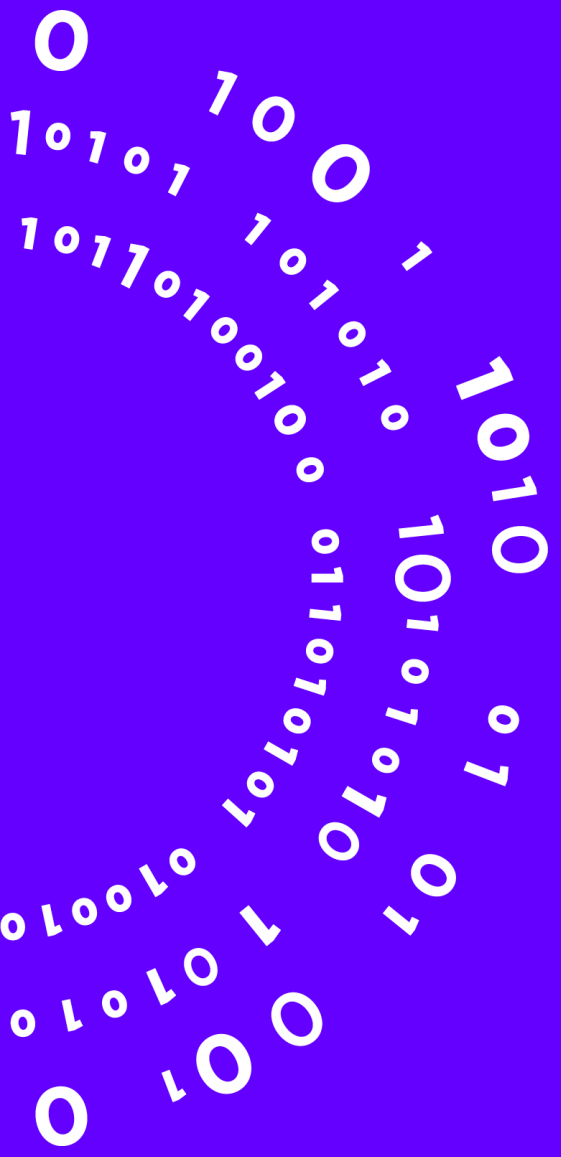


Cryptography Research Centre

Universal Hash Designs for an Accordion Mode

Jean Paul Degabriele, Jan Gilcher, Jérôme Govinden and Kenneth G. Paterson

ETH zürich

TII Technology Innovation Institute

TECHNISCHE UNIVERSITÄT DARMSTADT

TII Technology Innovation Institute

Outline

Cryptography Research Centre

- Universal Hash Functions and Accordion Mode Candidates
- Reconsidering the Design of Poly1305
- Considerations for a Universal Hash in an Accordion Mode
- Summary and Ongoing Work

Universal Hash Functions and Accordion Mode Candidates

Δ-Universal Hash Functions

Cryptography Research Centre

- Syntax: $H: \mathcal{R} \times \mathcal{M} \rightarrow \mathcal{D}$
- Δ-Universality: For all $Z \in \mathcal{D}$ and $M \neq M' \in \mathcal{M}$ it holds that:

$$Pr_{r \leftarrow \mathcal{R}}[H_r(M) - H_r(M') = Z] \leq \epsilon$$

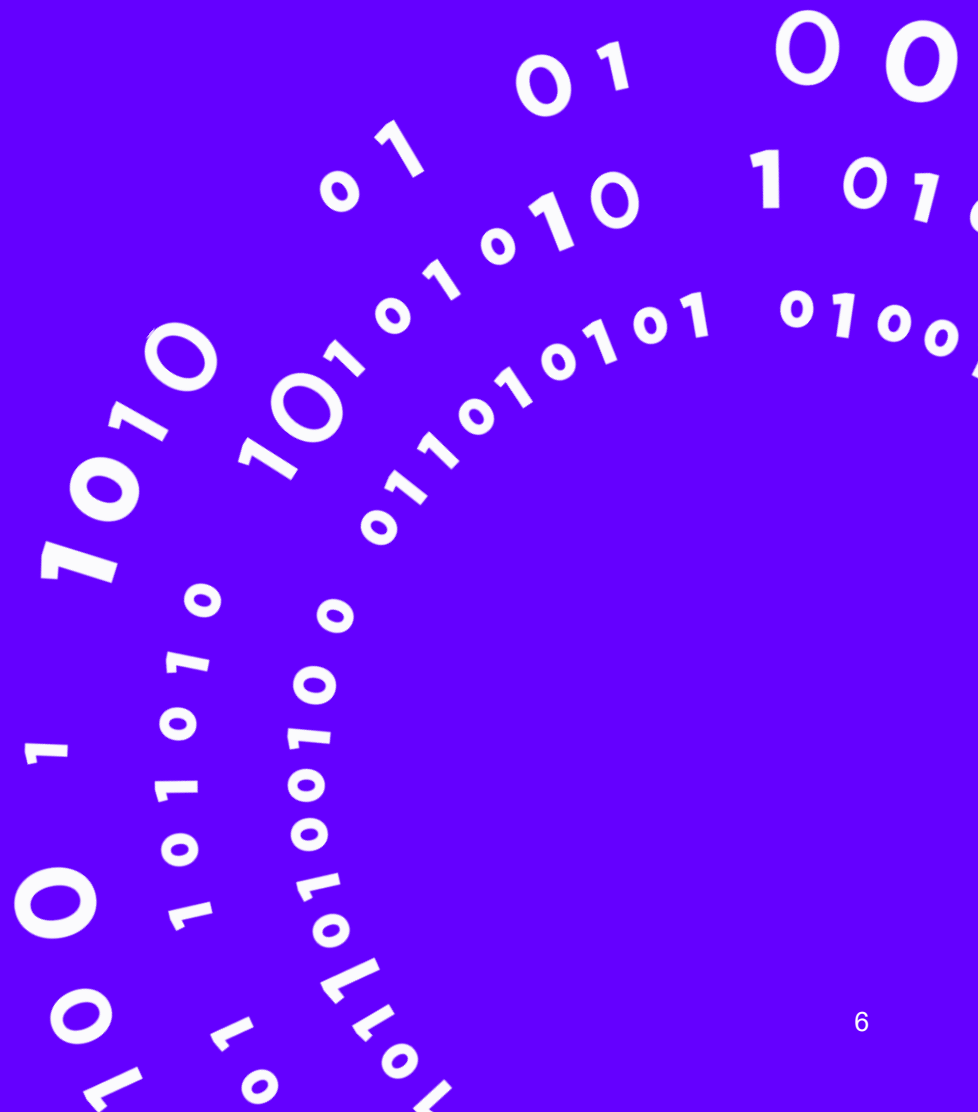
- Typical constructions are based on evaluating a polynomial over a finite field: $GF(2^n)$ or $GF(p)$.
- For polynomial hash functions, ϵ is typically an increasing function of the maximal message length in the message space.

Applications of Δ-Universal Hash Functions

Cryptography Research Centre

- Carter-Wegman MACs and AEAD: GMAC, Poly1305-AES, GCM, ChaCha-Poly1305, AES-GCM-SIV.
- Transforming a Blockcipher into a Tweakable Blockcipher: LRW2 construction.
- Various Blockcipher-based Accordion constructions: XCB, HCTR, HCTR2, TET, PIV (TCT1), AES-based Docked Double Decker, FAST, and more...
- E.g. HCTR uses **one instance of CTR and two hash instances**. Thus, a good choice of a hash function is (arguably) twice as important here.

Reconsidering the Design of Poly1305



Poly1305 and its Popularity

Cryptography Research Centre

- GHASH/POLYVAL and Poly1305 are the two dominant designs in practice.
- ChaCha-Poly1305 saw rapid adoption after RFC7539 (2015) and it is the **second most popular AEAD** scheme in use on the Internet today.
- It is known to outperform GCM on CPUs which **do not support** AES and Carry-Less-Multiplication instructions.
- It is used in TLS and is the default choice in **Wireguard**, **OpenSSH**, **OTRv4**, and the **Bitcoin Lightning Network**.

Another Look at Poly1305

Cryptography Research Centre

- It was designed by Bernstein in 2005 **for use with AES** as a CW MAC.
- Its design was intended for 32-bit architectures and specifically intended to **benefit from** the (then) superior performance of **FPUs**.
- However, all modern implementations employ **integer arithmetic** instead.
- Openssl poly1305-x86.pl:

"Besides SSE2 there are floating-point and AVX options; FP is deemed unnecessary, because pre-SSE2 processors are too old to care about, while it is not the fastest option on SSE capable ones."

