- PERK -

N. Aaraj, S. Bettaieb, L. Bidoux, A. Budroni, V. Dyseryn, A. Esser, T. Feneuil, P. Gaborit
M. Kulkarni, V. Mateu, M. Palumbi, L. Perin, M. Rivain, J.P. Tillich, K. Xagawa

NIST Sixth PQC Standardization Conference (09/25)

Overview

PERK is a digital signature scheme named after the **PER**muted **K**ernel problem

- Fiat-Shamir (FS) based signature along with a Zero-Knowledge Proof of Knowledge (PoK)
- PoK built using the Multi-Party Computation in the Head (MPCitH) paradigm
- PoK relies on the hardness of the Permuted Kernel Problem (PKP)

https://pqc-perk.org

Agenda

1. Round 2 Updates
2. Permuted Kernel Problem
3. Scheme Overview
4. Sizes & Performances
5. Advantages & Limitations

Round 2 Updates

Round 2 Updates

New results since Round 1

- New modeling for PKP [BBGK24]
- New MPCitH frameworks - **TCitH** [FR25] & **VOLEitH** [BBD+23]

Round 2 Updates

New results since Round 1

- New modeling for PKP [BBGK24]
- New MPCitH frameworks - **TCitH** [FR25] & **VOLEitH** [BBD+23]

Modifications for Round 2

- v2.0.0 - No modification
- v2.1.0 - Design update using the new modeling along with new MPCitH frameworks
  Implementation update

Round 2 Updates

PERK Instance	Modeling	Proof System	Size (pk + sig.)
Round 1	Shared Permutation	MPCitH	5.8 - 8.1 kB
Round 2	**Permutation Matrix & Elementary Vectors**	**VOLEitH**	**3.5 - 4.4 kB**

Table 1: Modifications for PERK (sizes are given for NIST-1 security level)

Permuted Kernel Problem

Permuted Kernel Problem

Permuted Kernel Problem

Input

- Secret permutation $\pi \in \mathcal{S}_n$
- Public matrix $\mathbf{H} \in \mathbb{F}_q^{m \times n}$ and vector $\mathbf{x} \in \mathbb{F}_q^n$ such that $\mathbf{H}(\pi[\mathbf{x}]) = \mathbf{0}$

Goal

- Find $\tilde{\pi} \in \mathcal{S}_n$ such that $\mathbf{H}(\tilde{\pi}[\mathbf{x}]) = \mathbf{0}$

PERK & PKP

PERK Security

- PERK security **no longer** relies on a **relaxed** PKP variant (r-IPKP)
- PERK now relies on standard PKP but defined over $\mathbb{F}_q$ an extension field of $\mathbb{F}_2$

PERK & PKP

PERK Security

- PERK security **no longer** relies on a **relaxed** PKP variant (r-IPKP)
- PERK now relies on standard PKP but defined over $\mathbb{F}_q$ an extension field of $\mathbb{F}_2$

PKP Assumption

- No known attacks exploiting the structure of the extension field $\mathbb{F}_q$

PERK & PKP

PERK Security

- PERK security **no longer** relies on a **relaxed** PKP variant (r-IPKP)
- PERK now relies on standard PKP but defined over $\mathbb{F}_q$ an extension field of $\mathbb{F}_2$

PKP Assumption

- No known attacks exploiting the structure of the extension field $\mathbb{F}_q$
- New result on PKP cryptanalysis to appear soon [BDP25]

  $\Rightarrow$ ***No impact*** on PERK parameters

Scheme Overview

Modeling

PERK v2.1 relies on a modeling based on **permutation matrix** & **elementary vectors** [BBGK24]

Modeling

PERK v2.1 relies on a modeling based on **permutation matrix** & **elementary vectors** [BBGK24]

Permutation matrix

- Let $\mathbf{P}$ be the permutation matrix representation of the secret permutation π
- $\mathbf{P}$ is a permutation if (1) Each of its rows is an elementary vector
  (2) For each of its columns, the sum of all its coefficients is equal to 1
- $\mathbf{P}$ is a solution to a PKP instance $(\mathbf{H}, \mathbf{x})$ if $\mathbf{HPx} = \mathbf{0}$

Modeling

Elementary Vectors (EV)

- Vectors of $\mathbb{F}_2^n$ with one coordinate equal to 1 and the remaining ones equal to 0

$$\mathbf{v} = (0, \cdots, 0, 1, 0, \cdots, 0)$$

- EV of size n can be computed from tensor products of EV of smaller sizes

Modeling

Elementary Vectors (EV)

