New Designs of Multivariate-Polynomial Universal Hash Functions

Jean Paul Degabriele · Jan Gilcher · Jérôme Govinden · Kenneth G. Paterson

ACM CCS 2026







Outline

1 **Motivation** — Why improve universal hashing?

2 **Methodology** — How do we find better trade-offs?

3 **Results** — How much faster are our designs?

δ-Universal hash in practice

Fix a target difference δ and $M \neq M'$,

$$\Pr_{r \xleftarrow{\$} R}\left[H_r(M) - H_r(M') = \delta\right] \leq \varepsilon$$

Widely deployed in AEADs

AES-GCM [NIST SP 800-38D]
↳ **GHASH**

ChaCha20-Poly1305 [RFC 8439]
↳ **Poly1305**

Ongoing NIST standardization

wGCM
↳ **256-bit wide-GHASH**

HCTR2-based Accordion
Hash-encrypt-hash paradigm

Classical polynomial

$$H_r(M) = m_1 \times r^4 + m_2 \times r^3 + m_3 \times r^2 + m_4 \times r$$

GF(q) where $q=2^n$ or q is prime

Horner's algorithm

for each block m_i:

$$y \leftarrow (y + m_i) \times r$$

Properties

This talk
2^{128}, 2^{256}

Security bound $\varepsilon \leq \frac{|M|}{q}$

1 secret key r

Parallelizable

1 field multiplication per message block

Pseudo-inner product / NMH [HK97]

$$H_r(M) = \underbrace{(m_1+r_1)\times(m_2+r_2)}_{\text{Pair 1}} + \underbrace{(m_3+r_3)\times(m_4+r_4)}_{\text{Pair 2}}$$

Properties

Security bound $\varepsilon \leq \frac{1}{q}$

Key size = message size

Parallelizable

1 field multiplication per 2 message blocks

Bernstein–Rabin–Winograd (BRW) [Ber07]

$$H_r(M) = H_r(m_1, m_2, m_3) \times (m_4 + r^4) + H_r(m_5, m_6, m_7)$$

Properties

Security bound $\varepsilon \leq \frac{2|M|}{q}$

1 secret key r

Limited parallelism

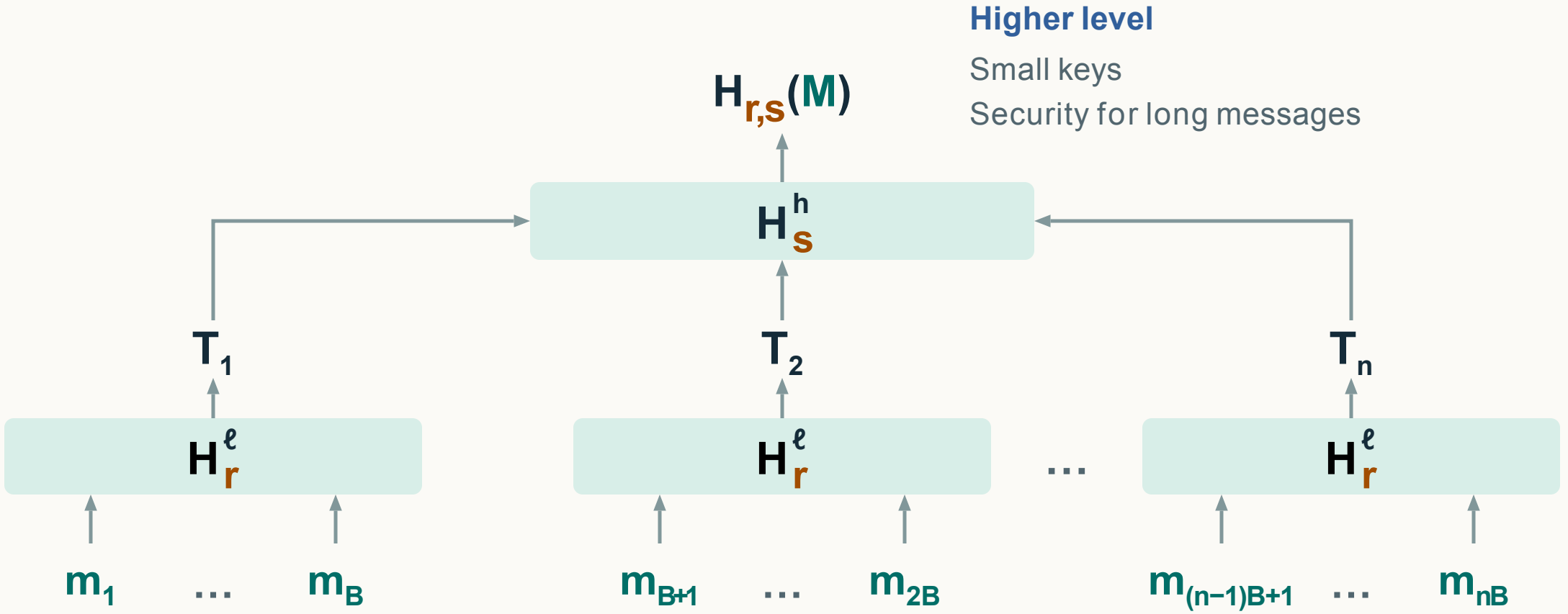
1 field multiplication per 2 message blocks

