IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2025, No. 1, pp. 623–628. DOI:10.46586/tosc.v2025.i1.623-628

Corrigendum to Fast AES-Based Universal Hash Functions and MACs

Augustin Bariant[1,2], Jules Baudrin[1,3], Gaëtan Leurent[1], Clara Pernot[1], Léo Perrin[1] and Thomas Peyrin[4]

[1] Inria, Paris, France
name.surname@inria.fr
[2] ANSSI, Paris, France
[3] UCLouvain, Louvain-La-Neuve, Belgium
jules.baudrin@uclouvain.be
[4] Nanyang Technological University, Singapore, Singapore
thomas.peyrin@ntu.edu.sg

Abstract. In ToSC 2024(2), Bariant *et al.* proposed a new framework for designing efficient AES-based Universal Hash Functions (UHFs) and Message Authentification Codes (MACs). They proposed two MAC instances aiming for 128-bit security, PetitMac and LeMac, based on two different UHF candidates. The security of the UHF candidates was evaluated with Mixed Integer Linear Programing (MILP) modeling, to find the minimum number of active S-boxes in differential trails from a non-zero message difference to a zero state difference. The designers claimed at least 26 active S-boxes for the UHF of LeMac.

In this corrigendum, we point out that there was a mistake when writing the LeMac specification from the MILP model. The UHF candidate of LeMac presented in the paper does not correspond to the construction analysed with the MILP solver. In particular, the erroneous candidate only guarantees 25 active S-boxes rather than 26. Therefore, we propose to rename the candidate from the original paper to LeMac-0, and propose a fixed version of LeMac, with the correct underlying UHF candidate. The change of specification of LeMac is motivated by the fact that the new specification possesses better security guarantees than LeMac-0 for similar performances.

Keywords: Universal hash function · MAC · AES · authentication · LeMac

1 Introduction

In [BBL+24], the authors introduced a framework of fast AES-based UHFs, exploiting the speed of AES-NI with maximum throughput with parallel AES rounds. In this framework, the performance and the security of the candidates can be automatically evaluated. On the one hand, the performance is measured by generating a C implementation, compiling, and benchmarking in an automatic manner. On the other hand, the security of the candidates is analyzed by computing the minimum number of active S-boxes in any differential trail with non-zero message difference leading to a collision in output, using MILP modeling. In Section 6 of [BBL+24], Bariant *et al.* presented a strategy to generate UHF candidates which are both secure and performant. Following an automated search, they proposed two concrete MAC candidates, LeMac and PetitMac, using the EWCDM construction [CS16] to convert a UHF into a MAC.

We found out that the original paper [BBL+24] wrongfully depicted the UHF candidate that was found during the search (Figure 2, Figure 4, and Algorithm 2 of [BBL+24]). The errors are consistent throughout the paper and the corresponding candidate also belongs

Received: 2024-11-22 Accepted: 2025-01-23 Published: 2025-03-07

$$L = \begin{bmatrix} 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&1&0 \\ 0&0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 1&0&0&0&0&0&0&0&0 \end{bmatrix} \quad T = \left[\begin{array}{ccc|ccc} 0&0&0&0&0&1&0&0&0 \\ 1&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ \hline 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&1&0&0&0&0&0 \end{array}\right] \quad A = (110111111) \quad r = 3 \quad s = 9 \quad m = 4 \ .$$

(a) Framework parameters of the UHF used in LeMac-0.

$$L = \begin{bmatrix} 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&1&0 \\ 0&0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 1&0&0&0&0&0&0&0&0 \end{bmatrix} \quad T = \left[\begin{array}{ccc|ccc} 0&0&0&0&0&1&0&0&0 \\ 1&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0 \\ \hline 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 0&0&1&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&1&0&0&0&0&0 \end{array}\right] \quad A = (110111111) \quad r = 4 \quad s = 9 \quad m = 4 \ .$$

(b) Framework parameters of the UHF used in LeMac.

Figure 1: Comparison of framework parameters of LeMac and LeMac-0.

to the framework, but with different security guarantees. The mistake came from a wrong reconstruction of the specification from the MILP parameters, where the message schedule is shifted by one round. We propose to rename the erroneous MAC of the original paper into LeMac-0, and propose new specifications for LeMac, actually corresponding to the best candidate identified during the search. This error does not affect the other candidates of [BBL+24, Table 2], whose specification, performance and security are correct.