- Vectors of $\mathbb{F}_2^n$ with one coordinate equal to 1 and the remaining ones equal to 0

$$\mathbf{v} = (0, \cdots, 0, 1, 0, \cdots, 0)$$

- EV of size n can be computed from tensor products of EV of smaller sizes

- [BBGK24] suggests EV of size 2 as input $\rightarrow$ modeling degree $d = \lceil \log_2(n) \rceil$
- PERK uses EV of size 2 or 4 as input $\rightarrow$ modeling degree $d = \lceil \log_4(n) \rceil$

Modeling

Elementary Vectors (EV)

- Vectors of $\mathbb{F}_2^n$ with one coordinate equal to 1 and the remaining ones equal to 0

$$\mathbf{v} = (0, \cdots, 0, 1, 0, \cdots, 0)$$

- EV of size n can be computed from tensor products of EV of smaller sizes

- [BBGK24] suggests EV of size 2 as input $\rightarrow$ modeling degree $d = \lceil \log_2(n) \rceil$
- PERK uses EV of size 2 or 4 as input $\rightarrow$ modeling degree $d = \lceil \log_4(n) \rceil$

$\mathbf{v} = (v_1, v_2, v_3, v_4)$ is an EV $\Leftrightarrow v_1 \oplus v_2 \oplus v_3 \oplus v_4 = 1$ and $v_1 \cdot v_2 = 0$ and $v_3 \cdot v_4 = 0$

Modeling

Protocol Overview

Public Input

- An instance $(\mathbf{H}, \mathbf{x})$ of the PKP problem

Private Input

- Representation of π using n sets of EV $(\mathbf{v}_1^{(i)}, \cdots, \mathbf{v}_d^{(i)})$ for all $i \in [n]$

Protocol

1. Verify that $\mathbf{v}_k^{(i)}$ is an EV for all $i \in [n]$ and $k \in [d]$
2. Compute $\mathbf{P} = (p_{ij})_{i \in [n], j \in [n]}$ where $(p_{i1}, \cdots, p_{in}) = \bigotimes_{k \in [d]} \mathbf{v}_k^{(i)}$ for all $i \in [n]$
3. Verify that $\sum_{i \in [n]} p_{ij} = 1$ for all $j \in [n]$
4. Verify that $\mathbf{HPx} = \mathbf{0}$

Proof System

MPCitH Frameworks

- Two recent improvements to the MPCitH paradigm - **TCitH** [FR25] & **VOLEitH** [BBD+23]
- TCitH and VOLEitH can be described using the PIOP formalism [Fen24]

Proof System

MPCitH Frameworks

- Two recent improvements to the MPCitH paradigm - **TCitH** [FR25] & **VOLEitH** [BBD+23]
- TCitH and VOLEitH can be described using the PIOP formalism [Fen24]

TCitH

- **5**-round protocol
- Computation over a small field
- Several protocol repetitions
- *Arguably* simpler

VOLEitH

- **7**-round protocol
- Computation over a large field
- One protocol execution
- Smaller signatures

Proof System

PERK & TCitH vs VOLEitH

- VOLEitH lead to smaller sizes (in particular for modeling with degree $d > 2$)
- PERK modeling features a degree $d \in [3, 4]$ depending on parameter sets

Proof System

PERK & TCitH vs VOLEitH

- VOLEitH lead to smaller sizes (in particular for modeling with degree $d > 2$)
- PERK modeling features a degree $d \in [3, 4]$ depending on parameter sets

PERK Instantiation

- PERK is instantiated with the **VOLEitH** framework
- PERK uses the *one tree* optimization for GGM trees [BBM+24]

Sizes & Performances

Implementation

Implementation Updates

- Implementation update matching PERK design update
- Several variants depending on symmetric primitives (AES/Rijndael or SHA3 for PRG & cmt)

Implementation

Implementation Updates

- Implementation update matching PERK design update
- Several variants depending on symmetric primitives (AES/Rijndael or SHA3 for PRG & cmt)

Benchmark & Ongoing Work

- Numbers reported for the fastest variant of the optimized implementation (avx2 & aes-ni)
- Ongoing work targeting additional performance improvements

Sizes & Performances

PERK-1 Instance		$\|sk\|$	$\|pk\|$	$\|sig.\|$	Keygen	Sign	Verify
Round 1	Short	32 B	241 B	5.8 kB	92 K	35 M	25 M
Round 2 (v2.1.0)	Short	32 B	104 B	3.5 kB	34 K	16 M	13 M
Round 1	Fast	32 B	241 B	8.1 kB	89 K	7.0 M	4.9 M
Round 2 (v2.1.0)	Fast	32 B	104 B	4.4 kB	33 K	5.0 M	3.4 M

