Radboud University

EliMAC: Speeding Up LightMAC by Around 20%

Christoph Dobraunig, Bart Mennink, Samuel Neves

FSE 2024

March 25, 2024

ESCADA

Introduction

- Using key K, message M is signed with tag T
- Verification takes K and (M, T) and outputs $\begin{cases} \top \text{ if tag is correct} \\ \bot \text{ if tag is incorrect} \end{cases}$

Message Authentication Codes

$$M \xrightarrow[*]{} \boxed{\mathrm{MAC}_K} \xrightarrow[t]{} T$$

- Using key K, message M is signed with tag T
- Verification takes K and (M, T) and outputs $\begin{cases} \top \text{ if tag is correct} \\ \bot \text{ if tag is incorrect} \end{cases}$
- Security goal: unforgeability

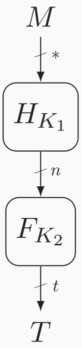
Message Authentication Codes

- Using key K, message M is signed with tag T
- Verification takes K and (M, T) and outputs $\begin{cases} \top \text{ if tag is correct} \\ \bot \text{ if tag is incorrect} \end{cases}$
- Security goal: unforgeability
- Often, one adopts a stronger notion: PRF security
  - MAC_K should behave like a random function
  - $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{MAC}}(q)$ should be small

Hash-then-PRF

Idea

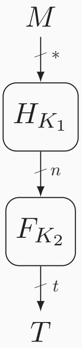
- Process arbitrary length M through "weaker" universal hash H_{K_1}
- Protect with "stronger" F_{K_2}



Hash-then-PRF

Idea

- Process arbitrary length M through "weaker" universal hash H_{K_1}
- Protect with "stronger" F_{K_2}



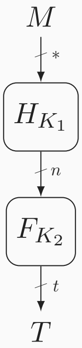
Security

- Secure MAC function if
  - H is ϵ-universal hash function
  - F is pseudorandom function
- F can be replaced by (truncation of) block cipher E at some loss
- Extra message block can be entered after H if it is ϵ-XOR-universal

Hash-then-PRF

Idea

- Process arbitrary length M through "weaker" universal hash H_{K_1}
- Protect with "stronger" F_{K_2}

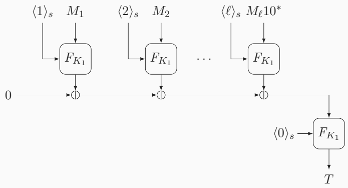


Security

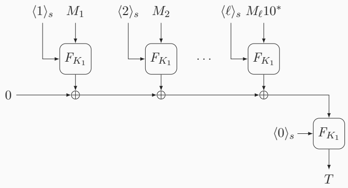
- Secure MAC function if
  - H is ϵ-universal hash function
  - F is pseudorandom function
- F can be replaced by (truncation of) block cipher E at some loss
- Extra message block can be entered after H if it is ϵ-XOR-universal

Ideally: H_{K_1} is parallelizable

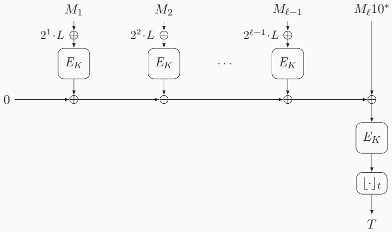
Protected counter sum [Ber99]



Protected counter sum [Ber99]

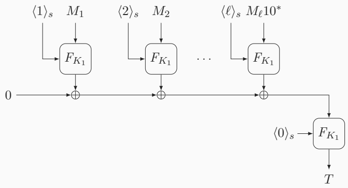


PMAC [BR02] $(L = E_K(0^n))$

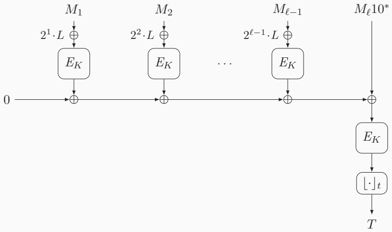


Examples of Parallelizable Hash-then-PRF Designs

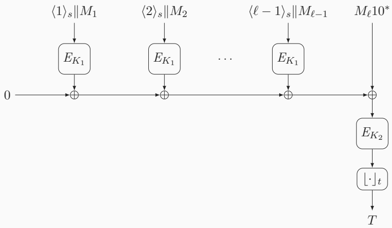
Protected counter sum [Ber99]



