The Picnic Digital Signature Algorithm

NIST Third PQC Standardization Conference
June 2021

Melissa Chase, David Derler, Steven Goldfeder, Daniel Kales, Jonathan Katz, Vladimir Kolesnikov, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Xiao Wang and **Greg Zaverucha**

Picnic Overview

Security depends only on problems related to symmetric key primitives

- Secure hash function (ROM/QROM analysis implies all the usual properties: CR, PR, etc.)
- Secure block cipher (key recovery given a single plaintext/ciphertext pair)
- Unique design, conservative assumptions

The core of Picnic is an efficient zero knowledge proof for binary circuits

- Create a signature scheme using a non-interactive proof
- Use the Fiat-Shamir transform

Performance characteristics

- Keys are small, signatures are relatively large, possible to tradeoff speed/size

Picnic Signatures

Key Generation:

Generate a random plaintext block p

Generate a random secret key sk

Compute C = LowMC(sk, p)

Picnic public key is pk = (C, p), signing key is sk

Sign(sk, pk, m):

Prove knowledge of sk such that C = LowMC(sk, p)

Message m and public key pk are bound to the proof when computing the challenge

Picnic signature is the proof

Picnic Signatures

Key Generation:

Generate a random plaintext block p

Generate a random secret key sk

Compute C = LowMC(sk, p) Must be hard to recover sk

Picnic public key is pk = (C, p), secret key is sk

Sign(sk, pk, m):

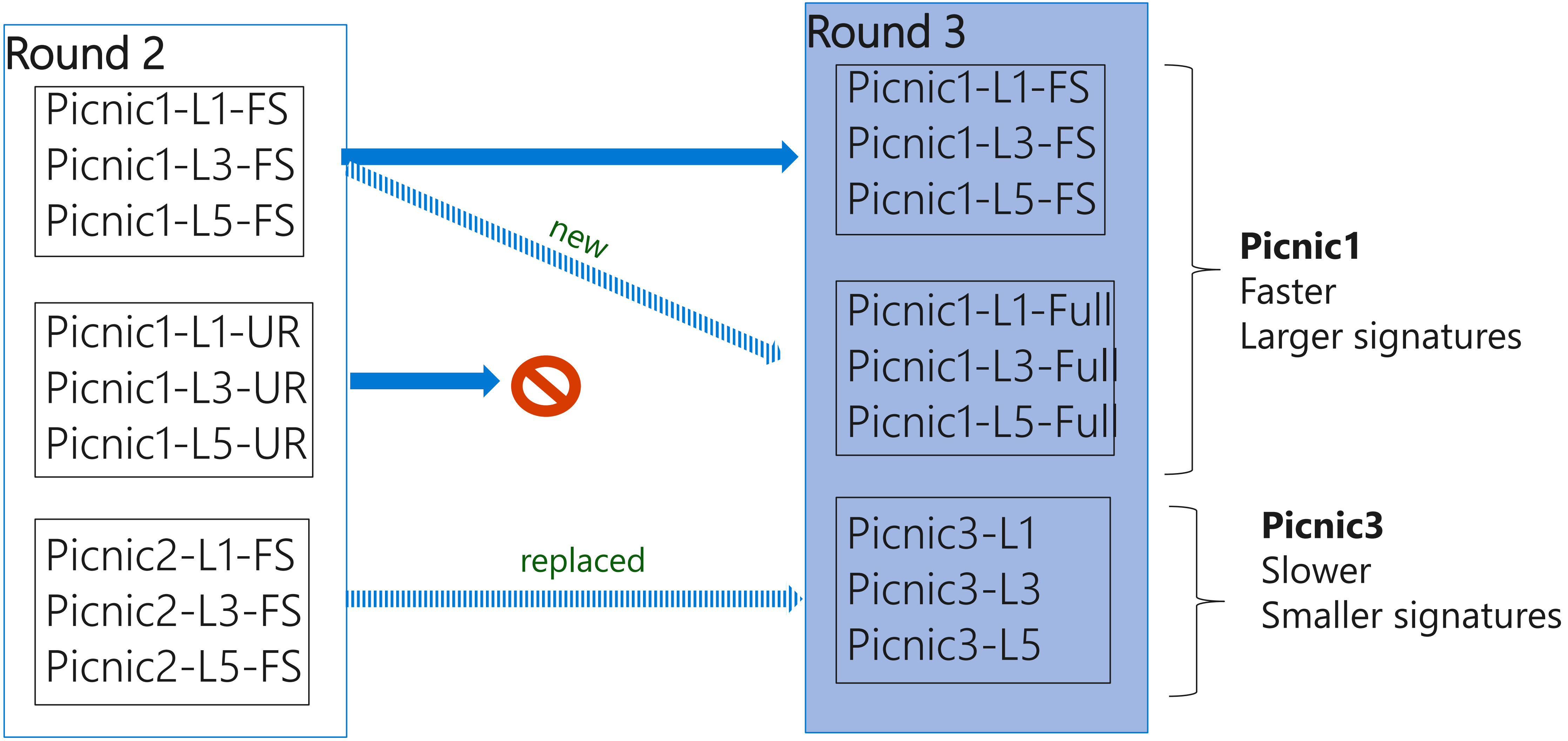
Prove knowledge of sk such that C = LowMC(sk, p)