Table 2: Sizes and performances (CPU cycles) of PERK for NIST-1 security level

Sizes & Performances

PERK-5 Instance		$\lvert sk \rvert$	$\lvert pk \rvert$	$\lvert sig. \rvert$	Keygen	Sign	Verify
Round 1	Short	64 B	507 B	23.0 kB	333 K	168 M	131 M
Round 2 (v2.1.0)	Short	64 B	195 B	14.9 kB	102 K	191 M	172 M
Round 1	Fast	64 B	507 B	31.7 kB	324 K	33 M	26 M
Round 2 (v2.1.0)	Fast	64 B	195 B	18.3 kB	100 K	53 M	37 M

Table 3: Sizes and performances (CPU cycles) of PERK for NIST-5 security level

Comparison to other schemes

- Stay tuned till the end of the session -

Overview of MPCitH based Signatures using the ***PQ-SORT*** *benchmarking tool*

Advantages & Limitations

Advantages & Limitations

Advantages

- *Security* - Well established security assumption introduced ~35 years ago
  PERK no longer relies on a variant of PKP

Advantages & Limitations

Advantages

- *Security* - Well established security assumption introduced ~35 years ago
  PERK no longer relies on a variant of PKP
- *Parameters* - Adaptive and easily tunable parameters & Resilience against attacks

Advantages & Limitations

Advantages

- *Security* - Well established security assumption introduced ~35 years ago
  PERK no longer relies on a variant of PKP
- *Parameters* - Adaptive and easily tunable parameters & Resilience against attacks
- *Size* - Small public keys & Competitive signature size
  $|\text{pk+ sig.}| \Rightarrow$ **3.5 kB** for PERK, **3.7 kB** for ML-DSA, **7.8 kB** for SLH-DSA (for NIST-1 level)

Advantages & Limitations

Advantages

- *Security* - Well established security assumption introduced ~35 years ago
  PERK no longer relies on a variant of PKP
- *Parameters* - Adaptive and easily tunable parameters & Resilience against attacks
- *Size* - Small public keys & Competitive signature size
  $|\text{pk+ sig.}| \Rightarrow$ **3.5 kB** for PERK, **3.7 kB** for ML-DSA, **7.8 kB** for SLH-DSA (for NIST-1 level)

Limitations

- *Size* - Quadratic growth of signature sizes with respect to security level

Advantages & Limitations

Advantages

- *Security* - Well established security assumption introduced ~35 years ago
  PERK no longer relies on a variant of PKP
- *Parameters* - Adaptive and easily tunable parameters & Resilience against attacks
- *Size* - Small public keys & Competitive signature size
  $|\text{pk+ sig.}| \Rightarrow$ **3.5 kB** for PERK, **3.7 kB** for ML-DSA, **7.8 kB** for SLH-DSA (for NIST-1 level)

Limitations

- *Size* - Quadratic growth of signature sizes with respect to security level
- *Performances* - Slower than lattice-based signature schemes
  But competitive with many other post-quantum signatures

Thank you for your attention.

References I

[BBD+23] Carsten Baum, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Emmanuela Orsini, Lawrence Roy, and Peter Scholl.
Publicly verifiable zero-knowledge and post-quantum signatures from VOLE-in-the-head.
In Helena Handschuh and Anna Lysyanskaya, editors, CRYPTO 2023, Part V, volume 14085 of LNCS, pages 581–615. Springer, Cham, August 2023.

[BBGK24] Slim Bettaieb, Loïc Bidoux, Philippe Gaborit, and Mukul Kulkarni.
Modelings for generic PoK and Applications: Shorter SD and PKP based Signatures.
Cryptology ePrint Archive, 2024.

[BBM+24] Carsten Baum, Ward Beullens, Shibam Mukherjee, Emmanuela Orsini, Sebastian Ramacher, Christian Rechberger, Lawrence Roy, and Peter Scholl.
One tree to rule them all: Optimizing ggm trees and owfs for post-quantum signatures.
In International Conference on the Theory and Application of Cryptology and Information Security, pages 463–493. Springer, 2024.

[BDP25] Alessandro Budroni, Marco Defranceschi, and Federico Pintore.
Refined Algorithms for Solving the Permuted Kernel Problem.
In Preparation, 2025.

References II

[Fen24] Thibauld Feneuil.
The polynomial-iop vision of the latest mpcith framework for signature schemes.
PQ Algebraic Cryptography Workshop, 2024.

[FR25] Thibauld Feneuil and Matthieu Rivain.
Threshold computation in the head: Improved framework for post-quantum signatures and zero-knowledge arguments.
Journal of Cryptology, 38(3):28, 2025.