PMAC [BR02] $(L = E_K(0^n))$



LightMAC [LPTY16]



“Expensive” Parallelizable Universal Hashing

- Parallelizable universal hashing is almost always built from block cipher
- Intuitively, it should be possible to build it from universal hashes

“Expensive” Parallelizable Universal Hashing

- Parallelizable universal hashing is almost always built from block cipher
- Intuitively, it should be possible to build it from universal hashes

Goal: parallelizable domain extender for universal hashing

"Expensive" Parallelizable Universal Hashing

- Parallelizable universal hashing is almost always built from block cipher
- Intuitively, it should be possible to build it from universal hashes

Goal: parallelizable domain extender for universal hashing

1. EliHash: fully parallelizable universal hash from fixed-length hashes
2. EliMAC: MAC function on top of EliHash
3. Instantiation of EliMAC using round-reduced AES
4. Side-result: flaws in earlier attempt Marvin [SBB$^+$09, SB12]

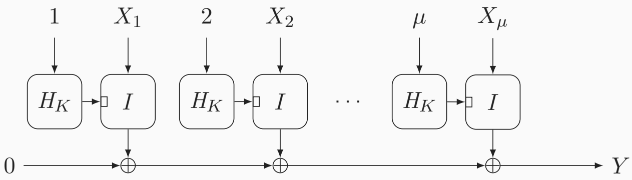
EliHash and EliMAC

Building Blocks

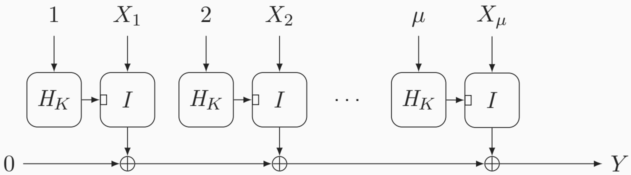
- Two – not necessarily independent – families of hash functions:
  - $H : \mathcal{K}' \times [1, \ldots, \mu] \to \mathcal{K}$
  - $I : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$

Design

- EliHash : $\mathcal{K}' \times \mathcal{X}^{[1\ldots\mu]} \to \mathcal{Y}$ is defined as



EliHash: Security (1/2)



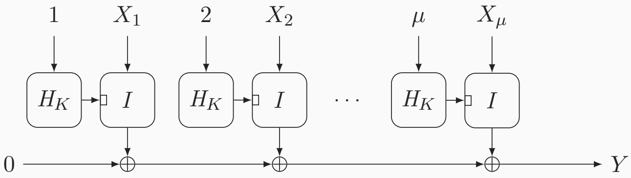
ε-XOR-universality

- For any distinct X, X' and any Y: $\mathbf{Pr}_K\left(\mathcal{H}_K(X) \oplus \mathcal{H}_K(X') = Y\right) \leq \varepsilon$

Goal

- XOR-universality of EliHash as long as H and I satisfy certain conditions

EliHash: Security (1/2)



ε-XOR-universality

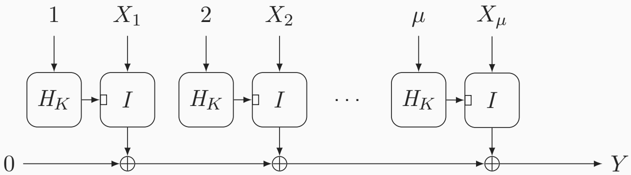
- For any distinct X, X' and any Y: $\mathbf{Pr}_K\left(\mathcal{H}_K(X) \oplus \mathcal{H}_K(X') = Y\right) \le \varepsilon$

Goal

- XOR-universality of EliHash as long as H and I satisfy certain conditions

Technical Complications

- Ideally, we rely on XOR-universality of H and I



ε-XOR-universality

- For any distinct X, X' and any Y: $\mathbf{Pr}_K\left(\mathcal{H}_K(X) \oplus \mathcal{H}_K(X') = Y\right) \leq \varepsilon$

δ-μ-independence

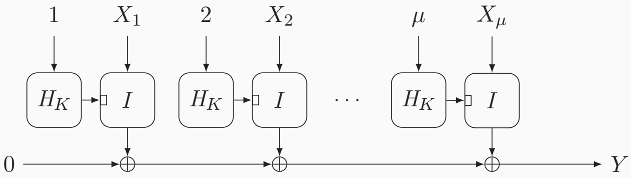
- For any distinct X_i and any Y_i: $\mathbf{Pr}_K\left(\forall_{i=1}^{\mu}\ \mathcal{H}_K(X_i) = Y_i\right) \leq \delta^\mu$

