Universal Hash Functions: Overview of Applications and a New AES-Based Design

Featuring LeMac and PetitMac

Augustin Bariant

ANSSI

October 10, 2025

Introduction to Universal Hash Functions and their applications

AES-based UHFs and MACs design paradigm

Our framework of fast AES-based UHFs and MACs

Universal Hash Functions [Carter & Wegman, 1977]

- A UHF is a family of functions $\{H_K : \{0,1\}^{\leq \ell} \to \{0,1\}^n \text{ for } K \in \mathcal{K}\}$.

Definition (ε-AU UHFs)

A UHF with $\mathcal{K} = \{0,1\}^k$ is ε-almost-universal if:

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

Two main types of UHF constructions:

- Provable UHFs: e.g. algebraic constructions.
- Heuristic UHFs: e.g. based on primitives such as a PRP.

UHF design strategy (1) : provable UHFs based on polynomial evaluation

Principle of UHFs based on polynomial evaluation

- Interpret the message $M = M_0 \ldots M_{n-1}$ as a polynomial in a finite field (e.g. $\mathbb{F}_{2^{128}}$):

$$P_M(x) = x^n + \sum_{i=0}^{n} M_i x^i .$$

- K interpreted as a finite field element. Hash defined as:

$$H_K(M) = P_M(K).$$

UHF design strategy (1) : provable UHFs based on polynomial evaluation

Principle of UHFs based on polynomial evaluation

- Interpret the message $M = M_0 \ldots M_{n-1}$ as a polynomial in a finite field (e.g. $\mathbb{F}_{2^{128}}$):

$$P_M(x) = x^n + \sum_{i=0}^{n} M_i x^i.$$

- K interpreted as a finite field element. Hash defined as:

$$H_K(M) = P_M(K).$$

- ε-AU property: given $M \neq M'$ of length $n, n' \leq \ell$,

$$|\{K \text{ s.t. } H_K(M) = H_K(M')\}| \leq \max(n, n') \leq \ell.$$

- Examples: GHash, POLYVAL, Poly1305.

UHF design strategy (1) : pros and cons

Pros

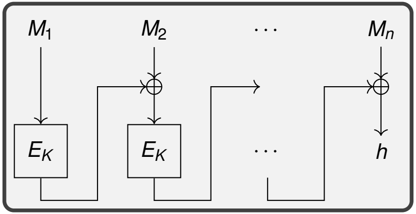
- Provable ε-AU UHF, with simple proofs.
- Efficient implementation : 1 field multiplication per block.

Main drawback: the forbidden attack

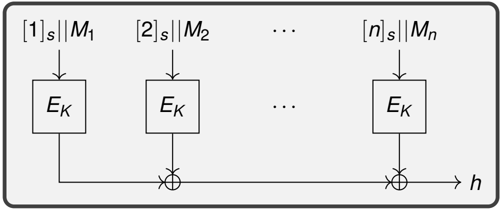
If the attacker knows M, M' s.t. their hashes collide, he can recover K. [Joux, 2004]

- K is such that $P_M(K) = P_{M'}(K)$.
- Finding the root of $P_M - P_{M'}$ costs $\tilde{O}(\text{len}(M) + \text{len}(M'))$ operations.

UHF design strategy (2) : UHFs based on block ciphers



CBC-MAC's UHF [BKR, JCSS'00]



LightMac's UHF [LPTY, FSE'16]

- Other examples: EliHash, PMAC...

UHF design strategy (2) : pros and cons

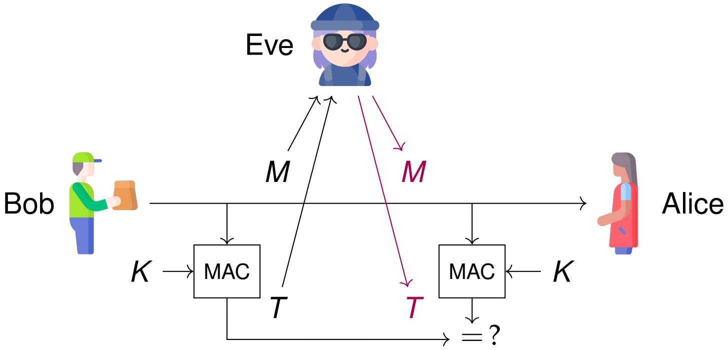
Pros

- Very fast software implementation when instantiated with AES.
- No direct key recovery upon hash collision.

Cons

UHF security sometimes relies on the PRP security of the underlying block cipher.