- Overview of MPCitH based Signatures -

L. Bidoux

NIST Sixth PQC Standardization Conference (09/25)

PQ-SORT: Post Quantum Signatures On-Ramp Tests

O. Adjonyo, A. Alblooshi, I. Canales-Martinez, J. Chavez-Saab, G. Dione & F. Rodriguez Henriquez

PQ-SORT: Post Quantum Signatures On-Ramp Tests

O. Adjonyo, A. Alblooshi, I. Canales-Martinez, J. Chavez-Saab, G. Dione & F. Rodriguez Henriquez

Benchmarking Tool

- *Comparison* - All round 2 candidates within NIST Standardization of Additional Signatures
- *Visualization* - Interactive tool providing scatter plots, tables, graphs & Filtering capabilities

PQ-SORT: Post Quantum Signatures On-Ramp Tests

O. Adjonyo, A. Alblooshi, I. Canales-Martinez, J. Chavez-Saab, G. Dione & F. Rodriguez Henriquez

Benchmarking Tool

- *Comparison* - All round 2 candidates within NIST Standardization of Additional Signatures
- *Visualization* - Interactive tool providing scatter plots, tables, graphs & Filtering capabilities

Methodology

- *Fairness* - All algorithms are tested under the same controlled environment
- *Transparency* - Detailled methodology available on the project website
- *Platform* - Intel CPU (i7 - 13 700K @ 3.4 GHz), Ubuntu 24.04.1 & gcc version 13.3.0

PQ-SORT: Post Quantum Signatures On-Ramp Tests

O. Adjonyo, A. Alblooshi, I. Canales-Martinez, J. Chavez-Saab, G. Dione & F. Rodriguez Henriquez

Benchmarking Tool

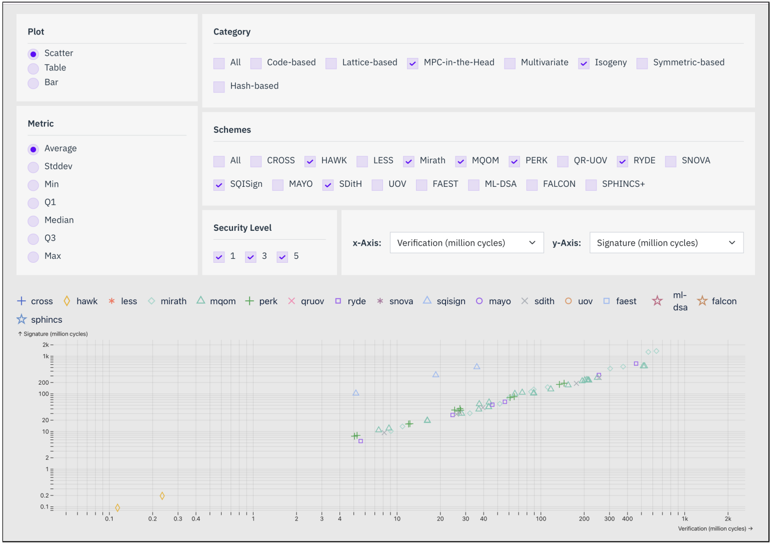
- *Comparison* - All round 2 candidates within NIST Standardization of Additional Signatures
- *Visualization* - Interactive tool providing scatter plots, tables, graphs & Filtering capabilities

Methodology

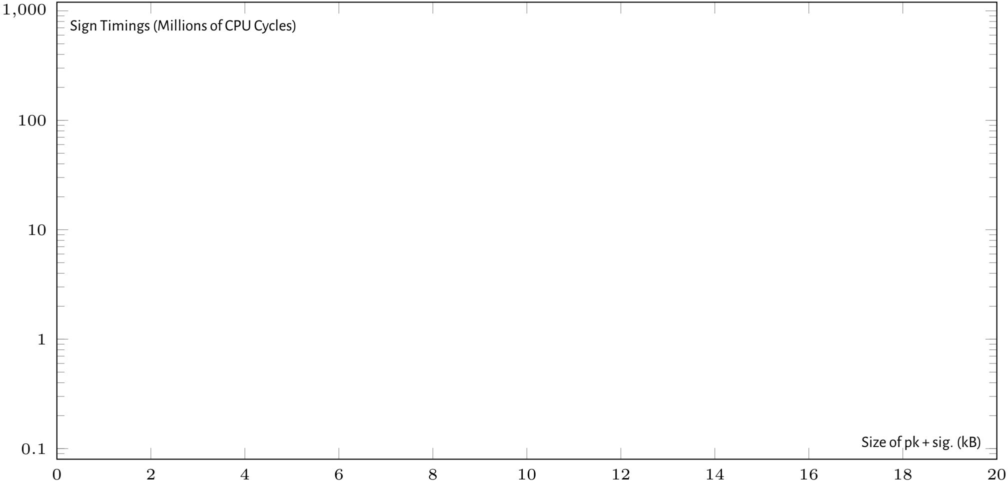
- *Fairness* - All algorithms are tested under the same controlled environment
- *Transparency* - Detailled methodology available on the project website
- *Platform* - Intel CPU (i7 - 13 700K @ 3.4 GHz), Ubuntu 24.04.1 & gcc version 13.3.0