Goal

- XOR-universality of EliHash as long as H and I satisfy certain conditions

Technical Complications

- Ideally, we rely on XOR-universality of H and I
- Typically more than two evaluations of I are XORed
- We will have to rely on slightly stronger property of H: μ-independence



ε-XOR-universality

- For any distinct X, X' and any Y:
  $\mathbf{Pr}_K\left(\mathcal{H}_K(X) \oplus \mathcal{H}_K(X') = Y\right) \leq \varepsilon$

δ-μ-independence

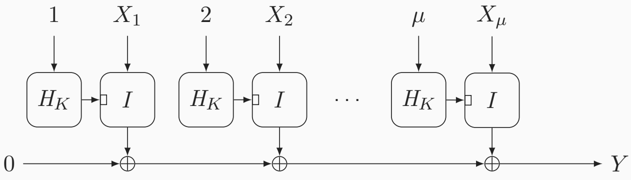
- For any distinct X_i and any Y_i:
  $\mathbf{Pr}_K\left(\forall_{i=1}^{\mu}\ \mathcal{H}_K(X_i) = Y_i\right) \leq \delta^\mu$

Goal

- XOR-universality of EliHash as long as H and I satisfy certain conditions

Technical Complications

- Ideally, we rely on XOR-universality of H and I
- Typically more than two evaluations of I are XORed
- We will have to rely on slightly stronger property of H: μ-independence
- Okay in our case as H has very small domain: $[1, \ldots, \mu]$



ε-XOR-universality

- For any distinct X, X' and any Y: $\mathbf{Pr}_K\left(\mathcal{H}_K(X) \oplus \mathcal{H}_K(X') = Y\right) \leq \varepsilon$

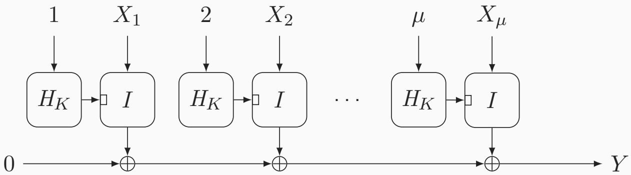
δ-μ-independence

- For any distinct X_i and any Y_i: $\mathbf{Pr}_K\left(\forall_{i=1}^{\mu}\ \mathcal{H}_K(X_i) = Y_i\right) \leq \delta^\mu$

General Result

- Let $\mu \in \mathbb{N}$ be maximal message length to EliHash
- Let $H : \mathcal{K}' \times [1, \ldots, \mu] \to \mathcal{K}$ be δ-μ-independent
- Let $I : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be ε-XOR-universal

EliHash: Security (2/3)



ε-XOR-universality

- For any distinct X, X' and any Y: $\mathbf{Pr}_K\left(\mathcal{H}_K(X) \oplus \mathcal{H}_K(X') = Y\right) \leq \varepsilon$

δ-μ-independence

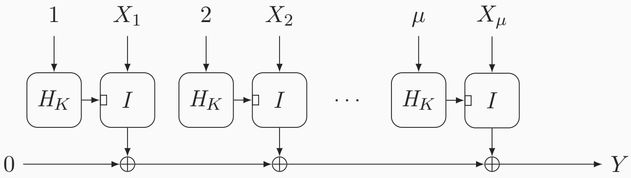
- For any distinct X_i and any Y_i: $\mathbf{Pr}_K\left(\forall_{i=1}^{\mu}\ \mathcal{H}_K(X_i) = Y_i\right) \leq \delta^\mu$

General Result

- Let $\mu \in \mathbb{N}$ be maximal message length to EliHash
- Let $H : \mathcal{K}' \times [1, \ldots, \mu] \to \mathcal{K}$ be δ-μ-independent
- Let $I : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be ε-XOR-universal

- Then, EliHash : $\mathcal{K}' \times \mathcal{X}^{[1 \ldots \mu]} \to \mathcal{Y}$ is $(|\mathcal{K}|\delta)^\mu \varepsilon$-XOR-universal

EliHash: Security (3/3)