2 LeMac versus LeMac-0

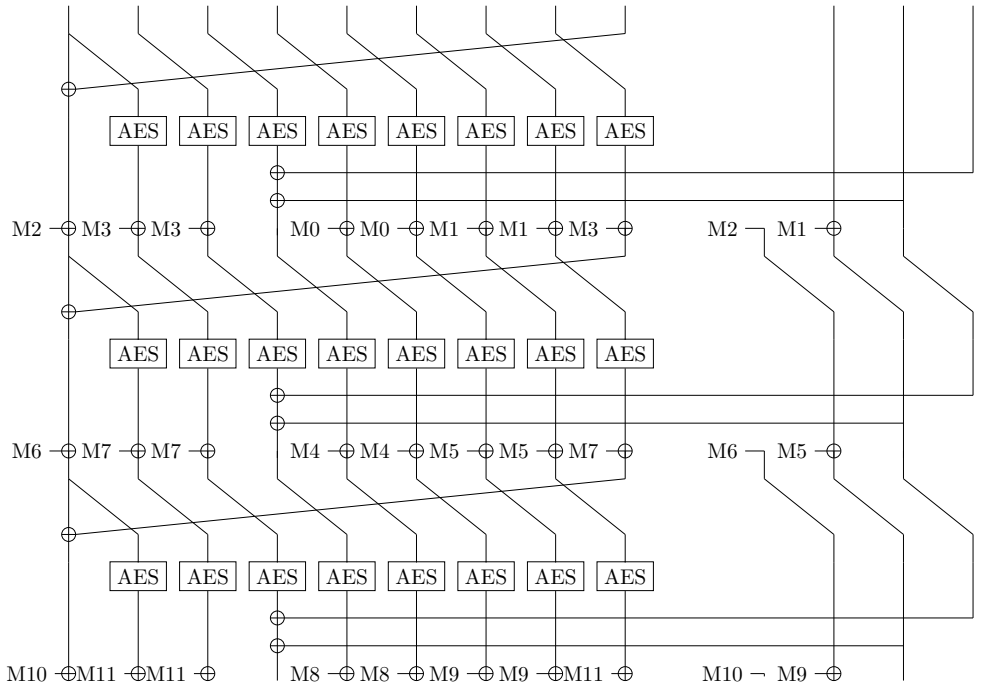
The parameters of the best MILP model identified during the parameter search explained in [BBL+24] are shown in Figure 1b, with the notations of the framework from [BBL+24]. The corresponding specification of LeMac is given in Figure 2b and Algorithm 2. For comparison, the erroneous specification of LeMac-0 is given in Figure 2a and Algorithm 1.

The difference lies in the message schedule: the message addition was wrongly performed before the wire rotation in LeMac-0 (in particular, this allows to reduce the memory size r in the message schedule from 4 to 3). In the true LeMac however, the message addition is performed after the wire rotation.

