|  |  |  |
| --- | --- | --- |
| Topic | Paper | Sum Up |
| Quantum Threat to Classic Encryption | Quantum computing RSA encryption: a threat and a solution (FMTAD, 2023) | FMTAD highlights quantum’s capability with ~372 qubits and suggests NFC/AES hardware-based mitigations |
| Quantum Computing: The Demise of Traditional Cryptography (Pathum, 2024) | Pathum provides a broader overview—how quantum breaks long-standing encryption and why it demands new defenses . |
| Polynomial-Time Algorithms for Prime Factorization… (Shor, 1997) | Shor introduced a quantum method to factor integers and compute discrete logarithms in polynomial time, undermining RSA and ECC foundations . This remains the fundamental threat driver. |
| The Impact of Quantum Computing … (Mavroeidis et al., 2018) | Mavroeidis et al. overview quantum’s impact on current systems |
| The Future of Cybersecurity in the Age of Quantum Computers(Raheman, 2022) | Highlights real-world breaks of PQC schemes (including Rainbow and another NIST candidate) and proposes "zero-vulnerability computing" (ZVC) as an architectural defense beyond cryptography. |
| What Is Quantum Computing? | IBM (Shneider et al. 2024) | Introductory explainer on quantum computing principles and their emerging real-world implications |
| NIST process & post-quantum readiness | Post‑Quantum Cryptography Standardization (NIST CSRC, 2017) | These NIST CSRC reports define how to evaluate future PQC: security strength, implementation cost, and algorithmic properties for candidates in standardization |
| Security Evaluation Criteria… & Cost…Criteria…Security… & Algorithm & Implementation Characteristics (NIST CSRC 2025) |
| Post‑Quantum Cryptography: Digital Signature Schemes | CSRC | NIST(Nist CSRC, 2022) | Official NIST overview page outlining the post‑quantum digital signature standardization effort, including updates on Dilithium, Falcon, SPHINCS+, etc. |
| Digital Signature Standards | Digital Signature Standard (DSS) FIPS 186‑5 (NIST, 2023) | Details current DSS e.g. RSA/ECDSA |
| Module‑Lattice‑Based Digital Signature Standard FIPS 204 (2024) | Introduce lattice-based (likely CRYSTALS‑Dilithium) and stateless hash-based (SPHINCS+) schemes respectively, as NIST’s first PQC signature standards |
| Stateless Hash‑Based Digital Signature Standard FIPS 205 (2024) |
| PQC algorithm profiles | Post‑quantum cryptography Algorithm's standardization… (Kumar, 2022) | Kumar surveys PQC families (lattice, code-based, isogeny, hash, multivariate) |
| Post Quantum Cryptography…Review… (Bavdekar et al., 2023) | Bavdekar et al. review PQC techniques, challenges, and NIST process |
| Challenges of PQ Digital Signing in Real Applications (Tan et al., 2022) | Surveys PQ signature adoption across 14 sectors, assessing suitability of six NIST-pq3 candidates, and identifies remaining deployment gaps. |
| Post‑Quantum Digital Signatures in Transport Documents (Moskvin, 2022) | Discusses PQ digital signatures’ role in transport/logistics e-docs, stressing urgent need for standardized quantum-resistant schemes. |
| CSRC Presentation: Navigating Floating‑Point Challenges in Falcon(NIST CSRC 2024) | Discusses floating‑point concerns in Falcon’s keygen, with mitigation strategies for robust FIPS‑compliant implementation. |
| PQC Signature Schemes Analysis and/or Comparison | BUFFing signature schemes…post‑quantum signatures (Cremers et al., 2021) | Cremers et al. analyze security properties of PQC signatures beyond unforgeability |
| Drop‑In‑Replaceability Analysis… (TSP et al., 2023) | TSP et al. compare NIST PQC signatures (Kyber, Dilithium, Falcon, SPHINCS+) in performance and integration ability, as well as security |
| Performance…Android Email Plug-in (Mandev & Kavun, 2023) | Tests PQC signatures (via liboqs) in Android email, finding Dilithium fast in key operations |