ε-XOR-universality

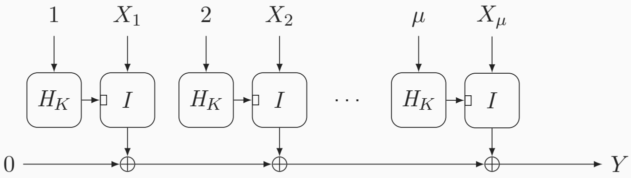
- For any distinct X, X' and any Y: $\mathbf{Pr}_K \left(\mathcal{H}_K(X) \oplus \mathcal{H}_K(X') = Y \right) \leq \varepsilon$

δ-μ-independence

- For any distinct X_i and any Y_i: $\mathbf{Pr}_K \left(\forall_{i=1}^{\mu}\ \mathcal{H}_K(X_i) = Y_i \right) \leq \delta^\mu$

Rough Proof Idea

- Consider any distinct $\boldsymbol{X} = (X_1, \ldots, X_\ell)$, $\boldsymbol{X}' = (X'_1, \ldots, X'_{\ell'})$ and any Y
- We have to upper bound $\mathbf{Pr}_K \left(\text{EliHash}_K(\boldsymbol{X}) \oplus \text{EliHash}_K(\boldsymbol{X}') = Y \right)$



ε-XOR-universality

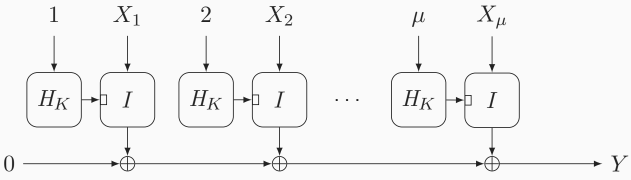
- For any distinct X, X' and any Y: $\mathbf{Pr}_K\left(\mathcal{H}_K(X) \oplus \mathcal{H}_K(X') = Y\right) \leq \varepsilon$

δ-μ-independence

- For any distinct X_i and any Y_i: $\mathbf{Pr}_K\left(\forall_{i=1}^{\mu}\, \mathcal{H}_K(X_i) = Y_i\right) \leq \delta^\mu$

Rough Proof Idea

- Consider any distinct $\boldsymbol{X} = (X_1, \ldots, X_\ell)$, $\boldsymbol{X}' = (X'_1, \ldots, X'_{\ell'})$ and any Y
- We have to upper bound $\mathbf{Pr}_K\left(\text{EliHash}_K(\boldsymbol{X}) \oplus \text{EliHash}_K(\boldsymbol{X}') = Y\right)$
  1. Count the number of key tuples to I that fulfill a XOR-collision



ε-XOR-universality

- For any distinct X, X' and any Y: $\mathbf{Pr}_K\left(\mathcal{H}_K(X) \oplus \mathcal{H}_K(X') = Y\right) \le \varepsilon$

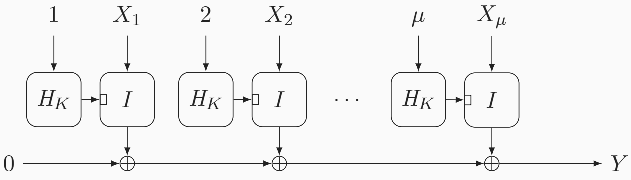
δ-μ-independence

- For any distinct X_i and any Y_i: $\mathbf{Pr}_K\left(\forall_{i=1}^{\mu}\ \mathcal{H}_K(X_i) = Y_i\right) \le \delta^\mu$

Rough Proof Idea

- Consider any distinct $\boldsymbol{X} = (X_1, \ldots, X_\ell)$, $\boldsymbol{X}' = (X'_1, \ldots, X'_{\ell'})$ and any Y
- We have to upper bound $\mathbf{Pr}_K\left(\text{EliHash}_K(\boldsymbol{X}) \oplus \text{EliHash}_K(\boldsymbol{X}') = Y\right)$
  1. Count the number of key tuples to I that fulfill a XOR-collision
  2. Bound the probability that H hits one of these key tuples: $\le \delta^\mu$

EliHash: Security (3/3)