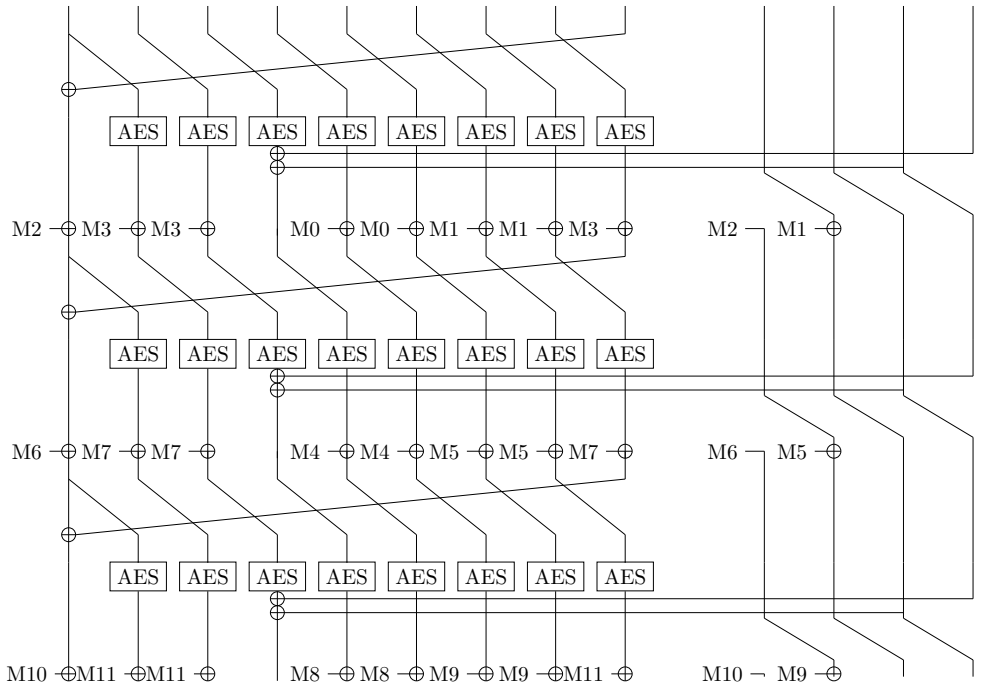
Security of LeMac-0. While LeMac-0 is not the result of the parameter search in the original paper, it also belongs to the framework, with the parameters of Figure 1a. In particular, we can use the original MILP model to study its security. We found that LeMac-0 guarantees at least 25 active S-boxes, while LeMac guarantees 26 active S-boxes. Moreover, when removing constraints based on linear incompatibilities in the MILP model, LeMac still guarantees 24 active S-boxes while LeMac-0 only guarantees 20. Although the trails with 24/20 active S-boxes on LeMac/LeMac-0 are not instantiable because of linear incompatibilities, the bound of 20 for LeMac-0 lead us to reconsider our claims. Therefore, we claim 128-bit security on LeMac, but do not claim 128-bit security anymore on LeMac-0 because the security margin seems too low. However, we do not know of any attack on LeMac-0; we still expect it to be secure and consider it as an interesting candidate to study, as it can give deeper insights on the security of LeMac.

Performance of LeMac. The reference implementation[1] has been updated to match the correct specification. The benchmark results in Table 4 of the original paper measure

[1] https://github.com/AugustinBariant/Implementations_LeMac_PetitMac



(a) The UHF candidate of LeMac-0.



(b) The UHF candidate of LeMac.

Figure 2: Comparison of the round functions of LeMac and LeMac-0.
The state wires were permuted for more readability, compared to the framework candidates presented in Figure 1, according to the permutation {2,3,4,5,0,8,1,6,7}.

Algorithm 1 LeMac-0.

▷ Key derivation
$\mathsf{Kinit} \leftarrow (\mathsf{AES}_K(0), ...\mathsf{AES}_K(8))$
$\mathsf{Kfinal} \leftarrow (\mathsf{AES}_K(9), ...\mathsf{AES}_K(26))$
$k_2 \leftarrow \mathsf{AES}_K(27)$
$k_3 \leftarrow \mathsf{AES}_K(28)$
▷ UHF
$X^0 \leftarrow \mathsf{Kinit}$
$R^0 \leftarrow (0, 0, 0)$
▷ Padding
$\ell \leftarrow \lceil (\text{bitlen}(M) + 1)/512 \rceil$
$M_0, ..., M_{4\ell-1} \leftarrow M \| 10^*$
$M_{4\ell}, ..., M_{4\ell+11} \leftarrow 0$
for all $0 \leqslant i < \ell + 3$ **do**
  $X_0^{i+1} \leftarrow X_0^i + X_8^i$
  for all $1 \leqslant j < 9$ **do**
    $X_j^{i+1} \leftarrow A(X_{j-1}^i)$
  end for
  $X_0^{i+1} \leftarrow X_0^{i+1} + M_{4i+2}$
  $X_1^{i+1} \leftarrow X_1^{i+1} + M_{4i+3}$
  $X_2^{i+1} \leftarrow X_2^{i+1} + M_{4i+3}$
  $X_3^{i+1} \leftarrow X_3^{i+1} + R_1^i + R_2^i$
  $X_4^{i+1} \leftarrow X_4^{i+1} + M_{4i}$
  $X_5^{i+1} \leftarrow X_5^{i+1} + M_{4i}$
  $X_6^{i+1} \leftarrow X_6^{i+1} + M_{4i+1}$
  $X_7^{i+1} \leftarrow X_7^{i+1} + M_{4i+1}$
  $X_8^{i+1} \leftarrow X_8^{i+1} + M_{4i+3}$
  $R_0^{i+1} \leftarrow M_{4i+2}$
  $R_1^{i+1} \leftarrow R_0^i + M_{4i+1}$
  $R_2^{i+1} \leftarrow R_1^i$
end for
for all $0 \leqslant j < 9$ **do**
  for all $0 \leqslant i < 10$ **do**
    $X_j^{\ell+3+i+1} \leftarrow A\left(X_j^{\ell+3+i} + \mathsf{Kfinal}_{i+j}\right)$
  end for
end for
$h \leftarrow \sum_{j=0}^{8} X_j^{\ell+13}$
▷ EWCDM
return $\mathsf{AES}_{k_3}\left(h + \mathsf{AES}_{k_2}(N) + N\right)$

