_Classic
BlueGeMSS128
GeMSS128
RedGeMSS128
LUOV-8-58-237-chacha
LUOV-8-58-237-keccak
MQDSS-48
Picnic2l1fs-avx2
Picnicl1fs-avx2
Picnicl1ur-avx2
Sphincs-haraka-128f-simple
Sphincs-haraka-128s-simple
Sphincs-sha256-128f-simple
Sphincs-sha256-128s-simple
Sphincs-shake256-128f-simple
Sphincs-shake256-128s-simple
KeyGen
Sign
Verify

Stateful Hash-based signatures

- NIST plans to approve stateful hash-based signatures
  - 1) XMSS, specified in RFC 8391
  - 2) LMS, specified in RFC 8554
    - Will include their multi-tree variants, XMSS^MT and HSS
- In Feb 2019, NIST issued a request for public input on how to mitigate the potential misuse of stateful HBS schemes.
  - See comments received here
- Will recommend HBS schemes limited to scenarios in which a digital signature scheme needs to be deployed soon, but where risks of accidental one-time key reuse can be minimized
- NIST expects to have a draft Special Publication (SP) published by the end of 2019

What NIST wants

- Performance (hardware+software) will play more of a role
  - More benchmarks
  - For hardware, NIST asks to focus on Cortex M4 (with all options) and Artix-7
    - pqc-hardware-forum
  - How do schemes perform on constrained devices?
  - Side-channel analysis (concrete attacks, protection, etc…)
- Continued research and analysis on **ALL** of the 2nd round candidates
- See how submissions fit into applications/procotols. Any constraints?

Summary

- Round 2 is ongoing….
  - 26 candidate algorithms
    (17 encryption/KEM, 9 signatures)
- We will continue to work in an open and transparent manner with the crypto community for PQC standards
- Check out: www.nist.gov/pqcrypto
  - Sign up for the pqc-forum
- Talk to us: pqc-comments@nist.gov