Message Authentication Codes (MACs)



Requirement: integrity and authenticity

Eve shouldn't be able to modify the messages.

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

MACs : construction from UHFs

MACs are often constructed & proven from Universal Hash Functions (UHFs).

MAC construction from an UHF

- Keys K_0, K_1.
- Block cipher (E_{K_0}), UHF (H_{K_1}).
- Simple construction with $n/2$-bit security (birthday bound):

$$MAC_{K_0,K_1}(M) = E_{K_0}(H_{K_1}(M)).$$

- Possible to add a nonce for beyond-birthday security.

Accordion modes

NIST Launches Development of Cryptographic Accordions

June 06, 2025

A cryptographic accordion is a tweakable block cipher mode that is itself a cipher on variable-length input. NIST proposes to develop three general-purpose accordions:

- Acc128 to support typical usage (birthday bounds) with the Advanced Encryption Standard (AES)
- Acc256 to support typical usage with a 256-bit block cipher (possibly Rijndael-256)
- BBBAcc to support extended usage (beyond-birthday-bound) with AES

In particular, NIST proposes to develop variants of the HCTR2 technique for these accordions.

Goal : develop a new tweakable, provable VIL-SPRP.

Accordion modes: tweakable VIL-SPRP

VIL-SPRP = Variable Input Length - Strong PseudoRandom Permutation

- Constructed from smaller primitives with a proof.
- Can be used in larger constructions (e.g. Authenticated Encryption using encode-then-encipher).

Accordion modes: tweakable VIL-SPRP

VIL-SPRP = Variable Input Length - Strong PseudoRandom Permutation

- Constructed from smaller primitives with a proof.
- Can be used in larger constructions (e.g. Authenticated Encryption using encode-then-encipher).

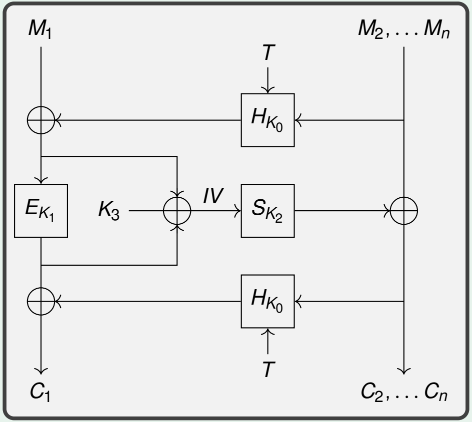
Motivation: limitations of current standardized modes of operation

- In most standardized encryption modes: C_i only depends on $IV, P_1, \ldots, P_i$.
  - Nonce misuse: an attacker can detect if messages begin alike.
- Disk encryption:
  - Ciphertext expansion rarely possible: no integrity.
  - AES-XTS: modifying a ciphertext block changes a single plaintext block.

HCTR2: an accordion mode constructed from an UHF

HCTR2 accordion candidate suggested by the NIST. *[CHB, EPRINT'21']*

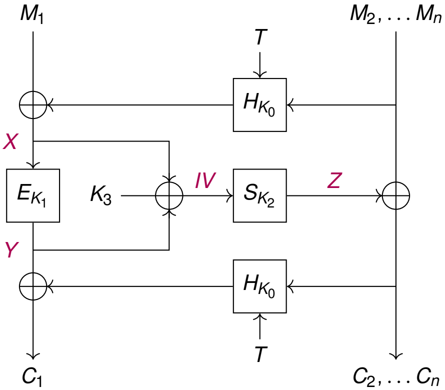
- Hash-encrypt-hash paradigm.
- Message split in 128-bit blocks.
- Block cipher $E_{K_1} = AES_{K_1}$.
- UHF $H_{K_0} = POLYVAL_{K_0}$.
- Stream cipher $S_{K_2} = XCTR_{K_2}$.
- Proven up to birthday bound.



A remark by the NSA on HCTR2 [NIST SP-800-197A Comments]

If (M, T) and (M', T') collide in X ($X = X'$):

- They also collide in Y, IV and Z.