Algorithm 2 LeMac.

▷ Key derivation
$\mathsf{Kinit} \leftarrow (\mathsf{AES}_K(0), ...\mathsf{AES}_K(8))$
$\mathsf{Kfinal} \leftarrow (\mathsf{AES}_K(9), ...\mathsf{AES}_K(26))$
$k_2 \leftarrow \mathsf{AES}_K(27)$
$k_3 \leftarrow \mathsf{AES}_K(28)$
▷ UHF
$X^0 \leftarrow \mathsf{Kinit}$
$R^0 \leftarrow (0, 0, 0, 0)$
▷ Padding
$\ell \leftarrow \lceil (\text{bitlen}(M) + 1)/512 \rceil$
$M_0, ..., M_{4\ell-1} \leftarrow M \| 10^*$
$M_{4\ell}, ..., M_{4\ell+15} \leftarrow 0$
for all $0 \leqslant i < \ell + 4$ **do**
  $X_0^{i+1} \leftarrow X_0^i + X_8^i$
  for all $1 \leqslant j < 9$ **do**
    $X_j^{i+1} \leftarrow A(X_{j-1}^i)$
  end for
  $X_0^{i+1} \leftarrow X_0^{i+1} + M_{4i+2}$
  $X_1^{i+1} \leftarrow X_1^{i+1} + M_{4i+3}$
  $X_2^{i+1} \leftarrow X_2^{i+1} + M_{4i+3}$
  $X_3^{i+1} \leftarrow X_3^{i+1} + R_2^i + R_3^i$
  $X_4^{i+1} \leftarrow X_4^{i+1} + M_{4i}$
  $X_5^{i+1} \leftarrow X_5^{i+1} + M_{4i}$
  $X_6^{i+1} \leftarrow X_6^{i+1} + M_{4i+1}$
  $X_7^{i+1} \leftarrow X_7^{i+1} + M_{4i+1}$
  $X_8^{i+1} \leftarrow X_8^{i+1} + M_{4i+3}$
  $R_0^{i+1} \leftarrow M_{4i+2}$
  $R_1^{i+1} \leftarrow R_0^i + M_{4i+1}$
  $R_2^{i+1} \leftarrow R_1^i$
  $R_3^{i+1} \leftarrow R_2^i$
end for
for all $0 \leqslant j < 9$ **do**
  for all $0 \leqslant i < 10$ **do**
    $X_j^{\ell+3+i+1} \leftarrow A\left(X_j^{\ell+3+i} + \mathsf{Kfinal}_{i+j}\right)$
  end for
end for
$h \leftarrow \sum_{j=0}^{8} X_j^{\ell+13}$
▷ EWCDM
return $\mathsf{AES}_{k_3}\left(h + \mathsf{AES}_{k_2}(N) + N\right)$

the performance of LeMac-0 and not of LeMac, but the performance differences between LeMac and LeMac-0 are marginal.

