The Picnic Digital Signature Algorithm

NIST Second PQC Standardization Conference
August 2019

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

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 or Unruh 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), 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

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):

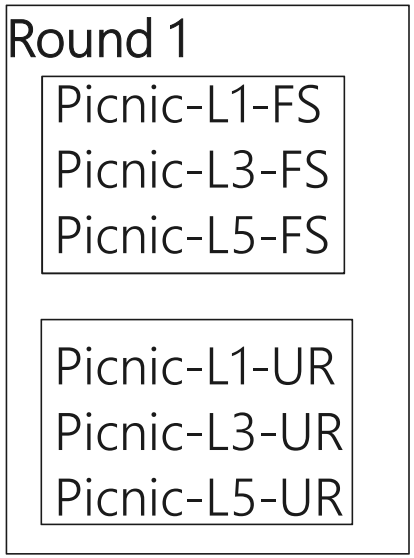
Prove knowledge of sk such that C = LowMC(sk, p)

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

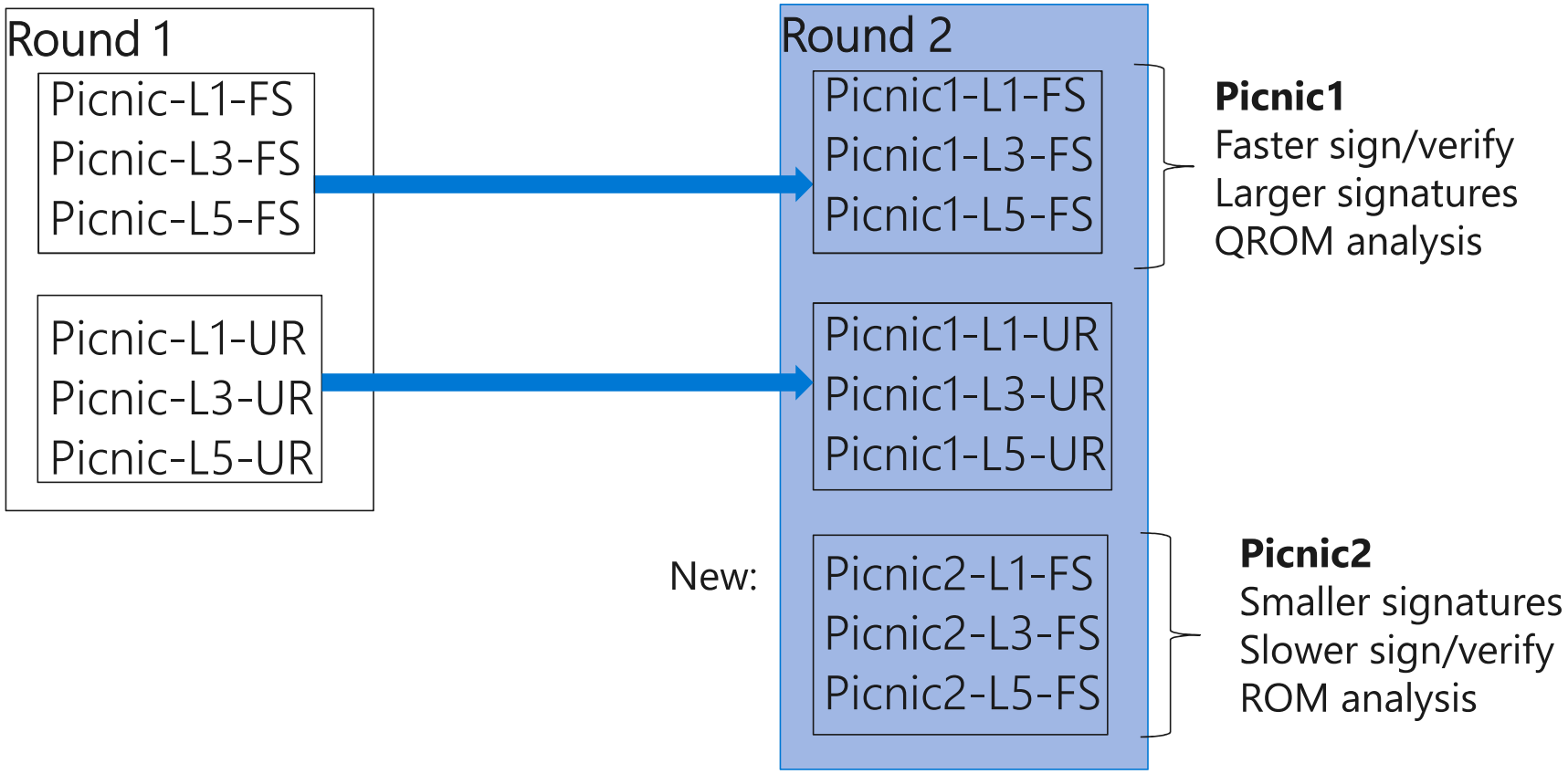
Picnic signature is the proof ← Must be zero-knowledge