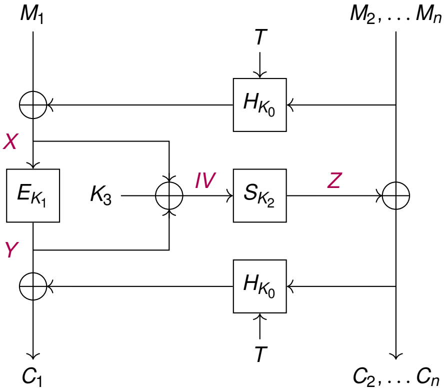


A remark by the NSA on HCTR2 [NIST SP-800-197A Comments]

If (M, T) and (M', T') collide in X ($X = X'$):

- They also collide in Y, IV and Z.
- It can be detected: $\forall i \geq 2, M_i + C_i = M'_i + C'_i$.
- Upon detection:

$$C_1 + H_{K_0}(T, C_2, \ldots C_n) = C'1 + H_{K_0}(T', C'_2, \ldots C'_n).$$

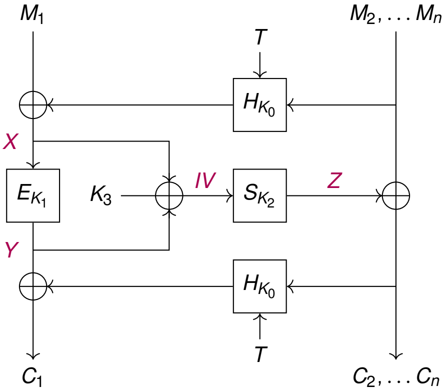


A remark by the NSA on HCTR2 [NIST SP-800-197A Comments]

If (M, T) and (M', T') collide in X ($X = X'$):

- They also collide in Y, IV and Z.
- It can be detected: $\forall i \geq 2, M_i + C_i = M_i' + C_i'$.
- Upon detection:

$$C_1 + H_{K_0}(T, C_2, \ldots C_n) = C'1 + H_{K_0}(T', C_2', \ldots C_n').$$



Birtday-bound attack

- Collision expected after $2^{n/2}$ data.
- Polynomial evaluation H_{K_0}: leaks K_0.
- Possible overcome: use block-cipher based H_{K_0}.

Introduction to Universal Hash Functions and their applications

AES-based UHFs and MACs design paradigm

Our framework of fast AES-based UHFs and MACs

Design of fast AES-based UHFs and MACs

"Fast AES-based Universal Hash Functions and MACs" [BBLPPP, ToSC'24]

- Joint work with Baudrin, Leurent, Pernot, Perrin & Peyrin presented FSE 2025

Overview of our work

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

New primitives

- Two proposed MAC instances: LeMac and PetitMac.

Another view at the definition of UHFs

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:

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

Another view at the definition of UHFs

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:

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

i.e.

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

where $\delta = M + M'$.

Another view at the definition of UHFs

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:

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

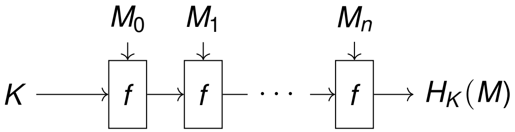
$$\text{i.e.} \qquad \Pr_{K \xleftarrow{\$} \{0,1\}^k}[H_K(M) = H_K(M+\delta)] \leq \varepsilon,$$

where $\delta = M + M'$.

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

Our UHF design strategy

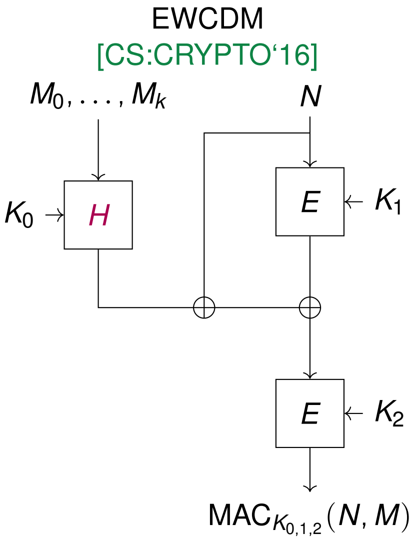
- Exploit AES-NI instructions for software performance.
- Design strategy similar to the round function of Rocca. [SLNKI,FSE'22]



- Heuristic assumptions:
  - ε-AU $\approx$ no high probability differential $\delta \to 0$.
  - 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 H.



Our MAC design strategy

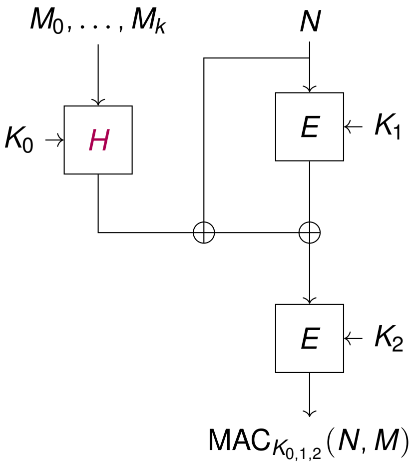
- Use EWCDM to convert a UHF into a MAC.
- Instantiate E with the AES.
- For long messages, the costly part is the UHF H.

