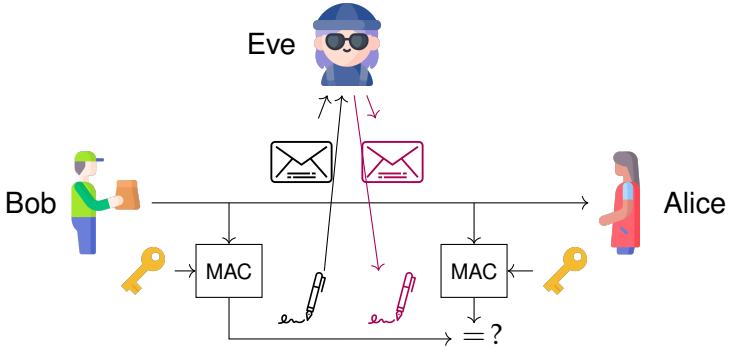
Fast AES-Based Universal Hash Functions and MACs

Featuring LeMac and PetitMac

Augustin Bariant, Jules Baudrin, Gaëtan Leurent,
Clara Pernot, Léo Perrin & Thomas Peyrin

FSE
March 20, 2025

Message Authentification Codes (MACs)



Requirement: integrity and authenticity

Eve shouldn't be able to modify the messages.

- Use Message Authentification Codes (MACs) with symmetric keys.
- Hard for Eve to predict the tag of an arbitrary message.

Different MAC design strategies

- MACs built from hash functions.
  - Ex: HMAC.
- MACs built from block ciphers.
  - Ex: CBC-MAC, CMAC, EMAC.
- MACs built from cryptographic permutations.
  - Ex: Duplex mode (AD only), Ascon mode (AD only).

Different MAC design strategies

- MACs built from hash functions.
  - Ex: HMAC.
- MACs built from block ciphers.
  - Ex: CBC-MAC, CMAC, EMAC.
- MACs built from cryptographic permutations.
  - Ex: Duplex mode (AD only), Ascon mode (AD only).

- MACs built from Universal Hash Functions (UHFs).
  - Introduced in [Carter & Wegman, 1977].
  - Some security properties of UHFs can be proven with algebraic constructions.
  - Still often requires a few block cipher calls.
  - Ex: GMAC, Poly1305, UMAC.

Our approach

- Design fast AES-based UHFs with input message of arbitrary length.
- Use generic constructions to convert UHFs into MACs.

Contribution

- New AES-based Universal Hash Function framework.
- Two proposed MAC instances: LeMac and PetitMac.

Universal Hash Functions [Carter & Wegman, 1977]

- A Universal Hash Function (UHF) is a family of functions $\{H_K : A \to B \text{ for } K \in \mathcal{K}\}$.

Definition (ε-AU UHFs)

A UHF $\{H_K : \{0,1\}^* \to \{0,1\}^n \text{ for } K \in \mathcal{K} = \{0,1\}^k\}$ is ε-almost-universal if:

$$\forall M \neq M' \in \{0,1\}^*, \qquad |\{K \in \{0,1\}^k : H_K(M) = H_K(M')\}| \leq \varepsilon|\mathcal{K}| = \varepsilon 2^k,$$

Universal Hash Functions [Carter & Wegman, 1977]

- A Universal Hash Function (UHF) is a family of functions $\{H_K : A \to B \text{ for } K \in \mathcal{K}\}$.

Definition (ε-AU UHFs)

A UHF $\{H_K : \{0,1\}^* \to \{0,1\}^n$ for $K \in \mathcal{K} = \{0,1\}^k\}$ is ε-almost-universal if:

$$\forall M \neq M' \in \{0,1\}^*, \qquad |\{K \in \{0,1\}^k : H_K(M) = H_K(M')\}| \leq \varepsilon|\mathcal{K}| = \varepsilon 2^k,$$

i.e.

$$\Pr_{K \xleftarrow{\$} \{0,1\}^k}[H_K(M) = H_K(M+\delta)] \leq \varepsilon,$$

where $\delta = M + M'$.