Changes for Round 2

Parameter Sets



Parameter Sets



Picnic2 Parameter Sets

- Picnic1 uses the ZKB++ proof system
- Picnic2 uses a different proof system "KKW" with shorter proofs
  - *Improved Non-Interactive Zero Knowledge with Applications to Post-Quantum Signatures*. J. Katz, V. Kolesnikov and X. Wang. CCS 2018.
- KKW has a similar design as ZKB++: works gate-by-gate, uses parallel repetition to boost soundness, same security assumptions and symmetric-key primitives
- KKW uses a different underlying MPC protocol
  - MPC in the pre-processing model
  - More flexibility in parameters, e.g., more parties
  - Better cut-and-choose strategy

Updated Security Analysis

Picnic2: ROM proof with concrete analysis (in design document). QROM security still conjectural.

Picnic1-FS: new QROM proofs

Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model
J. Don, S. Fehr, C. Majenz and C. Schaffner, CRYPTO 2019

Revisiting Post-Quantum Fiat-Shamir
Q. Liu and M. Zhandry, CRYPTO 2019

Quantum security of the Fiat-Shamir transform of commit and open protocols
A. Chailloux, ePrint 2019/699

Parameter sets using Unruh's transform may be obsolete

Other Round 2 Changes

Mitigation for multi-target attacks – each signature now includes a random salt included in each hash function call, along with a per-call counter

Multi-target Attacks on the Picnic Signature Scheme and Related Protocols. I. Dinur and N. Nadler. EUROCRYPT 2019

Optimized implementation uses a faster, more memory efficient LowMC implementation technique

Linear Equivalence of Block Ciphers with Partial Non-Linear Layers: Application to LowMC. I. Dinur, D. Kales, A. Promitzer, S. Ramacher and C. Rechberger. EUROCRYPT 2019

Performance

Performance Highlights

AVX2 optimized implementation, Intel Xeon W-2133 CPU @ 3.60GHz

Parameter Set	*sk* (bytes)	*pk* (bytes)	Signature (bytes)	Sign (ms)	Verify (ms)
Picnic1-L1	16	32	32,838	**1.38**	**1.10**
Picnic2-L1	16	32	**12,359**	41.19	18.19
Picnic1-L3	24	48	74,134	**3.19**	**2.61**
Picnic2-L3	24	48	**27,173**	122.90	41.15
Picnic1-L5	32	64	128,176	**5.54**	**4.61**
Picnic2-L5	32	64	**46,282**	253.35	72.12

Benchmarks of the latest code
github.com/IAIK/Picnic

Significant improvements to Picnic2 speed and memory usage vs. code in the submission package

Hardware Implementation

New paper describes a VHDL implementation of LowMC and Picnic1-L1 signing and verification

Efficient FPGA Implementations of LowMC with Applications to Picnic and PSI Protocols.
D. Kales, S. Ramacher, C. Rechberger, R. Walch, and M. Werner. Under submission.