Security

- Security proved in $\mathcal{O}(2^n)$.
- With nonce misuse: security proved in $\mathcal{O}(2^{n/2})$.

EWCDM

[CS:CRYPTO'16]



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:

$$SB \to SR \to MC \to AK\,.$$

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

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

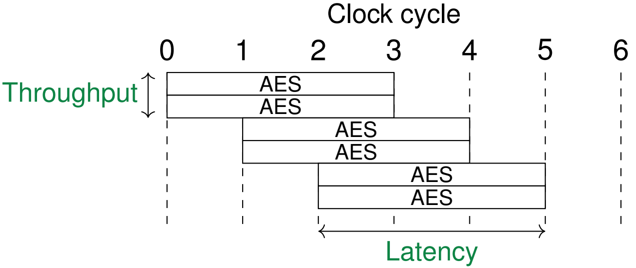
$$\text{rate} = \frac{\#\text{AES-NI instructions}}{\#\text{128-bit message blocks}}$$

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

Scheduling of AES-NI instructions

On modern processors:

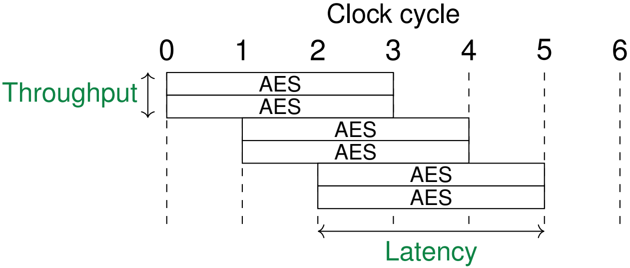
- Throughput: 2 AES per cycle.
- Latency: 3-4 cycles.



Scheduling of AES-NI instructions

On modern processors:

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



Theoretical bound

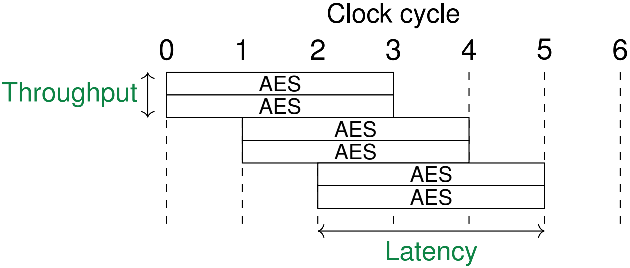
Rate-r constructions require $\geq \frac{r}{2}$ cycles per 128 bits of message.

- 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.

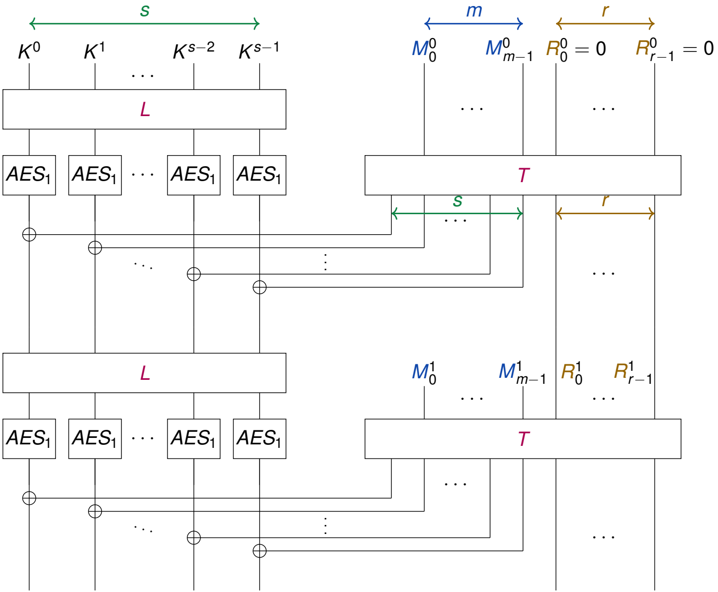
- Observation: existing rate-2 UHFs are slower than this bound (bad parallelization).

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 R.
  - Initialized to zero.
- Sparse linear matrix T.

Main state (left)

- Design similar to a SPN.
- State S of s wires.
  - Initialized to the key K.
- 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.