Message m and public key pk are bound to the proof when computing the challenge

Picnic signature is the proof Must be zero-knowledge

Changes for Round 3

Parameter Sets



Picnic1-Full

- Same as Picnic1-FS but with alternative LowMC parameters
  - The number of 3-bit S-boxes per round is flexible in LowMC
  - Picnic previously used only LowMC instances with a **partial** S-box layer
  - Partial: needs more rounds (e.g., 20) vs Full: needs far fewer (e.g., 4)
  - LowMC Cryptanalysis Challenge [LCC] contains instances of both type
  - Partial has (arguably) more margin due to Dinur’s cryptanalysis [Din21] of Full

- Fewer rounds means faster sign and verify times and fewer round constants (lower code size)
- Signatures are slightly shorter (fewer total AND gates)

Picnic3

- Tweaks to Picnic2 aimed at making Sign and Verify faster
- Picnic's ZK proof is done using the MPC-in-the-head technique
  - Prover simulates an N-party MPC protocol, commits to the execution, then reveals parts of it to the verifier
  - **Reduce the number of MPC parties** N from 64 to 16. ~4x less computation
- Change LowMC parameters to use a Full S-box layer
- Tweaks to the MPC protocol
- Security analysis still holds

Considering adding a more conservative option with a Partial S-box layer LowMC instance, due to [Din21] (as we have with Picnic1)

Performance

Security level L1, Intel Xeon W-1233 @ 3.6GHz

Parameter set	Sign (ms)	Verify (ms)	Size (bytes)
Picnic1-FS	1.37	1.10	32,862
Picnic1-Full	1.0	0.8	30,821
Picnic2	40.95	18.20	12,341
Picnic3	5.17	3.96	12,595
Picnic2, N=16	10.42	5.00	13,831

See [KZ20] for additional details and benchmarks for L3 and L5

New Picnic implementations:
Resource-constrained and side-channel protected

Resource-Constrained Implementations

- Our x64 implementations do not optimize for memory use or code size
- Recent effort to reduce RAM usage, target is the ARM Cortex-M4
  - Must carefully compute parts of the signature as-needed
  - Recompute some values, rather than store
  - E.g., in Picnic3-L1 there are 250 MPC instances, and the prover will need 36 of them to respond to the challenge; we recompute these 36 rather than storing all 250
- Simple tweaks can further reduce RAM usage and improve times
  - (e.g., order that values are hashed/derived)

Masked Picnic

[GSE20] and [SBWE20] demonstrated some probing side-channel attacks on Picnic1.

We recently applied these attacks to Picnic3, and found a new one [ABE+21]. Analyzed, implemented and benchmarked a masked version of Picnic signing.

Takeaway: Under mild assumptions, masking overheads can be as low as 1.86x (for first-order protection). Watch the talk on Tuesday!

ARM Cortex-M4 Benchmarks

Using [pqm4] on the STM32F4 device, security level L1

Parameters	Implementation	Sign	Verify	Sig. Size
Picnic3-L1	opt	304M cycles 32KB RAM	203M cycles 32KB RAM	12.6KB
	opt-mem	310M cycles 24KB RAM		
	masked	546M cycles 32KB RAM		
Picnic1-L1-FS	opt	289M cycles 4KB RAM	126M cycles 4KB RAM	32.9KB
Picnic1-L1-Full + Tweaks	lowmem-mod	152M cycles 4.2KB RAM	55M cycles 3.5KB RAM	30.8KB

Ongoing effort: https://github.com/dkales/picnic_m4

Research and Security Analysis

Security Analysis

[GHHM20]: found a mistake in the QROM proof of Picnic3 and helped correct it.

Also generalizes results on the fault-resistance of Picnic (and related FS schemes) from [AOTZ20] from the ROM to the QROM

Tight QROM security of Picnic ([DFMS21], [Cha19])

- The recent results in [DFMS21] improve the existing, highly non-tight results for Picnic3.
- Alternate approach to tighter Picnic1 QROM analysis in [Cha19, 2021 revision].

Security Analysis

[CDF+20]: Picnic has stronger security properties than standard strong unforgeability. These properties are sometimes implicitly relied on by protocols. (Watch the talk on Wednesday!)

Efficient Implementation

[WBS20]: multiple ways to reduce Picnic1 RAM usage, and how to stream parts of the signature from a resource constrained device to a host device. Some ideas implemented in our M4 implementations of Picnic1.

AES-Based Signatures

FAQ: Can Picnic use AES instead of LowMC?