ε-XOR-universality

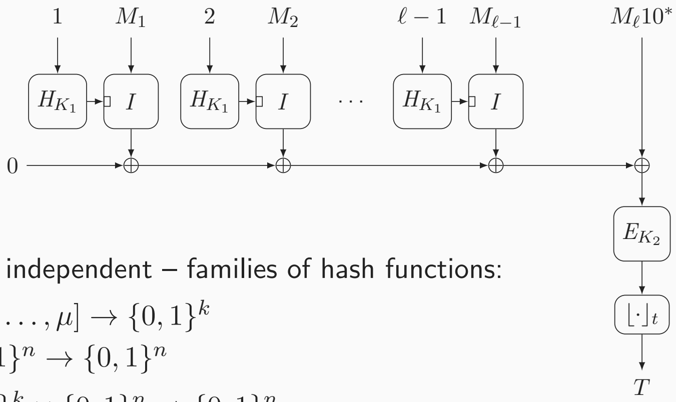
- For any distinct X, X' and any Y: $\mathbf{Pr}_K\left(\mathcal{H}_K(X) \oplus \mathcal{H}_K(X') = Y\right) \leq \varepsilon$

δ-μ-independence

- For any distinct X_i and any Y_i: $\mathbf{Pr}_K\left(\forall_{i=1}^{\mu}\ \mathcal{H}_K(X_i) = Y_i\right) \leq \delta^\mu$

Rough Proof Idea

- Consider any distinct $\boldsymbol{X} = (X_1, \ldots, X_\ell)$, $\boldsymbol{X}' = (X'_1, \ldots, X'_{\ell'})$ and any Y
- We have to upper bound $\mathbf{Pr}_K\left(\text{EliHash}_K(\boldsymbol{X}) \oplus \text{EliHash}_K(\boldsymbol{X}') = Y\right)$
  1. Count the number of key tuples to I that fulfill a XOR-collision
  2. Bound the probability that H hits one of these key tuples: $\leq \delta^\mu$
  3. Bound the number of possible key tuples: $\leq (|\mathcal{K}|)^\mu \varepsilon$



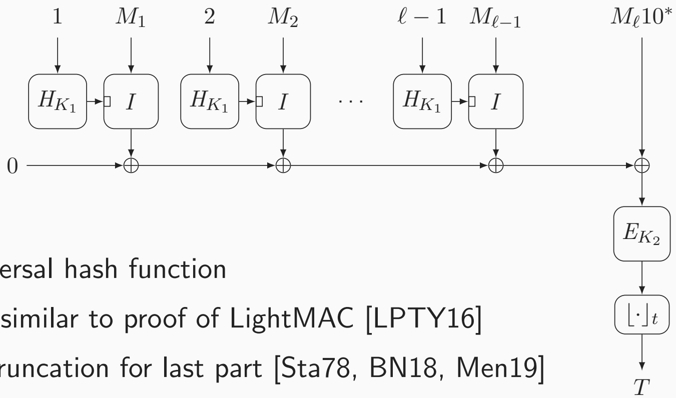
Building Blocks

- Two – not necessarily independent – families of hash functions:
  - $H : \{0,1\}^{k'} \times [1,\ldots,\mu] \to \{0,1\}^k$
  - $I : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$
- Block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$

Design

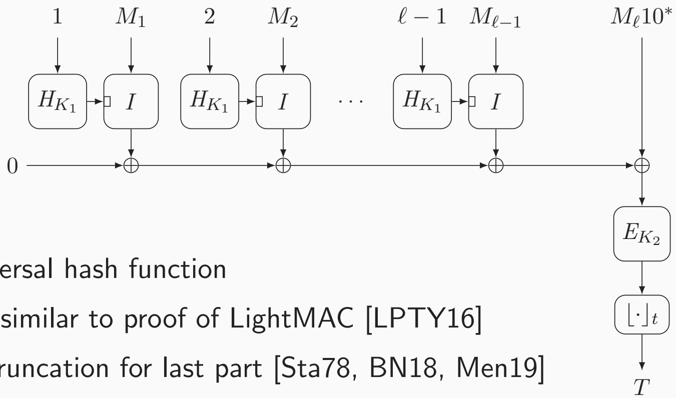
- EliMAC $\approx$ LightMAC but with hashing part replaced by EliHash

EliMAC: Security



Security Proof

- Views EliHash as universal hash function
- Composition to MAC similar to proof of LightMAC [LPTY16]
- Relies on security of truncation for last part [Sta78, BN18, Men19]