- **pqsort.tii.ae** -

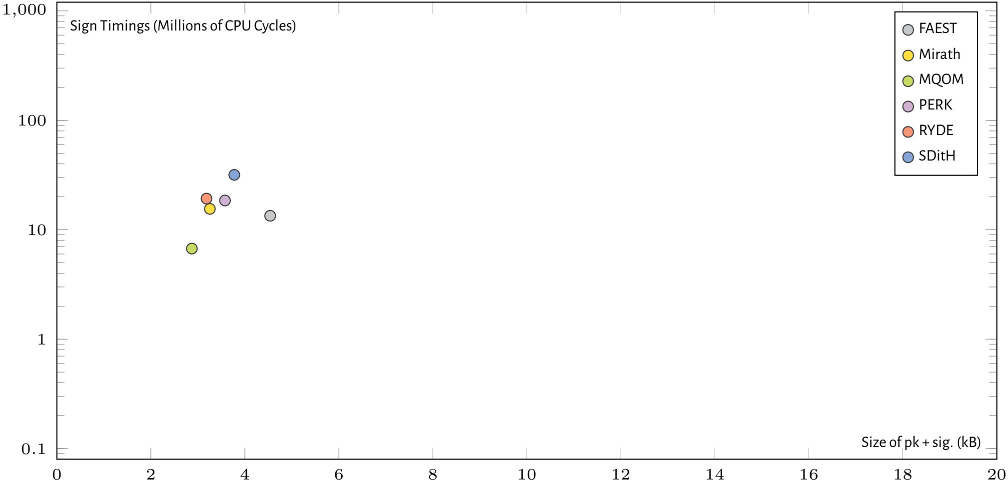
PQ-SORT: Post Quantum Signatures On-Ramp Tests



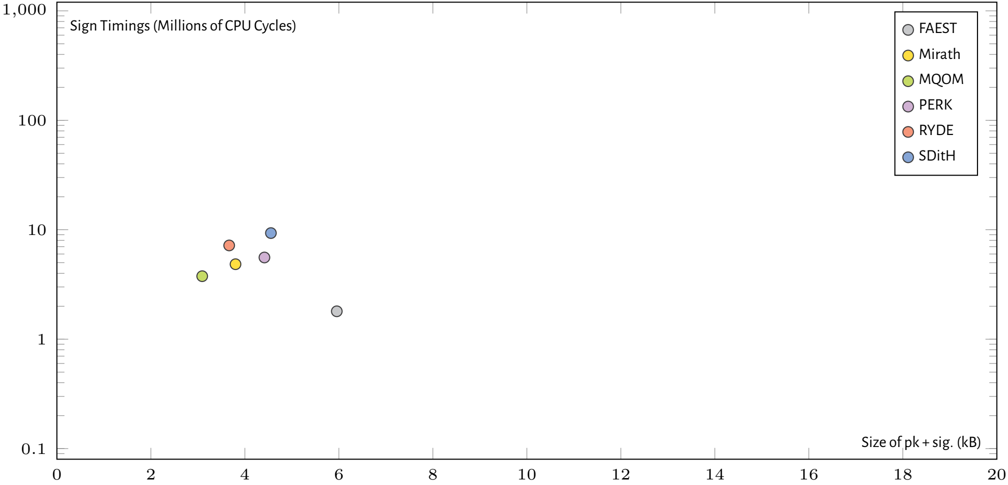
Comparaison between MPCitH and other Signatures