| Security Comparisons of PQC Signatures (Raavi et al., 2021) | Compares Dilithium, Falcon, and Rainbow using DW-cost metrics and analyses in TLS/TCP contexts—offers design trade-offs between security and implementation load |
| Applicability in Constrained Environments (Vidakovic & Milicevic, 2023) | Evaluates Dilithium, Falcon, SPHINCS+ across IoT/smart cards/blockchain—finds Dilithium leads in low-power, Falcon excels in verification speed, SPHINCS+ strongest security at cost of efficiency |
| Metric Application on Dilithium/Falcon (Rautell et al., 2022) | Assesses cryptographic metrics on lattice-signatures and suggests improvements for more comprehensive evaluation during PQC standardization. |
| Performance Analysis for Wireless Sensor Networks (Senor et al., 2024) | Simulates large WSN operations using Dilithium, Falcon, SPHINCS+, Kyber, NTRU, Saber—Falcon+Kyber is best for scalability, though combinations vary per context. |
| Falcon / CRYSTALS / Rainbow / SPHINCS+ spec sites (and docs) | The PQC finalist sites (Falcon, Rainbow, CRYSTALS, SPHINCS+) document the designs of NIST finalist schemes |
| Mathematical Perspective on PQC (Richter et al., 2022) | Offers algebraic overview of NIST Round 3 PQC finalists—Kyber, NTRU, Saber, McEliece, Dilithium, Falcon, Rainbow—targeted at mathematics researchers. |
| Performance Analysis of Post‑Quantum Cryptography Algorithms for Digital Signature(Opiłka et al. 2024) | Benchmarks Dilithium, Falcon, SPHINCS+ (via liboqs) against RSA, focusing on keygen, signing, verify—useful for 5G/6G service selection. |
| PQC Schemes Optimizations | CUSPX: Efficient GPU Implementations of **SPHINCS+** (Wang et al., 2024) | Wang et al. accelerate SPHINCS+ by 5 100× on RTX 3090 GPUs by achieving novel ways of parallelism |
| Efficient Hardware RNS Decomposition for **Falcon** (Coulon et al., 2023) | Coulon et al. propose FPGA-based residue decomposition blocks speeding Falcon key-gen by ~3.9× over software . |
| Accelerating **Falcon** on ARMv8 (Y. Kim et al., 2022) | Optimizes Falcon's polynomial FFT/NTT via ARMv8 NEON (a kind of parallel processing unit) for Cortex-A series, yielding 15–69% performance improvements across key generation, signing, and verifying. |
| Winograd for NTT…FPGA (Mandal & Basu Roy, 2024) | Applies high-radix Winograd NTT to PQC—radix‑16 for Dilithium, radix‑8 for Falcon, mixed-radix for Kyber—resulting in lower latency and fewer modular multipliers. Confirmed via FPGA implementation |
| KiD Framework: Unified NTT for Kyber & **Dilithium** (Mandal & Basu Roy, 2023) | FPGA design distributing radix‑2 butterfly units, shared memory pipeline, supporting both Kyber and Dilithium—outperforming standalone implementations. |
| Verifiable Random Subsets for **SPHINCS+** (Yehia et al., 2021) | Proposes a verifiable ORS mechanism improving SPHINCS+ performance (~27% fewer hashes and provides a 82.9% reduction in computation costs), which may help close the performance gap for hash-based PQC |
| A low‑cost configurable hash computing circuit for PQC (Xi et al., 2023) | Proposes an FPGA-based Keccak hash unit shared across Kyber and Dilithium, saving ≈40% LUTs and 14% FFs, and clocked at 391 MHz. |
| Optimizing **Dilithium** Implementation with AVX2/-512 (Runqing et al., 2024) | Enhances Dilithium performance by ∼23%, 17%, and 14% for keygen, sign, verify under AVX2/AVX512 via parallel NTT optimizations and advanced sampling/packing. |
| Efficient Error Detection for **Falcon** & Saber hardware (Sarker et al., 2022) | Introduces error-detection schemes in FPGA for Falcon’s Gaussian sampler and Saber KEM, achieving ~99.9975% coverage with ≤23% resource overhead. |
| CRYSTALS‑Dilithium Engine on GPGPU (Wright et al., 2022) | Demonstrates GPU-accelerated Dilithium (via RBC + PUFs) achieving 70–90× speedups over CPU implementations across security levels |