What is Wrong with Poly1305?

Cryptography Research Centre

- To enable FPU implementations, 22 bits in the key are ‘clamped’ to zero, resulting in a **loss of 22 bits of security** (without a performance benefit).

- The choice of prime $(2^{130} - 5)$ is very wasteful on 64-bit architectures, resulting in a **lot of unused space** and **extra computation** when translating field elements into 64-bit limbs.

- Poly1305 and ChaCha were **designed independently** and only combined into an AEAD around a decade later by Langley.

- As such they have wildly **mismatching security levels** where the multiuser security of ChaCha-Poly1305 is not as strong as one would expect.

New Polynomial Hashes Over Prime Fields

Cryptography Research Centre

- We aimed to identify new hash designs, that **retain the benefits** offered by Poly1305, but attain **better security** and **performance** that suit today's applications and technology.

- The design space is surprisingly broad, with several complex interactions between **design** and **implementation** choices.

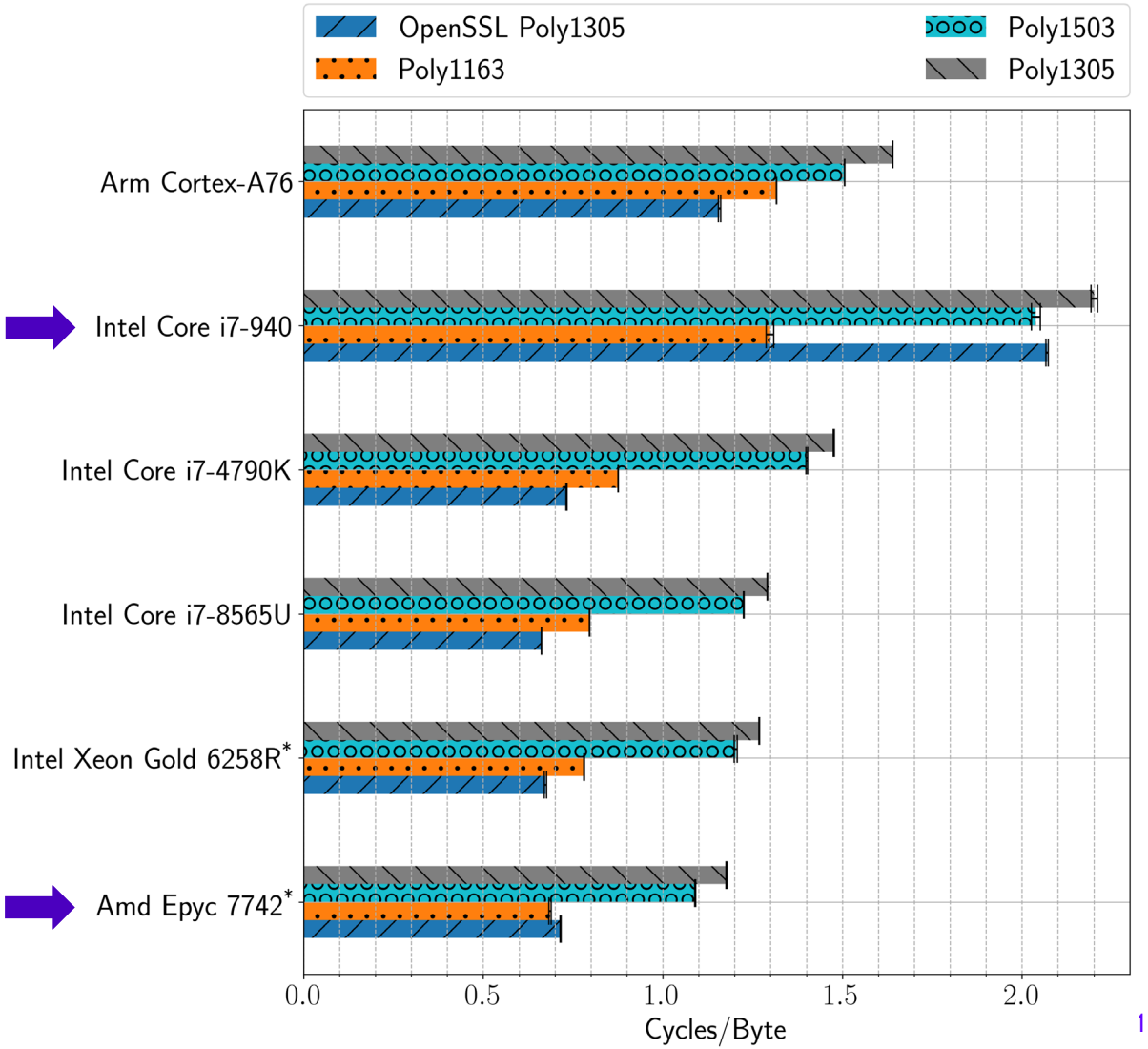
- To explore this space systematically, we built a **framework** through which we could **configure design** and **implementation parameters** and generate a C implementation of that hash.