Acknowledgments

The authors would like to thank Quentin Rataud for pointing out this mistake. This work was supported by the bilateral NRF/ANR grant SELECT (NRF-NRFI08-2022-0013/ANR-20-CE48-0017). Thomas Peyrin is supported by the Singapore NRF Investigatorship grant (NRF-NRFI08-2022-0013). Léo Perrin is supported by the European Research Council (ERC, grant agreement no. 101041545 "ReSCALE"). This work was also supported by project Cryptanalyse from PEPR Cybersécurité (22-PECY-0010).

Table 1: Benchmark results.

		Speed (c/B)		
CPU	Cipher	1kB	16kB	256kB
Intel Haswell (Xeon E5-2630 v3)	GCM (AD only)	1.175	0.716	0.637
	Rocca (AD only) 0.604	0.225	0.201	
	Rocca-S (AD only)	0.656	0.291	0.269
	AEGIS128 (AD only)	0.818	0.579	0.564
	AEGIS128L (AD only)	0.560	0.304	0.288
	Tiaoxin-346 v2 (AD only)	0.494	0.209	0.194
	Jean-Nikolić	0.451	0.147	0.136
	LeMac-0	0.448	0.145	0.137
	LeMac	0.460	0.146	0.128
	PetitMac	1.125	0.891	0.876
Intel Skylake (Xeon Gold 6130)	GCM (AD only)	0.883	0.448	0.414
	Rocca (AD only)	0.503	0.187	0.166
	Rocca-S (AD only)	0.497	0.210	0.190
	AEGIS128 (AD only)	0.668	0.472	0.458
	AEGIS128L (AD only)	0.491	0.266	0.252
	Tiaoxin-346 v2 (AD only)	0.476	0.207	0.189
	Jean-Nikolić	0.395	0.142	0.126
	LeMac-0	0.372	0.141	0.126
	LeMac	0.383	0.141	0.126
	PetitMac	0.799	0.636	0.626
Intel Ice Lake (Xeon Gold 5320)	GCM (AD only)	0.737	0.345	0.321
	Rocca (AD only)	0.438	0.167	0.149
	Rocca-S (AD only)	0.392	0.165	0.151
	AEGIS128 (AD only)	0.586	0.401	0.389
	AEGIS128L (AD only)	0.393	0.207	0.195
	Tiaoxin-346 v2 (AD only)	0.346	0.134	0.123
	Jean-Nikolić	0.298	0.137	0.110
	LeMac-0	0.274	0.083	0.074
	LeMac	0.285	0.092	0.079
	PetitMac	0.522	0.384	0.376
AMD Zen1 (EPYC 7301)	GCM (AD only)	0.932	0.567	0.538
	Rocca (AD only)	0.424	0.140	0.122
	Rocca-S (AD only)	0.438	0.159	0.142
	AEGIS128 (AD only)	0.509	0.325	0.376
	AEGIS128L (AD only)	0.373	0.193	0.181
	Tiaoxin-346 v2 (AD only)	0.354	0.145	0.131
	Jean-Nikolić	0.339	0.127	0.111
	LeMac-0	0.298	0.089	0.075
	LeMac	0.319	0.097	0.083
	PetitMac	0.683	0.511	0.501
AMD Zen3 (EPYC 7513)	GCM (AD only)	0.816	0.479	0.466
	Rocca (AD only)	0.392	0.140	0.124
	Rocca-S (AD only)	0.413	0.158	0.141
	AEGIS128 (AD only)	0.498	0.338	0.329
	AEGIS128L (AD only)	0.358	0.183	0.174
	Tiaoxin-346 v2 (AD only)	0.311	0.120	0.109
	Jean-Nikolić	0.301	0.111	0.098
	LeMac-0	0.270	0.082	0.070
	LeMac	0.272	0.085	0.069
	PetitMac	0.669	0.511	0.501

References

[BBL+24] Augustin Bariant, Jules Baudrin, Gaëtan Leurent, Clara Pernot, Léo Perrin, and Thomas Peyrin. Fast AES-based universal hash functions and MACs featuring LeMac and PetitMac. *IACR Trans. Symm. Cryptol.*, 2024(2):35–67, 2024.

[CS16] Benoît Cogliati and Yannick Seurin. EWCDM: An efficient, beyond-birthday secure, nonce-misuse resistant MAC. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 121–149, August 2016.