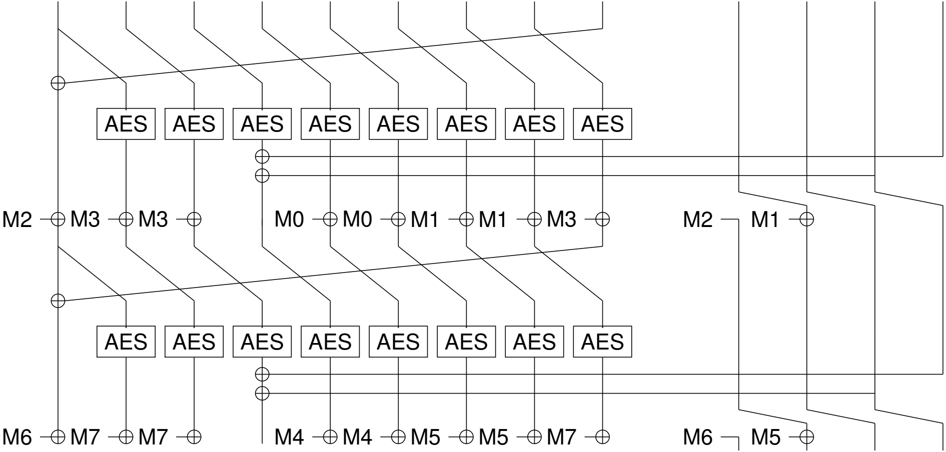
Security check

- 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



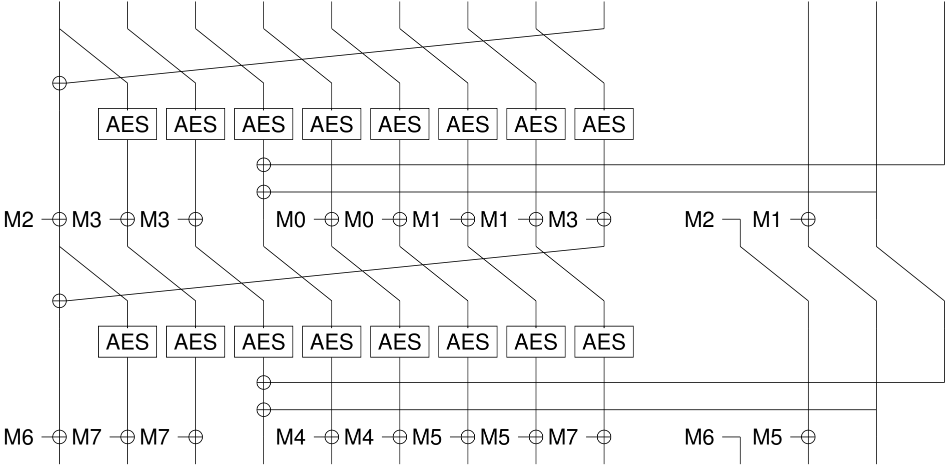
Security

$\geq$ 26 active S-boxes.

Performance

Rate 2 with good parallelization.

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



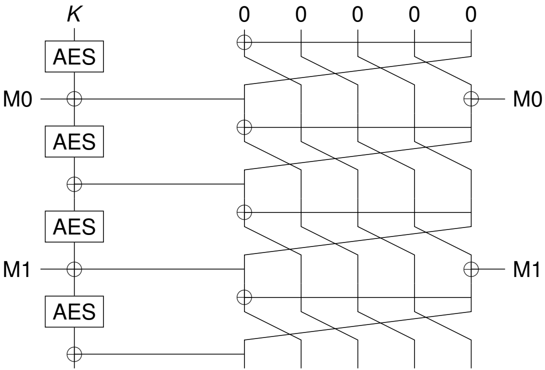
Security

$\geq$ 25 active S-boxes.

Performance

Rate 2 with good parallelization.

Round function of PetitMac's UHF



Security

≥ 26 active S-boxes.

Lightweight

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
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

LeMac's UHF (first row)

PetitMac's UHF (last row)

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

AES-based UHFs:

- Extremely fast on modern processors.
- No direct key recovery upon hash collision detection.

Results:

- Found the first rate-1.75 secure ε-AU UHF candidate.
- Two MAC instantiations with rate-2 : LeMac and PetitMac.

Future works (in the direction of LeMac and PetitMac):

- 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

AES-based UHFs:

- Extremely fast on modern processors.
- No direct key recovery upon hash collision detection.

Results:

- Found the first rate-1.75 secure ε-AU UHF candidate.
- Two MAC instantiations with rate-2 : LeMac and PetitMac.

Future works (in the direction of LeMac and PetitMac):

- 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