- Yes, but larger signatures

[BDK+21]: Banquet, a new AES-based MPCitH signature scheme

[DKR+21] and [DOT21]: Slightly faster and shorter signatures with AES

- Based on Banquet. Still slower or larger than Picnic3 (depending on parameters), but the gap is much smaller than previously
- E.g., 13.2KB signatures and ~20ms sign/verify

[DKR+21]: ciphers with large S-boxes (based on field inversion) improve performance **much** further

- E.g., 5.5 KB signatures and same sign/verify times as Picnic3, or
  8.5 KB signatures with sign/verify times <1 ms

Formal Verification

There is some recent work towards a formally verifying MPC-in-the-head (MPCitH) ZK proof protocols.

[SOS21]: formal verification of MPCitH protocols, EasyCrypt **machine checked security proof** of ZKBoo (a simpler version of the proof system used in Picnic1).

[ABEG+21]: Machine checked security proofs **and implementations** for MPCitH (more general than Picnic but simpler MPC protocol)

Summary

Compared to Round 2:

- Picnic3 is 4.5x to 13.9x faster than Picnic2
- New Picnic1 parameter sets are ~40% faster and ~10% shorter
- Implementations for the M4

Ongoing strong interest from the research community is amazing!

More information: microsoft.github.io/Picnic/

References are at the end of the presentation.

References

[ABEG+21] J. Almeida, M. Barbosa, K. Eldefrawy, S. Graham-Lengrand, H. Pacheco and V. Pereira. **Machine-checked ZKP for NP-relations: Formally Verified Security Proofs and Implementations of MPC-in-the-Head**. arXiv report 2104.05516.

[ABE+21] D. Aranha, S. Berndt, T. Eisenbarth, O. Seker, A. Takahashi, L. Wilke and G. Zaverucha. **Side-Channel Protections for Picnic Signatures.** 3rd NIST PQC Standardization Conference, IACR ePrint 2021/735.

[AOTZ20] D. Aranha, C. Orlandi, A. Takahashi and G. Zaverucha. **Security of Hedged Fiat-Shamir Signatures under Fault Attacks**. EUROCRYPT 2020, IACR ePrint 2019/956.

[BDK+21] C. Baum, C. de Saint Guilhem, D. Kales, E. Orsini, P. Scholl and G. Zaverucha. **Banquet: Short and Fast Signatures from AES**. PKC 2021, IACR ePrint 2021/068.

References

[Cha19] A. Chailloux. Tight quantum security of the Fiat-Shamir transform for commit-and-open identification schemes with applications to post-quantum signature schemes. IACR ePrint 2019/699.

[CDF+20] C. Cremers, S. Düzlü, R. Fiedler, M. Fischlin and C. Janson. BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures. IEEE S&P 2021, 3rd NIST PQC Standardization Conference, IACR ePrint 2020/1525.

[DKR+21] C. Dobraunig, D. Kales, C. Rechberger, M. Schofnegger, G. Zaverucha. Shorter Signatures Based on Tailor-Made Minimalist Symmetric-Key Crypto. IACR ePrint 2021/692.

[DFMS21] J. Don and S. Fehr and C. Majenz and C. Schaffner. Online-Extractability in the Quantum Random-Oracle Model. IACR ePrint 2021/280.

References

[DOT21] C. de Saint Guilhem, E. Orsini, T. Tanguy. **Limbo: Efficient Zero-knowledge MPCitH-based Arguments.** ACM CCS 2021, IACR ePrint 2021/215.

[Din21] I. Dinur. **Cryptanalytic Applications of the Polynomial Method for Solving Multivariate Equation Systems over GF(2).** EUROCRYPT 2021, IACR ePrint 2021/578.

[GSE20] T. Gellersen, O. Seker and T. Eisenbarth. **Differential power analysis of the Picnic signature scheme.** PQCrypto 2021, IACR ePrint 2020/267.

[GHHM20] A. Grilo, K. Hövelmanns, A. Hülsing, C. Majenz. **Tight adaptive reprogramming in the QROM.** QIP'21, IACR ePrint 2020/1361.

[KZ20] D. Kales and G. Zaverucha. **Improving the Performance of the Picnic Signature Scheme.** TCHES 2020, IACR ePrint 2020/427.

References

[pqm4] M. Kannwischer, J. Rijneveld, P. Schwabe and K. Stoffelen. PQM4: Post-quantum crypto library for the ARM Cortext-M4. https://github.com/mupq/pqm4

[SBWE20] O. Seker, S. Berndt, L. Wilke and T. Eisenbarth. SNI-in-the-head: Protecting MPC-in-the-head protocols against side-channel analysis. ACM CCS 2020, IACR ePrint 2020/544.

[SOS21] N. Sidorenco, S. Oechsner and B. Spitters. Formal security analysis of MPC-in-the-head zero-knowledge protocols. CSF 2021, IACR ePrint 2021/437.

[WBS+20] J. Winkler, A. Holler and C. Steger. Optimizing Picnic for Limited Memory Resources. Proceedings of Euromicro Conference on Digital System Design, DSD 2020.