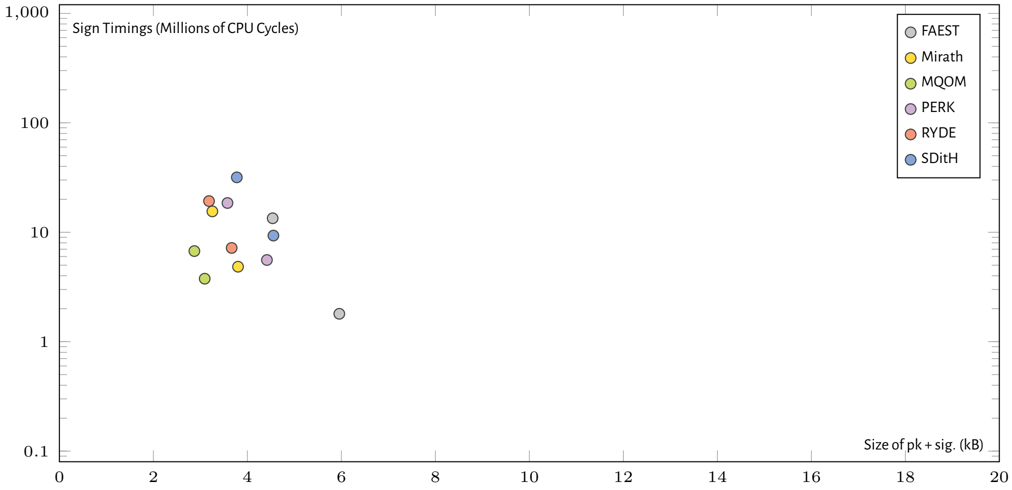
Comparaison between MPCitH and other Signatures



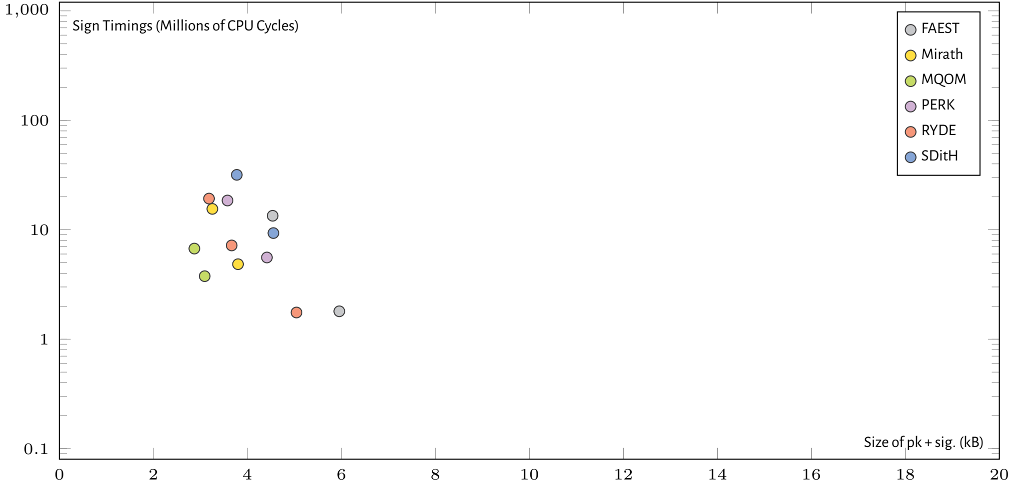
Comparaison between MPCitH and other Signatures



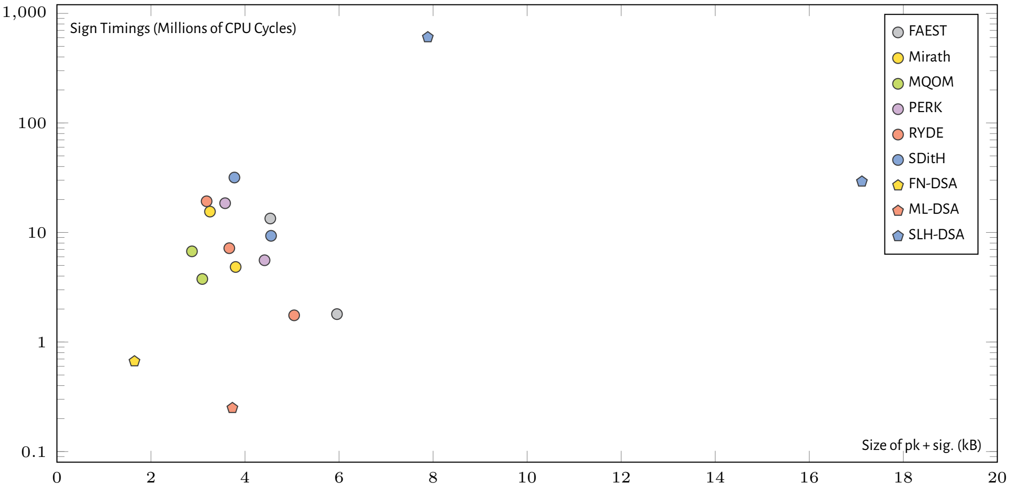
Comparaison between MPCitH and other Signatures