New Designs

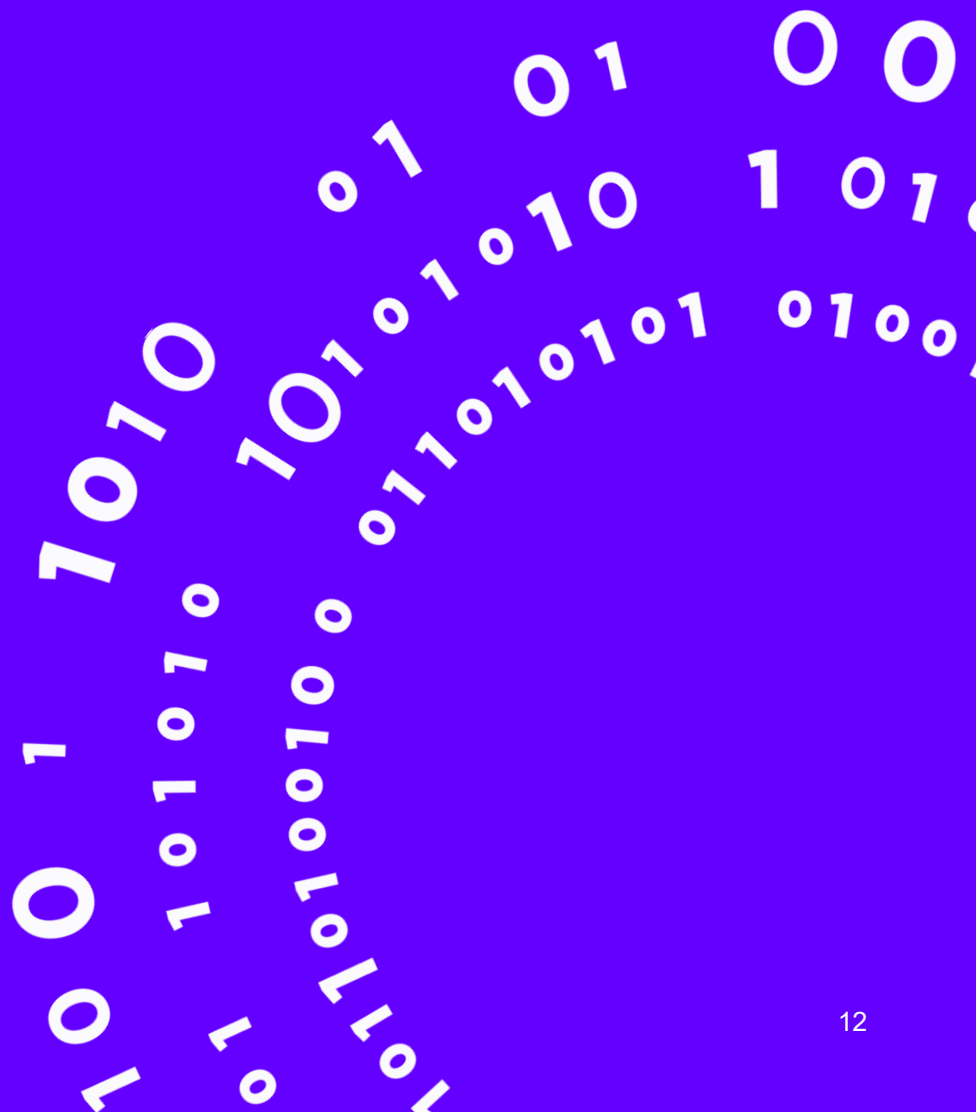
Cryptography Research Centre

Name	Prime	Message Block, Key & Tag Size	Security Level
Poly1305	$2^{130} - 5$	16 Byte	103
Poly2663	$2^{266} - 3$	32 Byte	245
Poly1743	$2^{174} - 3$	21 Byte	161
Poly1503	$2^{150} - 3$	18 Byte	137
Poly1223	$2^{122} - 3$	15 Byte	117
Poly1163	$2^{116} - 3$	14 Byte	107

- Our modular implementations achieve high performance without vectorization.
- We expect significant speedup with vectorization.