Universal Hash Functions [Carter & Wegman, 1977]

- A Universal Hash Function (UHF) is a family of functions $\{H_K : A \to B \text{ for } K \in \mathcal{K}\}$.

Definition (ε-AU UHFs)

A UHF $\{H_K : \{0,1\}^* \to \{0,1\}^n$ for $K \in \mathcal{K} = \{0,1\}^k\}$ is ε-almost-universal if:

$$\forall M \neq M' \in \{0,1\}^*, \qquad |\{K \in \{0,1\}^k : H_K(M) = H_K(M')\}| \leq \varepsilon|\mathcal{K}| = \varepsilon 2^k,$$

i.e.

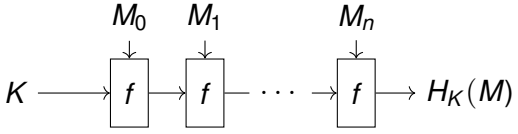
$$\Pr_{K \xleftarrow{\$} \{0,1\}^k}[H_K(M) = H_K(M+\delta)] \leq \varepsilon,$$

where $\delta = M + M'$.

- A UHF is ε-AU if no high probability differential $\delta \to 0$ exists.

Our UHF design strategy

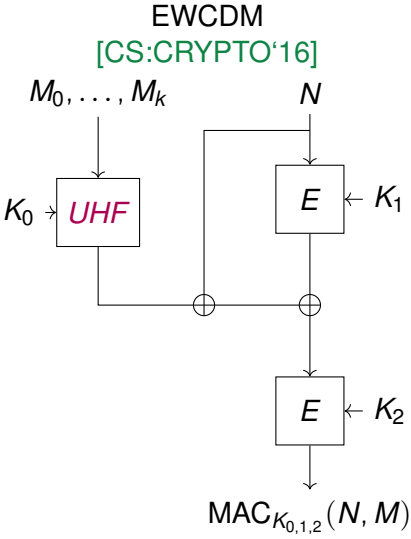
- Exploit AES-NI instructions for software performance.
- Design strategy similar to the round function of Rocca. [SLNKI,FSE'22]



- Heuristic assumptions:
  - Best differentials $\approx$ best differential trails.
  - Independent rounds.
- Security analysis: best differential trails leading to collision analysed with MILP.

Our MAC design strategy

- Use EWCDM to convert a UHF into a MAC.
- Instantiate E with the AES.
- For long messages, the costly part is the UHF.



Design of AES-based constructions

AES New Instructions (AES-NI) *[Intel, 2008]*

- Widely available instruction set on recent Intel/AMD processors
- 1 AESENC instruction = 1 AES round:

$$SB \rightarrow SR \rightarrow MC \rightarrow AK.$$

- Speed comparable to a 128-bit XOR/ADD instruction on modern processors.

Design of AES-based constructions

AES New Instructions (AES-NI) [Intel, 2008]

- Widely available instruction set on recent Intel/AMD processors
- 1 AESENC instruction = 1 AES round:

$$SB \rightarrow SR \rightarrow MC \rightarrow AK.$$

- Speed comparable to a 128-bit XOR/ADD instruction on modern processors.

Definition (Rate of an AES-based UHF/MAC)

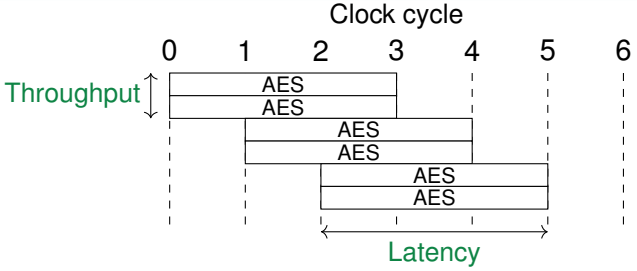
$$\text{rate} = \frac{\#\text{AES-NI instructions}}{\#\text{128-bit message blocks}}$$