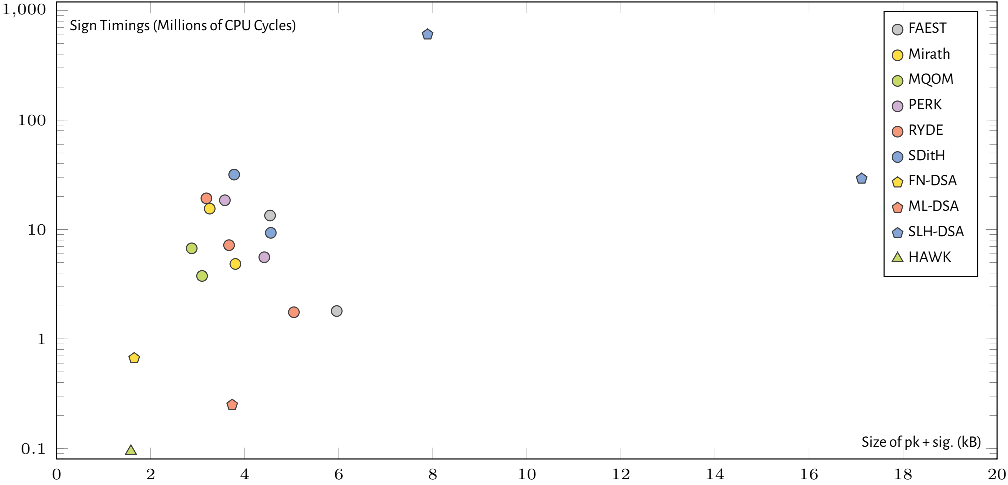
Comparaison between MPCitH and other Signatures



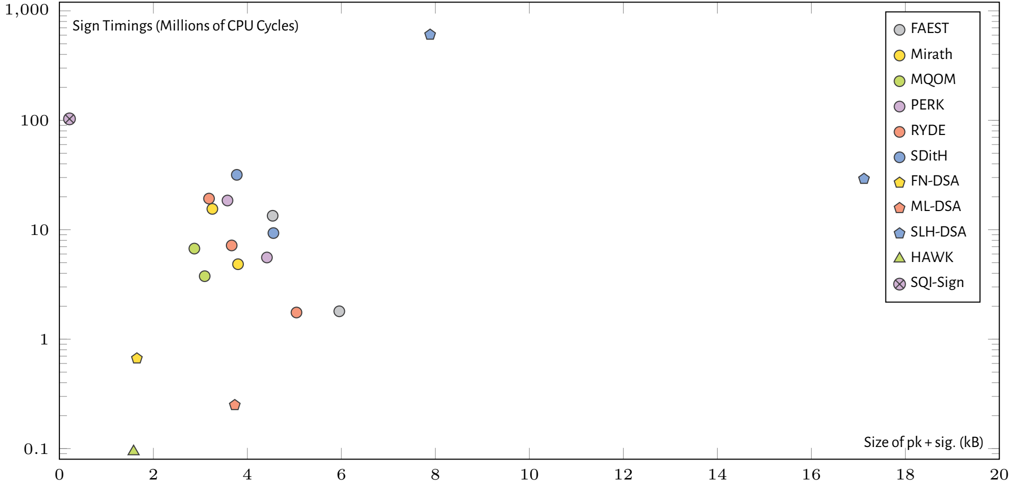
Comparaison between MPCitH and other Signatures



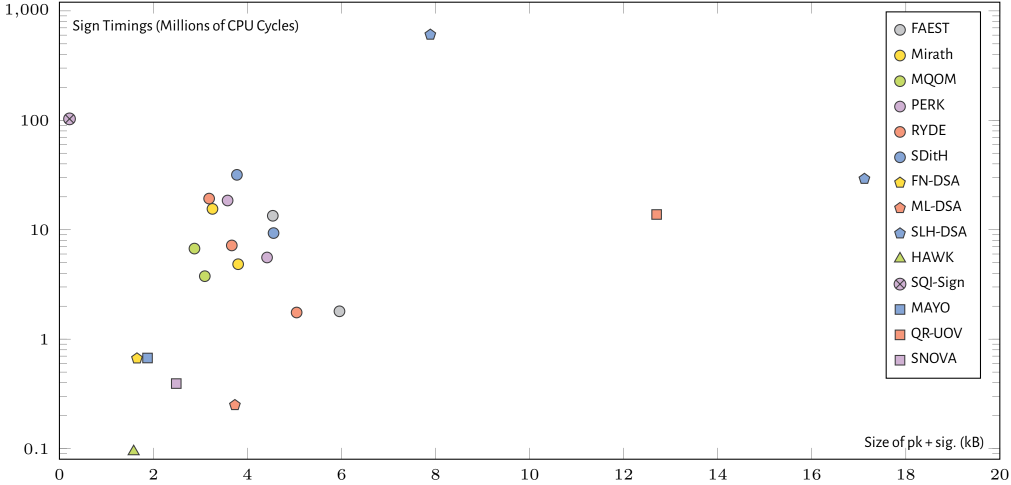
Comparaison between MPCitH and other Signatures