Considerations for a Universal Hash in an Accordion Mode



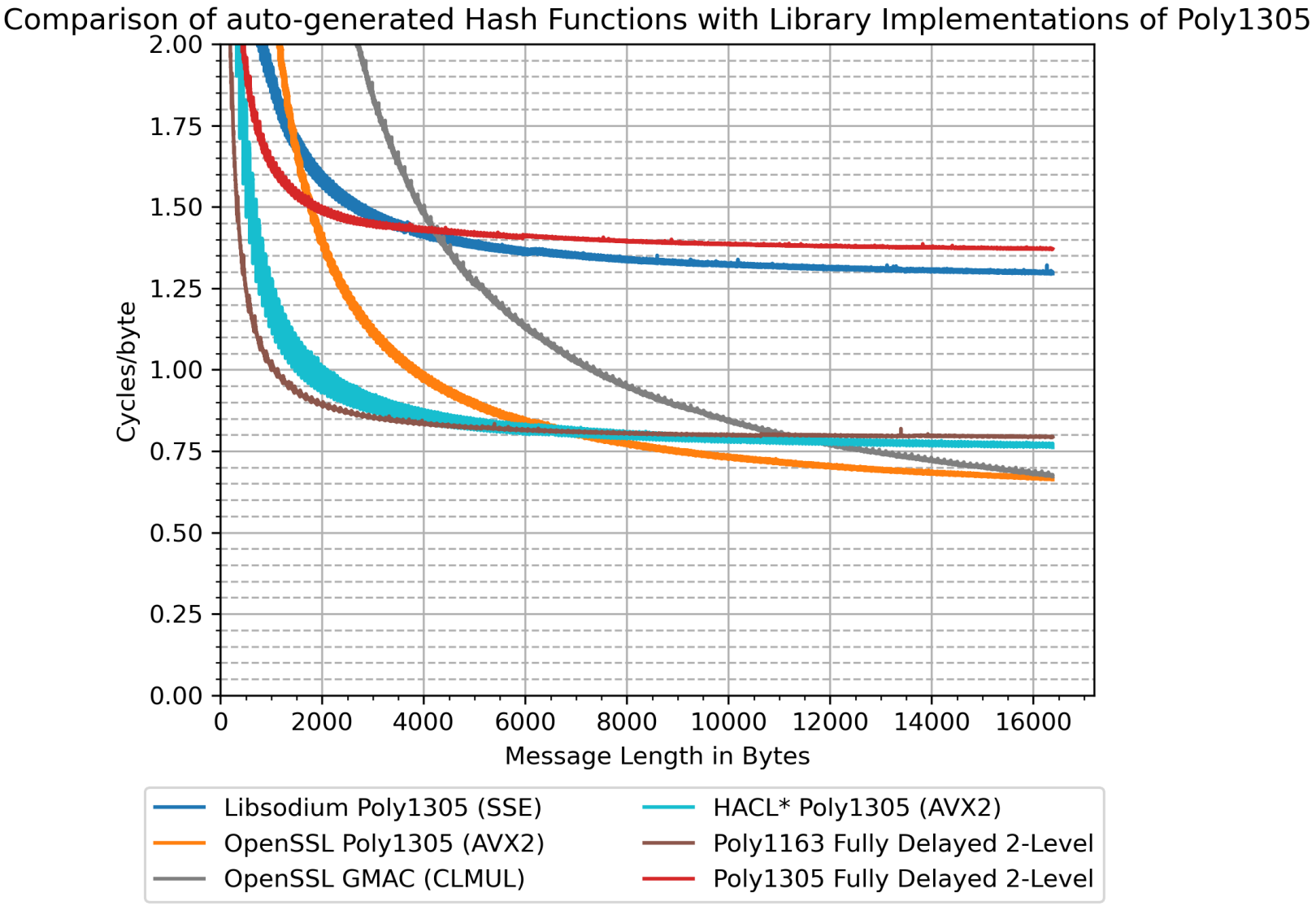
Performance Considerations

Cryptography Research Centre

- One aspect contributing to GCM's high performance is the fact that GHASH can be evaluated in **parallel** with AES on CPUs with native **AES** and **CMUL** instructions.
- **Accordion mode** designs can also (partially) exploit this. Moreover, this holds even if we use a **universal hash over a prime field**.
- Such a combination would **exploit AES-NI** when available, but also attain **better performance** when AES and CMUL instructions are **not supported**.
- Our benchmarks indicate that Poly1163 achieves **comparable performance** to GHASH even on CPUs with CMUL.