- Rate 4: PelicanMAC, PC-MAC, AEGIS-128L. [DR:EPRINT'05. MT:FSE'06. WC:SAC'13]
- Rate 3: Tiaoxin-346 (AD only). [Nikolić, CAESAR'14]
- Rate 2: Jean-Nikolić, Rocca (AD only), SMAC. [JN:FSE'16. SL+:FSE'22. WM+:FSE'25]

Scheduling of AES-NI instructions

On modern processors:

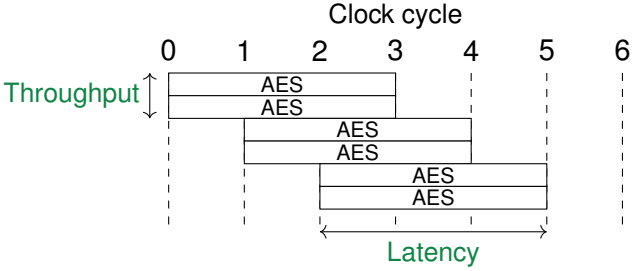
- Throughput: 2 AES per cycle.
- Latency: 3-4 cycles.



Scheduling of AES-NI instructions

On modern processors:

- Throughput: 2 AES per cycle.
- Latency: 3-4 cycles.



Theoretical bound

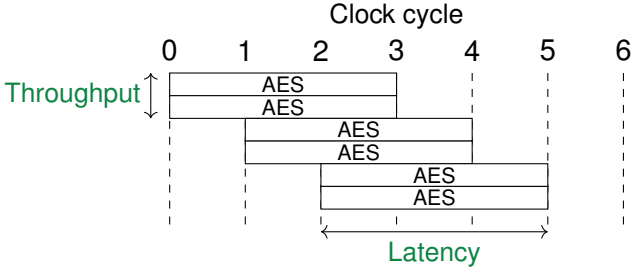
Rate-r constructions require $\geq \frac{r}{2}$ cycles per 128 bits of message.

- Observation: existing rate-2 UHFs are slower than this bound (bad parallelization).

Scheduling of AES-NI instructions

On modern processors:

- Throughput: 2 AES per cycle.
- Latency: 3-4 cycles.



Theoretical bound

Rate-r constructions require $\geq \frac{r}{2}$ cycles per 128 bits of message.

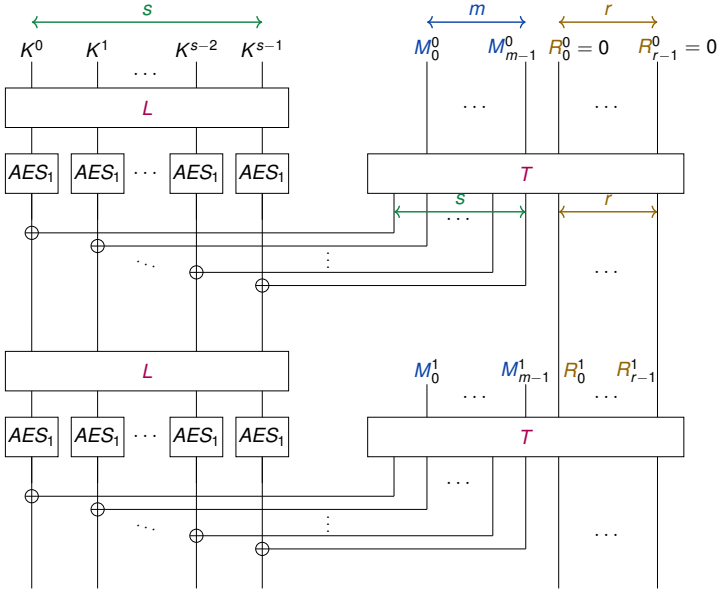
- Observation: existing rate-2 UHFs are slower than this bound (bad parallelization).

Our approach

Design a parallelization-oriented rate-2 AES-based UHF, and convert it to a MAC.

- Goal: reach the bound of 1 cycle/128-bit (= 0.0625 cycles/byte).

Our framework of UHF candidates



- Wire = 128-bit element.

Message schedule (right)

- Fully linear.
- Extra memory registers.
- Sparse linear matrix T.