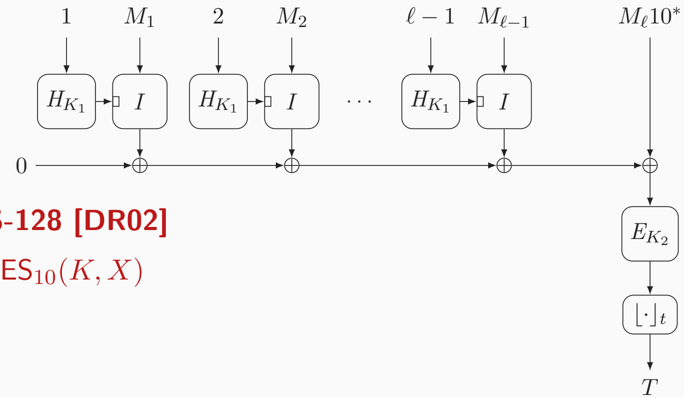
Security Proof

- Views EliHash as universal hash function
- Composition to MAC similar to proof of LightMAC [LPTY16]
- Relies on security of truncation for last part [Sta78, BN18, Men19]

Tightness

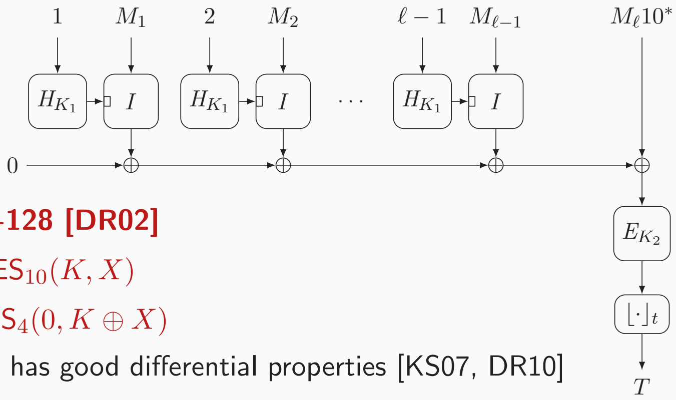
- Matching attacks given in paper

Instantiation



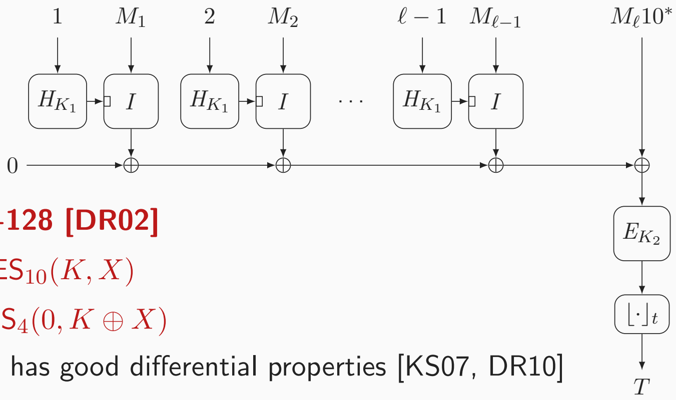
Instantiation Using AES-128 [DR02]

- Instantiation of E: $\mathsf{AES}_{10}(K, X)$



Instantiation Using AES-128 [DR02]

- Instantiation of E: AES$_{10}(K, X)$
- Instantiation of I: AES$_4(0, K \oplus X)$
  - 4-round AES-128 has good differential properties [KS07, DR10]



Instantiation Using AES-128 [DR02]

- Instantiation of E: $\text{AES}_{10}(K, X)$
- Instantiation of I: $\text{AES}_4(0, K \oplus X)$
  - 4-round AES-128 has good differential properties [KS07, DR10]
- Instantiation of H: $\text{AES}_7(K, \langle i\rangle_{32} \| \langle i\rangle_{32} \| \langle i\rangle_{32} \| \langle i\rangle_{32})$
  - μ-independence does not follow from XOR-universality
  - It appears that 7 rounds suffice [DFJ13] for $\mu \leq 2^{32}$

Theoretical Comparison

scheme	# AES rounds for ℓ blocks		
	pre	online	total
LightMAC	0	10ℓ	10ℓ
EliMAC	$7(\ell-1)$	$4(\ell-1)+10$	$11\ell-1$

- Here, $\ell \leq \mu = 2^{32}$, and counter values encoded using $s = 32$ bits
- EliMAC invokes slightly more AES rounds than LightMAC

