SoK: Efficient Design and Implementation of Polynomial Hash Functions over Prime Fields

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

jeanpaul.degabriele@tii.ae, jerome.govinden@tu-darmstadt.de,
{jan.gilcher, kenny.paterson}@inf.ethz.ch

IEEE S&P 2024

Technology Innovation Institute

ETH zürich

TECHNISCHE UNIVERSITÄT DARMSTADT

Δ-Universal Hash in Practice

- **Definition:** Given $z \in \mathcal{T}$ and $M \neq M' \in \mathcal{M}$,

$$\Pr_{r \leftarrow \$ \mathcal{R}}\left[H_r(M) - H_r(M') = z\right] \leq \epsilon(M, M')\,.$$

- **Various practical applications:**
  - Data Structures: hash tables [CW79].
  - Message Authentication Codes: UMAC, Badger, Poly1305-AES, GMAC [ISO/IEC 9797-3].
  - AEAD: AES-GCM, ChaCha20-Poly1305 [RFC 8446].

Poly1305 [Ber05]

For $M = M_1 \| \cdots \| M_n$,

$$\text{Poly1305}(r, M) = (c_1 x^n + c_2 x^{n-1} + \cdots + c_n x^1 \mod 2^{130} - 5) \mod 2^{128},$$

where $c_i = M_i \| 1$ and $x = \text{clamp}(r, 22)$.

Key Points:

- Widely deployed, default choice (with Chacha20) in OpenSSH and WireGuard.
- Good performance across all architectures without needing specific hardware support.
- *Clamping introduced for fast implementations using FPUs (Floating-Point Units).*
  $\rightarrow$ *Almost all implementations of Poly1305 use integer ALUs (Arithmetic Logic Units).*
  $\rightarrow$ *Provides only $\approx$103 bits of security with a 128-bit key and tag.*
- *Tailored for 32-bit architectures.*
- *Limited security of ChaChaPoly in the multi-user setting due to Poly1305 [DGGP21].*

Poly1305 [Ber05]

For $M = M_1 \| \cdots \| M_n$,

$$\text{Poly1305}(r, M) = (c_1 x^n + c_2 x^{n-1} + \cdots + c_n x^1 \mod 2^{130} - 5) \mod 2^{128},$$

where $c_i = M_i \| 1$ and $x = \text{clamp}(r, 22)$.

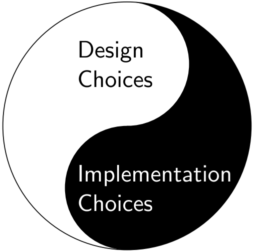
Given today's advancements and applications, would we still converge to this same design?

Systematization of Knowledge (SoK)

Current Standpoint:

- Broad design space.
- Multiple interactions between available choices.
- Knowledge spreads across research papers, cryptographic libraries, and developers' blogs.

Our Exposition [DGGP24]:



Brief Description of the Design Space

key R

message M

M_1	M_2	$\cdots$	M_ℓ

$\in \{0,1\}^*$

r

$m_1, m_2, \ldots, m_\ell$

$\in \mathbb{F}_p^*$

$P_r(m_1, m_2, \ldots, m_\ell)$

Polynomial

Brief Description of the Design Space

key R

message M

M_1 | M_2 | $\cdots$ | M_ℓ $\in \{0,1\}^*$

r

$m_1, m_2, \ldots, m_\ell$ $\in \mathbb{F}_p^*$

$$m_1 \cdot r^\ell + m_2 \cdot r^{\ell-1} + \cdots + m_\ell \cdot r$$

Classical Polynomial

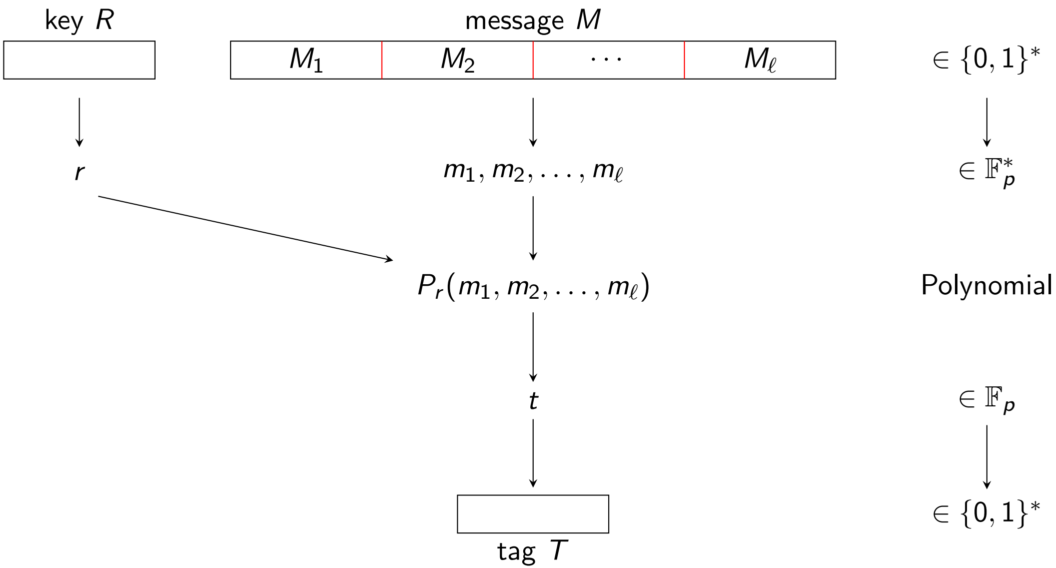
Horner

Parallel Horner

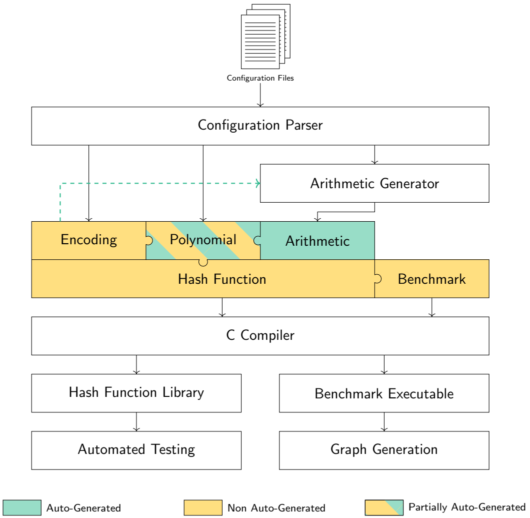
2-level

Polynomial Evaluation Strategy

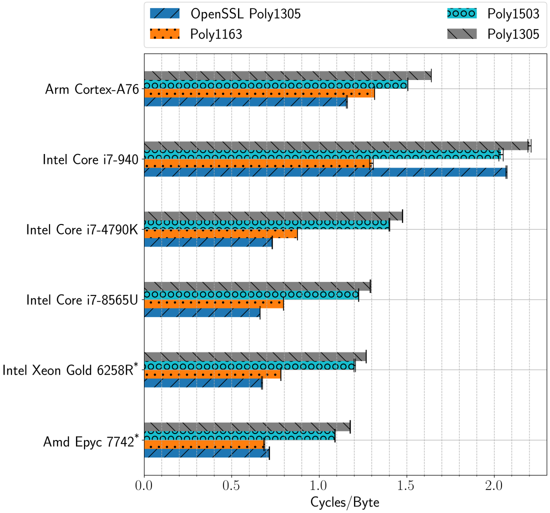
Brief Description of the Design Space



Modular Benchmarking Framework



Benchmarking New Designs



*Turbo Boost/Core Adjusted

Results:

- Our modular implementations achieve **high performance without vectorization or hand-optimization.**
- Poly1163 performance makes it **suitable as drop-in replacement for Poly1305.**

Our Expectations for Vectorization:

- Poly1163: Significantly outperforms Poly1305 at the same security level.
- Poly1503: Replacement for Poly1305 with 34 bits of extra security (103 $\rightarrow$ 137) at similar performance.

Where to Find More Details

SoK on Polynomial Hash:

https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00132

Code of Polynomial Hash Framework:

https://github.com/jangilcher/polynomial_hashing_framework

References I

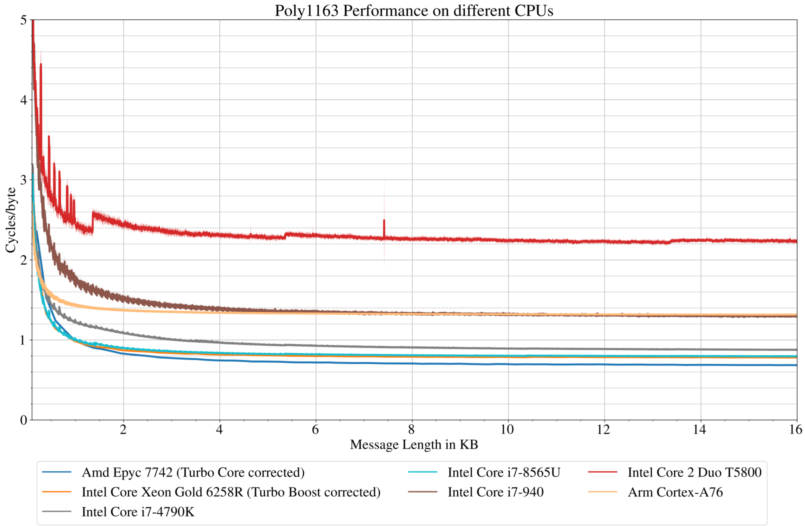
Daniel J. Bernstein.
The poly1305-AES message-authentication code.
In Henri Gilbert and Helena Handschuh, editors, *FSE 2005*, volume 3557 of *LNCS*, pages 32–49. Springer, Heidelberg, February 2005.

J Lawrence Carter and Mark N Wegman.
Universal classes of hash functions.
Journal of computer and system sciences, 18(2):143–154, 1979.

Jean Paul Degabriele, Jérôme Govinden, Felix Günther, and Kenneth G. Paterson.
The security of ChaCha20-Poly1305 in the multi-user setting.
In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 1981–2003. ACM Press, November 2021.

Jean Paul Degabriele, Jan Gilcher, Jérôme Govinden, and Kenneth G. Paterson.
Sok: Efficient design and implementation of polynomial hash functions over prime fields.
In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 131–131, Los Alamitos, CA, USA, may 2024. IEEE Computer Society.

Benchmarks: Poly1163



Benchmarks: Poly1503