Each design falls one goal short

Design	Short key	Reduced multiplication cost	Parallel and optimization-friendly
Classical polynomial	Yes	No	Yes
Pseudo inner product	No	Yes	Yes
BRW polynomial	Yes	Yes	No known method

This work achieves all three

Generic two-level construction



How we compared the designs

01

Select the polynomials

Lower-level polynomials

MMH^+ SQH^+
NMH^+ HKM

Higher-level polynomials

tMMH tSQH tNMH
t̂BRW MHP

Security of Multivariate Polynomials

1 key

$$\frac{|M|}{q}$$

$\longrightarrow$

v independent keys

$$\frac{v \cdot |M|^{1/v}}{q}$$

How we compared the designs

01

Select the polynomials

4 lower × 5 higher level polynomials.

02

Generate the code

- 2 prime + 2 binary fields.
- 9 lower-level input sizes.
- 6 delay-reduction options.

And more…

03

Benchmark & compare

Compare designs with optimized reference implementations.

04

Use factor analysis

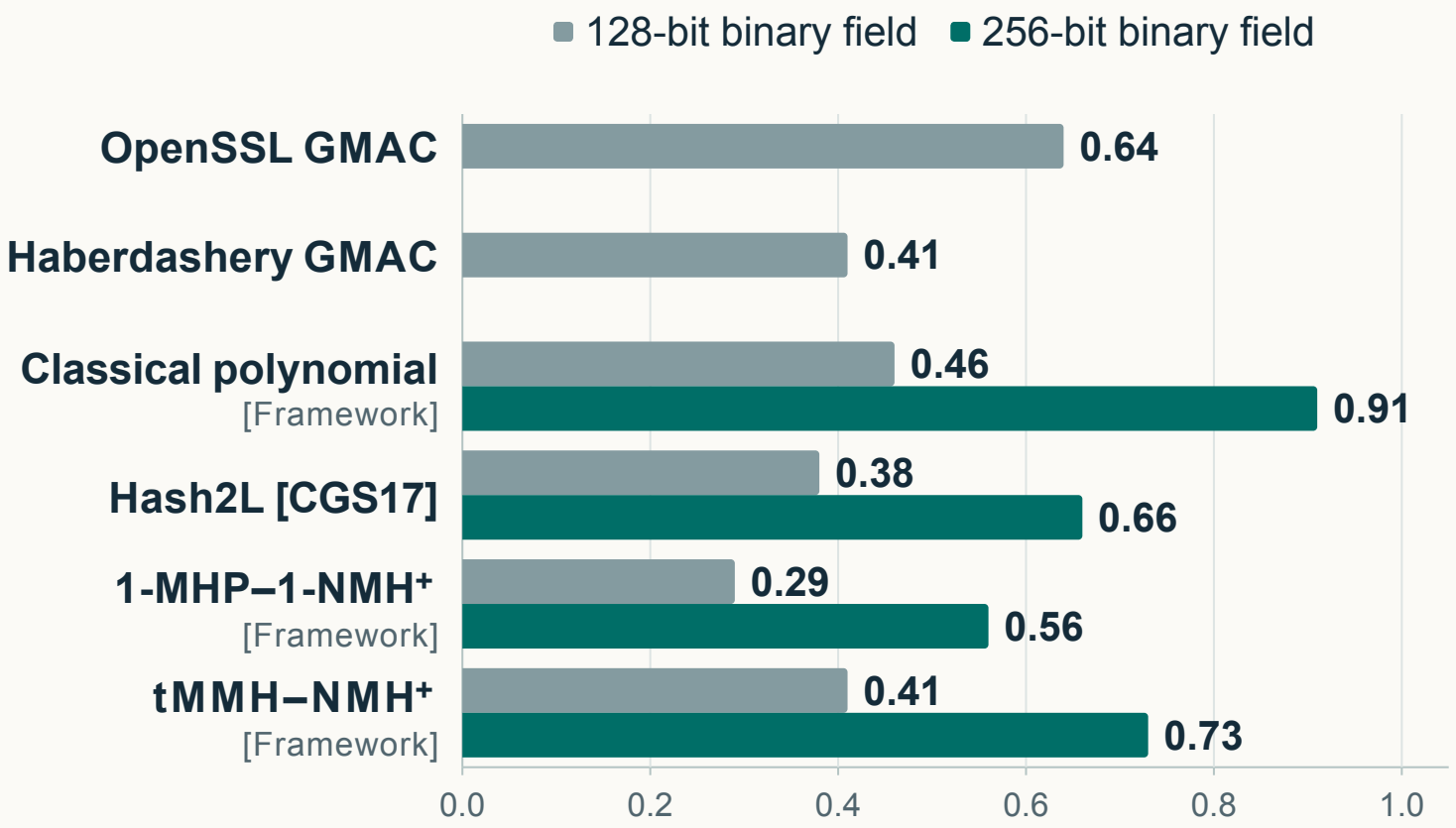
Understand interactions between design choices.

Extension of our prior work (IEEE S&P 2024)

Generates and benchmarks two-level multivariate polynomials.



Better speed–security trade-offs



Cycles per byte at 16kB · Intel Xeon Gold 6258R, 2.7 GHz.

1-MHP–1-NMH+

Fewer cycles

128-bit field	256-bit field
vs classical polynomial	
37.0%	**38.5%**
vs Hash2L	
23.7%	**15.2%**