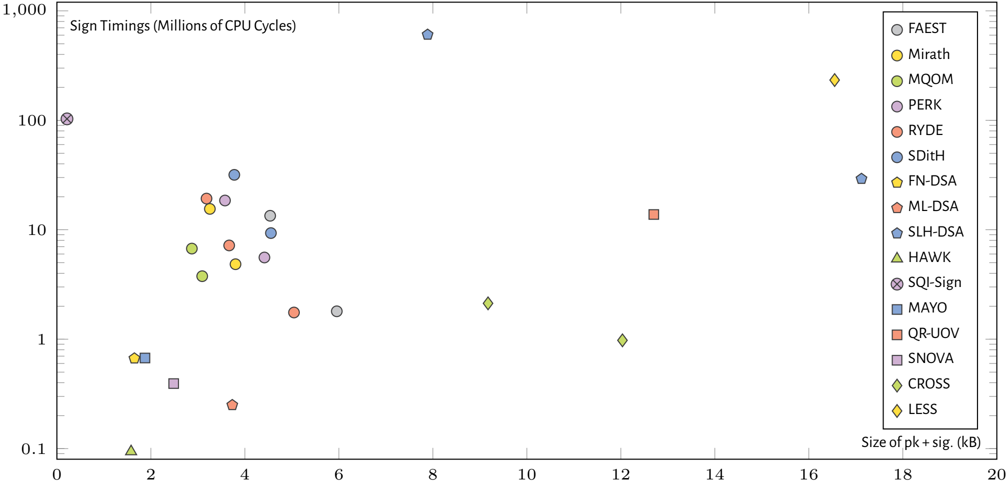
Comparaison between MPCitH and other Signatures



Comparaison between MPCitH and other Signatures



Comparaison between MPCitH and other Signatures



Concluding Remarks

Security - Conservative schemes based on the hardness of ***unstructured*** problems

Concluding Remarks

Security - Conservative schemes based on the hardness of *unstructured* problems

- FAEST - Based on **AES** one-wayness assumption
- Mirath - Based on **MinRank** problem
- MQOM - Based on Multivariate Quadratic (**MQ**) problem
- PERK - Based on the Permuted Kernel Problem (**PKP**)
- RYDE - Based on the Rank Syndrome Decoding (**RSD**) problem
- SDitH - Based on the Syndrome Decoding (**SD**) problem

Concluding Remarks

Security - Conservative schemes based on the hardness of ***unstructured*** problems

Design - Many improvements between round 1 & 2, design now stabilizing

- *Modeling* - All the schemes now employ "natural and short" witness representation
- *MPCitH* - Efficient proof systems with small overhead (scaling with modeling degree)
- *GGM Trees* - No consensus yet, different choices & instantiations across schemes

Concluding Remarks

Security - Conservative schemes based on the hardness of ***unstructured*** problems

Design - Many improvements between round 1 & 2, design now stabilizing

- *Modeling* - All the schemes now employ “natural and short” witness representation
- *MPCitH* - Efficient proof systems with small overhead (scaling with modeling degree)
- *GGM Trees* - No consensus yet, different choices & instantiations across schemes

Implementation - Significant improvements between round 1 & 2

- *Symmetric Primitives* - Different variants & choices of instantiation across schemes
- *Future Improvements* - Performance gap between schemes may decrease over time

Concluding Remarks

Security - Conservative schemes based on the hardness of ***unstructured*** problems

Design - Many improvements between round 1 & 2, design now stabilizing

- *Modeling* - All the schemes now employ "natural and short" witness representation
- *MPCitH* - Efficient proof systems with small overhead (scaling with modeling degree)
- *GGM Trees* - No consensus yet, different choices & instantiations across schemes

Implementation - Significant improvements between round 1 & 2

- *Symmetric Primitives* - Different variants & choices of instantiation across schemes
- *Future Improvements* - Performance gap between schemes may decrease over time

- MPCitH schemes are interesting candidates to complement NIST PQC portofolio -

Thank you for your attention.