Evaluated on a Xilinx Artix-7 and Kintex-7 FPGAs.

Used 44% of Kintex-7 area, 67% of Artix-7 area

Kintex-7 clock frequency is 125 MHz, Sign takes 0.25 ms

Implementations were optimized for speed, not area

Possible Picnic2 Tweaks

Improved Picnic2 Parameters (August 2019)

Parameter Set	N	M	T	Sign (ms)	Verify (ms)	Size (bytes)
Picnic2-L1	64	343	27	41.16	18.21	12,347
Picnic2-L1-new	16	252	36	10.42 (-3.9x)	5.0 (-3.6x)	13,831 (+1.12x)
Picnic2-L3	64	570	39	123.21	41.25	27,173
Picnic2-L3-new	16	420	52	29.85 (-4.1x)	11.77 (-3.5x)	30,542 (+1.12x)
Picnic2-L5*	64	803	50	253.17	71.32	46,162
Picnic2-L5-new	16	604	68	61.09 (-4.1x)	21.19 (-3.4x)	52,863 (+1.14x)

N: number of simulated parties
M: number of simulated MPC instances
T: number of challenged MPC instances

* Only provides 252-bit security – need to increase M to 806, T to 51

Research and Security Analysis

Fault Attacks

New paper shows that *hedged* FS signatures are secure against ***some*** fault attacks

Security of Hedged Fiat–Shamir Signatures Under Fault Attacks.

D. Aranha, C. Orlandi, A. Takahashi and G. Zaverucha (ePrint)

Hedged schemes derive the per-signature randomness from a random input, the signing key and the message.

Uses Picnic2 as an example – establishes security under fault attacks

Many results apply to general FS signatures (other PQ candidates?)

Research – LowMC Security

There is now an implementation of LowMC in Q#, a language for expressing quantum algorithms. This establishes a baseline estimate the cost of a quantum key recovery attack using Grover's algorithm. Allows some comparison to AES.

Implementations of the Grover oracle for key search on AES and LowMC.
S. Jacques, M. Naehrig, M. Roetteler and F. Virdia. In preparation.

We have also specialized the existing analysis of LowMC to the parameter sets used by Picnic. When only a single plaintext-ciphertext pair is output per key, best attack works for only 8 rounds of 20 at L1 (12 of 30 at L3, and 18 of 38 at L5).

Research – Alternative block ciphers

The **Hades** block cipher has a small number of AND gates, and the linear layer is much faster than LowMC. Estimated that sign/verify performance would be 2-8x faster, with signatures 10% smaller.

Starkad and Poseidon: New Hash Functions for Zero Knowledge Proof Systems.
L. Grassi, D. Kales, D. Khovratovich, A. Roy, C. Rechberger and M. Schofnegger. ePrint 2019/458

Can we use AES instead of LowMC? Yes, but to be efficient requires changing the Picnic ZK proof to use arithmetic circuits. Signature sizes are over 2x larger than Picnic2. CPU cost TBD.

BBQ: Using AES in Picnic Signatures.
C. Delpech de Saint Guilhem, L. De Meyer, E. Orsini and N. Smart. SAC 2019

Research – Alternative Hash Functions

About 30% of the total Picnic2 CPU cost is hashing
Using KangarooTwelve instead of SHA-3 gives an overall 1.15x speedup

	Sign (ms)	Verify (ms)	Size (bytes)
Picnic2-L1-SHA3	41.16	18.21	12,347
Picnic2-L1-K12	35.51	13.26	12,347
Picnic2-L1-new-SHA3	10.42	5.0	13,831
Picnic2-L1-new-K12	8.72	3.81	13,831

Summary

Compared to Round 1:

The Picnic algorithm offers additional flexibility

Shorter signatures are now an option with Picnic2

Picnic1-FS has a QROM security analysis

We appreciate the strong interest from the research community spanning theory and implementation!

More information: microsoft.github.io/Picnic/