Main state (left)

- Design similar to a SPN.
- Non-linear (AES rounds).
- Sparse linear matrix L.

Procedure for finding fast ε-AU candidates

Procedure: generate many random candidates of the framework. For each:

- Check the security with MILP.
- Check the performance with automatic benchmark.
- Keep candidates that are secure and performant.

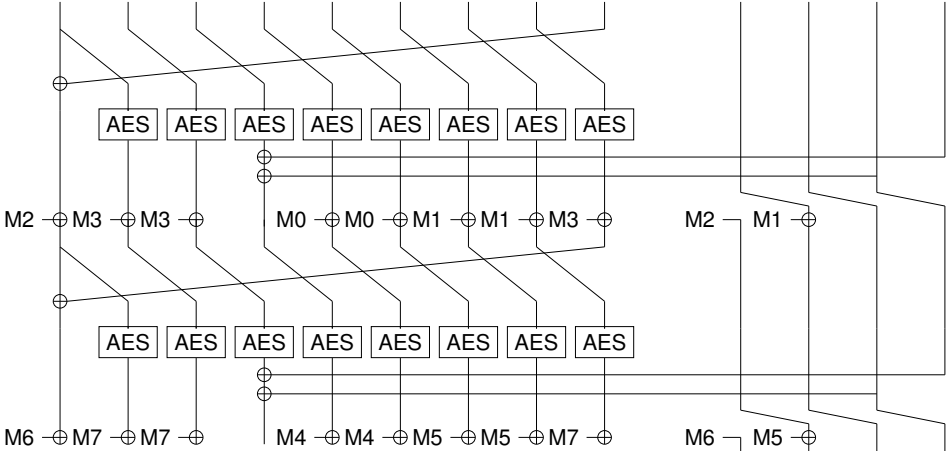
Security check

- Find the best differential trail leading to a collision with MILP.
- Secure if the number of active S-boxes is ≥ 22 (trail probability $\leq 2^{-22\times 6} = 2^{-132}$).

Performance check

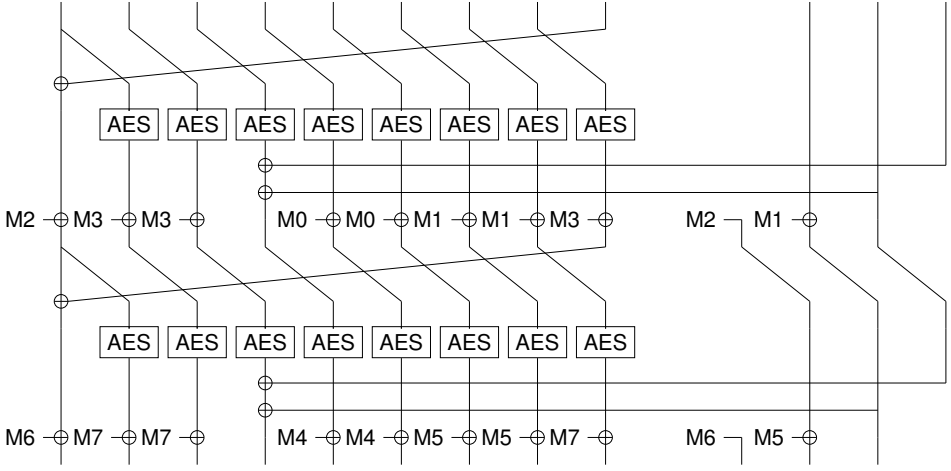
- Automatically generate a C implementation, compile and benchmark on the fly.

Round function of LeMac's UHF



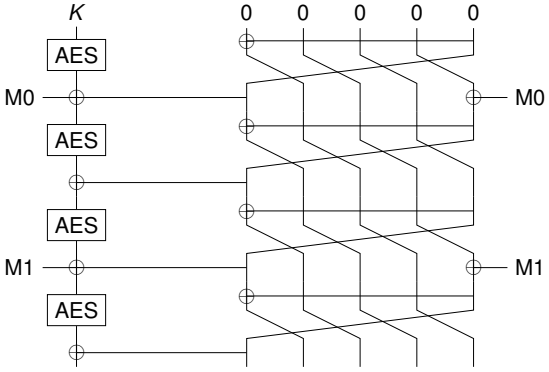
Security	*Performance*
$\geq$ 26 active S-boxes.	Rate 2 with good parallelization.