| Software/Hardware Co‑Design of Dilithium (Zhou et al., 2021) | FPGA co-design (Karatsuba modular mult, NTT twiddle generator) yields 11× and 7× faster signing/verification than C on soft-core and 51%/31% boosts on Cortex‑A9. |
| Rejection Sampling Revisited – **Dilithium** Parameters (Zheng et al., 2021) | Proposes tighter rejection-sampling bounds to avoid entropy trade-offs in PQC—boosts efficiency by ~60% and signature size by ~14% without reducing security. |
| Handling Vinegar Variables to Shorten **Rainbow** Keys (Zambonin et al., 2019) | Optimizes Rainbow keys by reusing vinegar variables, reducing private key size by ~85% while preserving security, and enabling 3.5× total key reduction. |
| Side Channel Resistant **Sphincs+**(Fluhrer et al., 2024) | Proposes an SLH‑DSA‑like signer for SPHINCS+ resilient to power/EM side‑channel attacks; incurs ~1.7× slowdown |
| On Protecting **SPHINCS+** Against Fault Attacks(Genêt et al., 2023) | Analyzes vulnerabilities in non-top subtree signing due to fault injection; proposes and evaluates countermeasures |
| Improving Speed of **Dilithium's** Signing Procedure(Ravi et al. 2020) | Proposes early‑rejection optimizations to significantly speed up Dilithium signing while preserving correctness. |
| Revisiting the Constant‑Sum Winternitz One‑Time Signature with Applications to **SPHINCS+** and XMSS (Zhang et al. 2023) | Improves WOTS checksum efficiency; proposes methods potentially useful for SPHINCS+ and XMSS implementations |
| PQC Schemes attacks | Side‑Channel Attack on **CRYSTALS‑Dilithium** (Chen et al., 2021) | Chen et al. show a CPA side-channel extract secret bits from Dilithium with ~157 power traces, improving attack runtime 7.8× |
| Fault Attacks Sensitivity of Public Parameters in the **Dilithium** Verification(Viera et al. 2024) | Identifies and models fault attacks on Dilithium’s verification and proposes practical countermeasures. |
| Breaking **Rainbow** Takes a Weekend on a Laptop(Ward. 2022) | Demonstrates a practical break of the Rainbow signature scheme in ~weekend on consumer hardware, showing its vulnerability despite NIST candidacy. |
| Practical Public Template Attacks on **CRYSTALS‑Dilithium**… (Qiao et al., 2023) | Introduces a side-channel Public Template Attack on both unprotected and masked Dilithium, recovering private keys within hours on real hardware with 10k–680k traces—a leap ahead of prior methods |
| In-depth Correlation Power Analysis…**Dilithium** (Wang et al., 2024) | Applies CPA and advanced POI/ITR techniques to FPGA implementations of Dilithium, using ≥70k traces to recover partial keys; optimization reduced required traces by up to 25% |
| Novel Power Analysis Attack against **Dilithium**… (Y. Liu et al., 2024) | Introduces two efficient CPA variants—optimized fast two-stage and single-bit—that outperform 2021 schemes by up to 367×, further compromising Dilithium on ARM implementations. |
| Signature Correction Attack on **Dilithium** Signature Scheme(Islam, 2022) | Includes RSA-based fault attacks via Rowhammer, signature correction, threshold signature vulnerabilities, and fault injection on verification—demonstrating broad practical threats. |
| Improved Power Analysis Attacks on **Falcon** (Zhang et al., 2023) | Analyzes Falcon’s Gaussian samplers; uses covariance-based CPA on both base and sign-flip leaks to recover Falcon-512 keys with ≤220k traces (~30 min), outperforming prior attacks. |
| Faulting Winternitz One‑Time Signatures to Forge LMS, XMSS, or **SPHINCS+** (Wagner et al., 2023) | Demonstrates a novel fault injection on WOTS that bypasses its checksum, enabling existential or universal forgeries across LMS/XMSS/SPHINCS+—affecting both signing and verification. Includes theoretical analysis and practical countermeasures |