Theoretical Comparison

scheme	# AES rounds for ℓ blocks			bit length of ℓ-block message
	pre	online	total	
LightMAC	0	10ℓ	10ℓ	$96\ell - 1$
EliMAC	$7(\ell - 1)$	$4(\ell - 1) + 10$	$11\ell - 1$	$128\ell - 1$

- Here, $\ell \leq \mu = 2^{32}$, and counter values encoded using $s = 32$ bits
- EliMAC invokes slightly more AES rounds than LightMAC
- However, it can process more message bits per block $\rightarrow$ improvement of $\approx$ 20%

Theoretical Comparison

scheme	# AES rounds for ℓ blocks			bit length of ℓ-block message
	pre	online	total	
LightMAC	0	10ℓ	10ℓ	$96\ell - 1$
EliMAC	$7(\ell - 1)$	$4(\ell - 1) + 10$	$11\ell - 1$	$128\ell - 1$

- Here, $\ell \leq \mu = 2^{32}$, and counter values encoded using $s = 32$ bits
- EliMAC invokes slightly more AES rounds than LightMAC
- However, it can process more message bits per block $\rightarrow$ improvement of $\approx$ 20%
- Precomputation can speed up EliMAC significantly

Theoretical Comparison

scheme	# AES rounds for ℓ blocks			bit length of ℓ-block message
	pre	online	total	
LightMAC	0	10ℓ	10ℓ	$96\ell - 1$
EliMAC	$7(\ell - 1)$	$4(\ell - 1) + 10$	$11\ell - 1$	$128\ell - 1$

- Here, $\ell \leq \mu = 2^{32}$, and counter values encoded using $s = 32$ bits
- EliMAC invokes slightly more AES rounds than LightMAC
- However, it can process more message bits per block $\rightarrow$ improvement of $\approx$ 20%
- Precomputation can speed up EliMAC significantly

- Note: difference in
  - assumptions (on round-reduced AES for EliMAC) and
  - generic security bounds (64-bit versus 56-bit)

Comparison

- EliMAC-AES: default and with key precomputation
- Comparison with:
  - LightMAC-AES [LPTY16]
  - PMAC2-AES [CCJN21]
  - ZMAC-Deoxys-TBC-256 [IMPS17, JNPS21]
- All parallelizable and with length independent bounds
- Cpb when authenticating $64/1536/4096$ byte messages

		64	1536	4096
Ivy Bridge	LightMAC	3.43	1.13	1.11
	EliMAC	2.18	1.02	0.98
	EliMAC p.c.	2.00	0.46	0.43
	PMAC2	4.50	1.28	1.22
	ZMAC	5.70	1.49	1.26
Broadwell	LightMAC	8.75	0.98	1.08
	EliMAC	1.94	0.76	0.74
	EliMAC p.c.	1.75	0.30	0.27
	PMAC2	3.25	1.13	1.09
	ZMAC	6.97	1.34	1.23
Skylake	LightMAC	2.53	0.86	0.85
	EliMAC	1.56	0.70	0.69
	EliMAC p.c.	1.31	0.27	0.26
	PMAC2	1.71	0.67	0.64
	ZMAC	4.64	0.91	0.84
Zen 2	LightMAC	2.18	0.58	0.58
	EliMAC	1.31	0.45	0.42
	EliMAC p.c.	0.87	0.14	0.13
	PMAC2	1.31	0.58	0.56
	ZMAC	4.34	0.88	0.81

Comparison

- EliMAC-AES: default and with key precomputation
- Comparison with:
  - LightMAC-AES [LPTY16]
  - PMAC2-AES [CCJN21]
  - ZMAC-Deoxys-TBC-256 [IMPS17, JNPS21]
- All parallelizable and with length independent bounds
- Cpb when authenticating $64/1536/4096$ byte messages

Notes

- Security assumptions and bounds differ
- Key precomputation ("EliMAC p.c.") is much faster but comes with added memory requirements