Performance Considerations

Cryptography Research Centre



Performance Considerations

Cryptography Research Centre

- Note that the **openssl implementation of GHASH** attains its **top speed** well beyond a message size of 16kB—the **maximum packet size in TLS**!
- In the case of an Accordion mode to be used for **disk encryption**, the typical disk sector size is around 4kB.
- The **asymptotic performance** (often quoted in the literature) is **misleading** for an Accordion mode. Benchmarks should be quoted for the range relevant for the intended applications.
- Currently the list of requirements **does not specify** the intended range of message lengths at which performance should benchmarked.

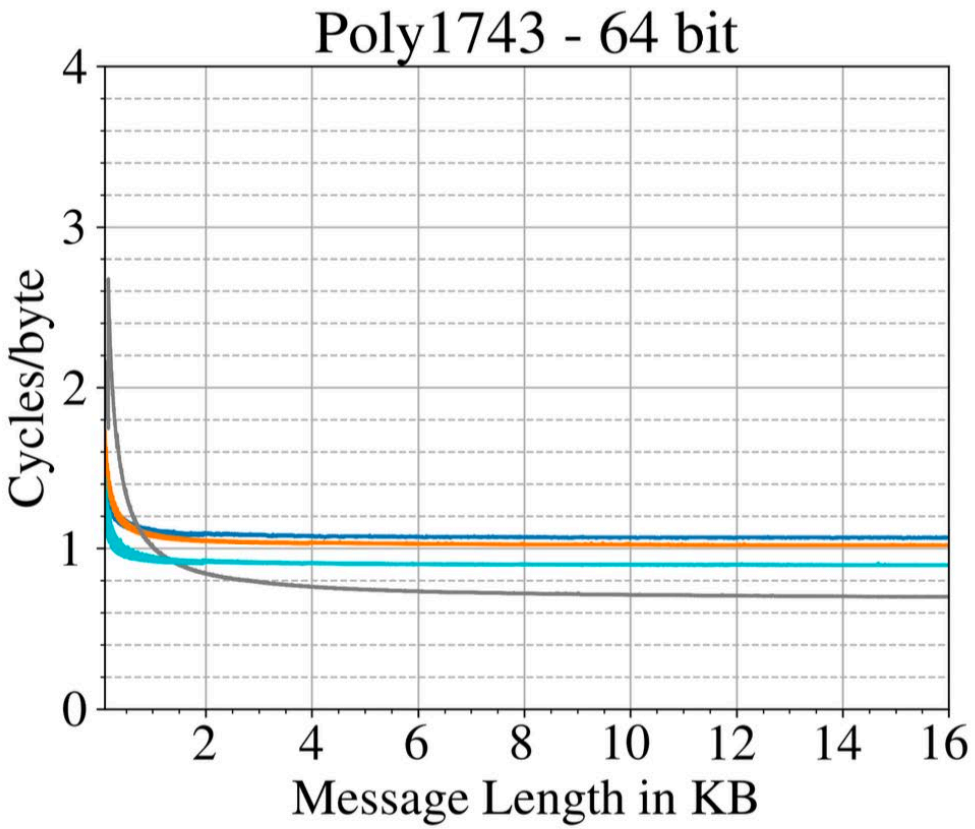
Security Considerations

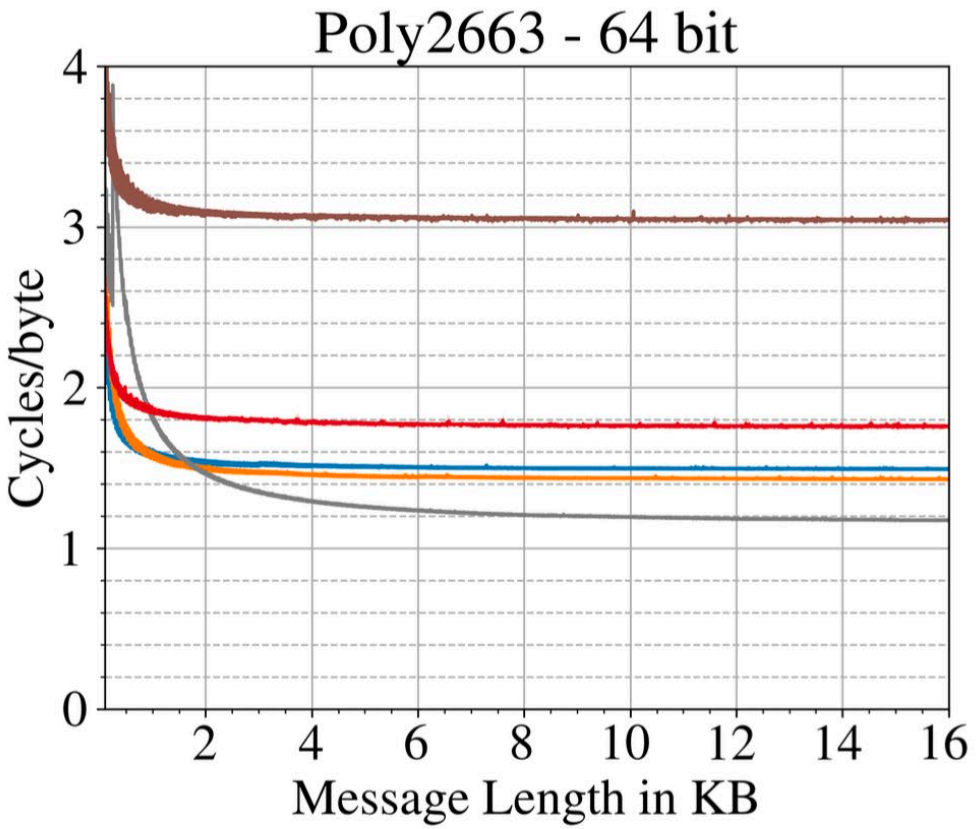
Cryptography Research Centre

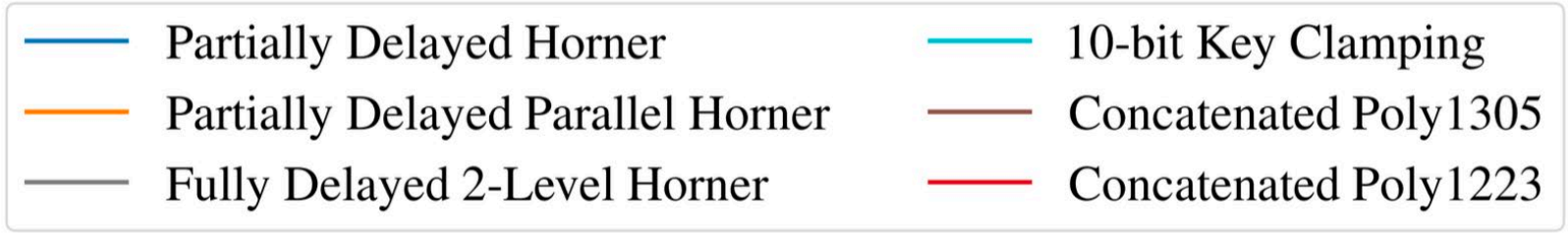
- A good choice of hash function should **match the security** of the constructions rather than limit it (unlike ChaCha-Poly1305)
- The current requirements specify a **security strength of 256 bits**—which seems to suggest a digest size of 256 bits?
- For Accordion modes using a blockcipher with a **block size of 256 bits**, what is a good choice of universal hash? Is **concatenation** a good option?
- Our new designs and benchmarks address this issue.

Security Considerations

Cryptography Research Centre







Summary and Ongoing Work

Summary and Ongoing Work

Cryptography Research Centre

- Combining AES with a universal hash over a prime field can be beneficial.
- We need **universal hash functions with larger parameters** to match a 256-bit-block blockcipher.
- We need universal hash functions that are **simple to implement** and attain good **performance** over the **intended range of message lengths**.
- Besides the new designs we recently proposed, we are **currently exploring** other hash designs both over $GF(p)$ and $GF(2^n)$.

The End