| Number "Not Used" Once – Practical Fault Attack... (Ravi et al., 2019) | Injecting faults on nonce usage in LWE-based schemes (NewHope, Kyber, Frodo, Dilithium) on ARM Cortex-M4 causes nonce reuse, enabling key & message recovery with ≤10 faults and 100% success |
| Correction Fault Attacks on Randomized **Dilithium** (Krahmer et al., 2024) | Studies vulnerabilities in hedged (randomized) Dilithium to fault correction attacks, filling gaps overlooked since deterministic mode exploits. |
| Single-Trace Side‑Channel Attacks on **Dilithium** (Wang et al., 2023) | Demonstrates a power analysis side-channel attack on Dilithium-2’s secret key unpacking. With deep learning and minimal traces (even a single trace with 9% success), the secret key can be partially or fully recovered, especially when aided by public key compression. Highlights critical risks of single-trace attacks on ARM Cortex-M4 implementations. |
| Efficient Side‑channel Attack on **Dilithium** (Qiao, Liu et al., 2024) | Shows that with just two signatures, private key disclosure in 5 mins is possible via regression/CNN-based profiled attacks on ARM Cortex. |
| On Protecting **SPHINCS+** Against Fault Attacks(Genêt et al., 2023) | Analyzes vulnerabilities in non-top subtree signing due to fault injection; proposes and evaluates countermeasures |
| Breaking Category Five **SPHINCS+** with SHA‑256(Perlner et al., 2022) | Demonstrates a forgery attack against SHA‑256–based SPHINCS+ (Cat‑5) reducing classical security by ≈40 bits |
| SHIFT SNARE: Uncovering Secret Keys in **FALCON** via Single‑Trace Analysis(Qiu et al., 2025) | Recovers full FALCON‑512 secret key from a single power trace targeting a 63‑bit right shift; ~99.9999% key recovery success |
| **FALCON** Down: Breaking FALCON Signature Scheme through Side‑Channel Attacks(Karabulut et al., 2021) | Uses EM leakage from FFT floating-point multiplications to extract full secret key in ~10k traces on Cortex‑M4 |
| Attack Analysis on Two‑party Signature and Threshold Signature Based on **Dilithium**(Wu et al. 2023) | Shows that two‑party and threshold protocols using Dilithium are insecure—private key and intermediate values can be exposed with nearly 100% success. |
| Exploiting Determinism **in Lattice-based Signatures**: Practical Fault Attacks on pqm4 Implementations of NIST candidates (Ravi et al. 2019) | Shows real-world fault attacks on deterministic Dilithium implementations on Cortex-M4; leaks secret key components and suggests mitigation. |
| On the Security of Lattice‑Based Fiat‑Shamir Signatures in the Presence of Randomness Leakage(Liu et al. 2021) | Shows that even minimal randomness leakage per signature enables full key recovery (e.g., Dilithium‑III in ~10 s), validated on Dilithium and qTESLA. |
| PQC Schemes Use-Cases | **Lattice-based** Access Authentication for Quantum Networks (Wang & Long, 2024) | Proposes an authentication scheme for quantum networks combining Dilithium signatures and Kyber KEM—achieving mutual authentication, confidentiality, integrity. |
| Post-quantum secure boot using **hash-based signatures** (Wagner et al., 2024) | Designs a hybrid software–hardware secure boot leveraging stateful (LMS/XMSS) or stateless (SPHINCS+) hash-based signatures, comparing implementations to classical schemes. |
| Challenges of PQ Digital Signing in Real Applications (Tan et al., 2022) | Surveys PQ signature adoption across 14 sectors, assessing suitability of six NIST-pq3 candidates, and identifies remaining deployment gaps. |
| Application and Implementation of **Multivariate** Public Key Cryptosystem in Blockchain(Shen et al. 2019) | Demonstrates integrating Rainbow signatures on a private Ethereum blockchain and compares their efficiency against ECDSA. |