Round function of LeMac-0's UHF (corrigeandum)



Security	*Performance*
$\geq$ 25 active S-boxes.	Rate 2 with good parallelization.

Round function of PetitMac's UHF



Security	*Lightweight*
$\geq$ 26 active S-boxes.	Rate 2 with a few registers.

Result of the search for good ε-AU candidates

- We found the first secure candidate with rate < 2 (but not optimally performant).
- Performances close to the rate-2 theoretical bound (0.0625 cycles per byte).

Rate	#AES	#Message	State size	#XOR	#Active Sboxes	Speed (cy/B) 16 kB	Speed (cy/B) 256 kB
2	8	4	13	4	26	0.074	0.067
1.75	7	4	15	5	23	0.079	0.068
2	6	3	11	4	25	0.086	0.080
2	4	2	10	3	24	0.104	0.099
2	2	1	7	4	23	0.180	0.175
2	1	0.5	6	3/1	26	0.374	0.371

Result of the search for good ε-AU candidates

- We found the first secure candidate with rate < 2 (but not optimally performant).
- Performances close to the rate-2 theoretical bound (0.0625 cycles per byte).

Rate	#AES	#Message	State size	#XOR	#Active Sboxes	Speed (cy/B) 16 kB	Speed (cy/B) 256 kB	
2	8	4	13	4	26	0.074	0.067	
1.75	7	4	15	5	23	0.079	0.068	LeMac's UHF
2	6	3	11	4	25	0.086	0.080	
2	4	2	10	3	24	0.104	0.099	
2	2	1	7	4	23	0.180	0.175	
2	1	0.5	6	3/1	26	0.374	0.371	PetitMac's UHF

Performance comparison

Cipher	State size	Rate	Theoretical bound	Speed (cycles per byte)					
				Intel Ice Lake			**AMD Zen3**		
				1kB	16kB	256kB	1kB	16kB	256kB
GCM (AD only)	1	-	-	0.737	0.345	0.321	0.816	0.479	0.466
AEGIS128L (AD only)	8	4	0.125	0.393	0.207	0.195	0.358	0.183	0.174
Tiaoxin-346 v2 (AD only)	13	3	0.094	0.346	0.134	0.123	0.311	0.120	0.109
Rocca (AD only)	8	2	0.063	0.438	0.167	0.149	0.392	0.140	0.124
Jean-Nikolić	12	2	0.063	0.298	0.137	0.110	0.301	0.111	0.098
LeMac-0	12	2	0.063	0.274	0.083	0.074	0.270	0.082	0.070
LeMac	13	2	0.063	0.285	0.092	0.079	0.272	0.085	0.069
PetitMac	6	2	0.063	0.522	0.384	0.376	0.669	0.511	0.501

- LeMac: extremely performant on modern processors.
- PetitMac: lightweight design for micro-controllers.

Conclusion and future works

Results:

- Framework for fast and secure AES-based UHFs.
- Found the first rate-1.75 secure ε-AU UHF candidate.
- Two MAC instantiations with rate-2 : LeMac and PetitMac.

Future works:

- Use AVX-256 or AVX-512 instructions for further speed-up.
- Design an AES-based MAC with ARM AES instructions in mind.
- Derive an AEAD from a similar framework.

Conclusion and future works

Results:

- Framework for fast and secure AES-based UHFs.
- Found the first rate-1.75 secure ε-AU UHF candidate.
- Two MAC instantiations with rate-2 : LeMac and PetitMac.

Future works:

- Use AVX-256 or AVX-512 instructions for further speed-up.
- Design an AES-based MAC with ARM AES instructions in mind.
- Derive an AEAD from a similar framework.

Thank you for your attention