tMMH–NMH+

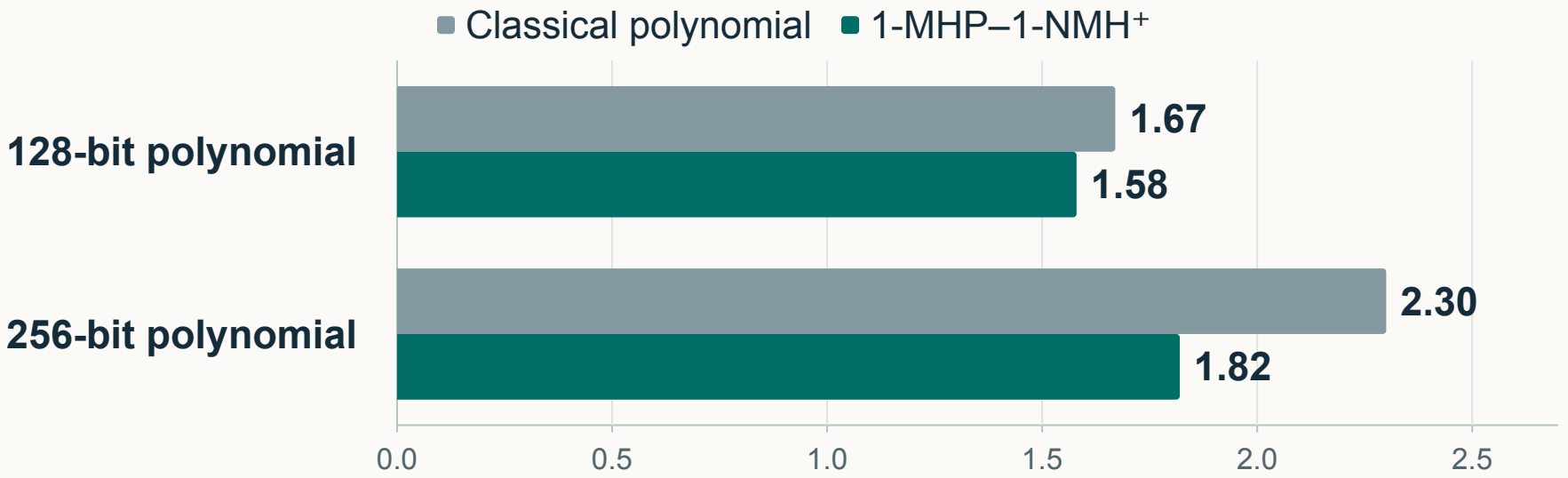
Better Security

$$\text{Classical} \quad \frac{|M|}{q} \geq \frac{\log_2 |M|}{q} \quad \text{tMMH–NMH+}$$

Performance improvements in larger primitives

In wGCM with Rijndael256

Measured cycles per byte



5.4%
fewer cycles

20.9%
fewer cycles

Cycles per byte at 16kB · Intel Xeon Gold 6258R, 2.7 GHz.

In Accordion mode with Rijndael256

Classical polynomial vs 1-MHP–1-NMH+ over a 256-bit field

up to 25.6%
estimated overall gain

Summary

- **Systematic exploration of the two-level design space**
  Multivariate designs compared for security, speed and key size.

- **New faster designs with comparable complexity**
  1-MHP–1-NMH$^+$ faster than the classical polynomial.
  tMMH–NMH$^+$ introduces a new security–efficiency trade-off.

- **Improved performance for NIST candidates**
  wGCM and Accordion under development.



ia.cr/2026/1602

Also in the full paper

Benchmarks over prime fields.

Key-size, memory and security trade-offs.

Questions?

References

[HK97] S. Halevi and H. Krawczyk. MMH: Software Message Authentication in the Gbit/second Rates.
FSE 1997, pp. 172–189.

[Ber07] D. J. Bernstein. Polynomial Evaluation and Message Authentication.
Manuscript, 22 October 2007.

[JL10] K. Jankowski and P. Laurent. Packed AES-GCM Algorithm Suitable for AES/PCLMULQDQ Instructions.
IEEE Transactions on Computers 60(1), pp. 135–138, 2011.

[GG15] M. Goll and S. Gueron. Vectorization of Poly1305 Message Authentication Code.
ITNG 2015, pp. 145–150.

[CGS17] D. Chakraborty, S. Ghosh and P. Sarkar. A Fast Single-Key Two-Level Universal Hash Function.
IACR Transactions on Symmetric Cryptology 2017(1), pp. 106–128.

[NIST SP 800-38D] M. Dworkin. Recommendation for Block Cipher Modes of Operation:
Galois/Counter Mode (GCM) and GMAC. NIST, November 2007.

[RFC 8439] Y. Nir and A. Langley. ChaCha20 and Poly1305 for IETF Protocols.
RFC 8439, June 2018.

Paper and artifact

Degabriele et al. · CCS 2026 · ePrint 2026/1602

Polynomial Hashing Framework · v2.0.3 (2026)

Degabriele et al. · Polynomial hashing SoK (2024)

Slides created with the help of ChatGPT.