		64	1536	4096
Ivy Bridge	LightMAC	3.43	1.13	1.11
	EliMAC	2.18	1.02	0.98
	EliMAC p.c.	2.00	0.46	0.43
	PMAC2	4.50	1.28	1.22
	ZMAC	5.70	1.49	1.26
Broadwell	LightMAC	8.75	0.98	1.08
	EliMAC	1.94	0.76	0.74
	EliMAC p.c.	1.75	0.30	0.27
	PMAC2	3.25	1.13	1.09
	ZMAC	6.97	1.34	1.23
Skylake	LightMAC	2.53	0.86	0.85
	EliMAC	1.56	0.70	0.69
	EliMAC p.c.	1.31	0.27	0.26
	PMAC2	1.71	0.67	0.64
	ZMAC	4.64	0.91	0.84
Zen 2	LightMAC	2.18	0.58	0.58
	EliMAC	1.31	0.45	0.42
	EliMAC p.c.	0.87	0.14	0.13
	PMAC2	1.31	0.58	0.56
	ZMAC	4.34	0.88	0.81

Conclusion

Conclusion

EliHash and EliMAC

- Domain extender for universal hashing and corresponding MAC
- Underlying hashes must be μ-independent and XOR-universal
- Potentially significant speed-up but under different assumptions

Conclusion

EliHash and EliMAC

- Domain extender for universal hashing and corresponding MAC
- Underlying hashes must be μ-independent and XOR-universal
- Potentially significant speed-up but under different assumptions

Future Research

- Avoiding μ-independence?
- Purely parallelizable universal hash function extender

Conclusion

EliHash and EliMAC

- Domain extender for universal hashing and corresponding MAC
- Underlying hashes must be μ-independent and XOR-universal
- Potentially significant speed-up but under different assumptions

Future Research

- Avoiding μ-independence?
- Purely parallelizable universal hash function extender

Thank you for your attention!

Daniel J. Bernstein.
How to Stretch Random Functions: The Security of Protected Counter Sums.
J. Cryptology, 12(3):185–192, 1999.

Srimanta Bhattacharya and Mridul Nandi.
A note on the chi-square method: A tool for proving cryptographic security.
Cryptography and Communications, 10(5):935–957, 2018.

John Black and Phillip Rogaway.
A Block-Cipher Mode of Operation for Parallelizable Message Authentication.
In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 384–397. Springer, 2002.

Bishwajit Chakraborty, Soumya Chattopadhyay, Ashwin Jha, and Mridul Nandi.
On Length Independent Security Bounds for the PMAC Family.
IACR Trans. Symmetric Cryptol., 2021(2):423–445, 2021.

Patrick Derbez, Pierre-Alain Fouque, and Jérémy Jean.
Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting.
In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 371–387. Springer, 2013.

Joan Daemen and Vincent Rijmen.
The Design of Rijndael: AES - The Advanced Encryption Standard.
Information Security and Cryptography. Springer, 2002.

Joan Daemen and Vincent Rijmen.
Refinements of the ALRED construction and MAC security claims.
IET Information Security, 4(3):149–157, 2010.

Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin.
ZMAC: A Fast Tweakable Block Cipher Mode for Highly Secure Message Authentication.
In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 34–65. Springer, 2017.

Jérémy Jean, Ivica Nikolic, Thomas Peyrin, and Yannick Seurin.
The Deoxys AEAD Family.
J. Cryptol., 34(3):31, 2021.

Liam Keliher and Jiayuan Sui.
Exact maximum expected differential and linear probability for two-round Advanced Encryption Standard.
IET Information Security, 1(2):53–57, 2007.

Atul Luykx, Bart Preneel, Elmar Tischhauser, and Kan Yasuda.
A MAC Mode for Lightweight Block Ciphers.
In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 43–59. Springer, 2016.

Bart Mennink.
Linking Stam's Bounds with Generalized Truncation.
In Mitsuru Matsui, editor, *Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings*, volume 11405 of *Lecture Notes in Computer Science*, pages 313–329. Springer, 2019.

- Marcos A. Simplício Jr. and Paulo S. L. M. Barreto.
  Revisiting the Security of the ALRED Design and Two of Its Variants: Marvin and LetterSoup.
  IEEE Trans. Inf. Theory, 58(9):6223–6238, 2012.

- Marcos A. Simplício Jr., Pedro d'Aquino F. F. S. Barbuda, Paulo S. L. M. Barreto, Tereza Cristina M. B. Carvalho, and Cintia B. Margi.
  The MARVIN message authentication code and the LETTERSOUP authenticated encryption scheme.
  Security and Communication Networks, 2(2):165–180, 2009.

- A. J. Stam.
  Distance between sampling with and without replacement.
  Statistica Neerlandica, 32(2):81–91, 1978.