Comparison of Multi-Purpose Cores of Keccak and AES

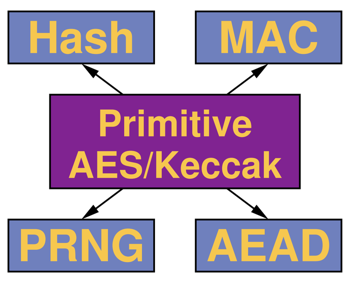
Panasayya Yalla, Ekawat Homsirikamol, Jens-Peter Kaps
Department of Electrical and Computer Engineering, George Mason University, Fairfax, Virginia 22030, USA

Abstract

Most widely used security protocols, Internet Protocol Security (IPSec), Secure Socket Layer (SSL), and Transport Layer Security (TLS), provide several cryptographic services which in turn require multiple dedicated cryptographic algorithms. A single cryptographic primitive for all secret key functions utilizing different mode of operations can overcome this constraint. This paper investigates the possibility of using AES and Keccak as the underlying primitives for high-speed and resource constrained applications.

Introduction and Motivation

1. **Integrity** is provided through a **Hash** Function.
2. **Authentication** and integrity are provided by a **Message Authentication Code (MAC)**.
3. **Confidentiality**, integrity, and authentication are provided simultaneously by **Authenticated Encryption (AE)**.
4. **Pseudo Random Numbers** required for secret keys and nonces are generated using **Pseudo Random Number Generators (PRNG)**.



- A single secret key algorithm such as AES could provide several of these services through application of various modes of operation.
- Another interesting option is using Keccak's f-permutation.

Cryptographic Algorithms

AES

- NIST standard for block ciphers.
- Based on Rijndael block cipher.
- 128-bit block size.
- 128/192/256-bit key size.

Keccak-$p[1600, n_r]$

- Permutation based on Keccak, winner of competition for next Secure Hash Algorithm (SHA-3).
- 1600-bit state size.

AES Modes

Hash→AES-Hash
- Based on Davies-Meyer.
- The message enters on the input for the key.
- Uses a block size of 256-bit (Rijndael-256).
- Not parallelizable.
- Not a NIST standardized mode.

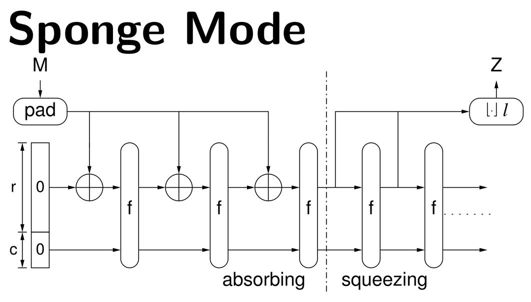
MAC→CMAC
- Equivalent to One-Key CBC-MAC (OMAC1).
- Recommended mode of operation by NIST.
- Uses two additional subkeys generated from original key.
- Not parallelizable.

AEAD→GCM (Galois/Counter Mode)
- It is a combination of counter mode with Galois field multiplication.
- Recommended mode by NIST.
- Parallelizable.
- Widely used for its efficiency and performance.

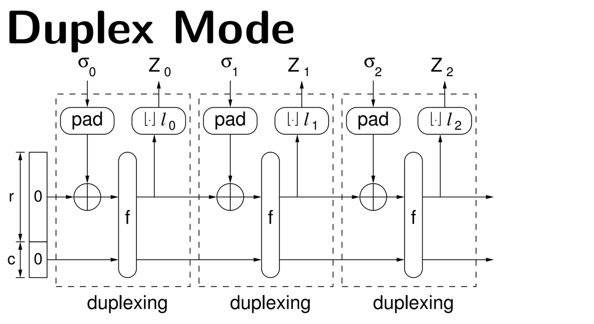
PRNG→Fortuna
- It is counter mode with 32-bit counter.
- Cryptographically secure PRNG.
- Not a NIST standardized mode.
- The seed is processed as key.
- Parallelizable.

Keccak Modes

Sponge Mode



Duplex Mode



Hash→Keccak
- r=1088, c=512, 24 rounds.
- $|P_{MS}(M)| = n \cdot 1088$

MAC→Sponge
- r=1088, c=512, 24 rounds.
- KeyPack is used to encode the secret key in a uniform way.
- $|P_{MS}(M)| = n \cdot 1088$
- $|P_{MS}(KeyPack\|IV)| = 1088$

Padding
P_{MS}: Padding for message in Sponge Mode
P_{MK}: Message padding for Keyak
P_{SD}: Padding for seed in PRNG Mode

AEAD → Keyak
- r=1348, c=252, 12 rounds.
- CAESAR candidate.
- $|P_{MK}(M_i\|3)| = 1348, \forall\, i \neq n-1;$ $|P_{MK}(M_{n-1}\|1)| = 1348$

PRNG→Duplex
- r=1344, c=252, 12 rounds.
- Random bits are generated using the seed as input.
- Additional random bits are generated using empty seed as input.
- $|P_{SD}(Seed)| = |P_{SD}(0)| = 1348$

Modes of Operation Summary

AES / Rijndael* and Keccak Modes (Rd. = Number of rounds)

	Operation	Mode	Block	Key	Rd.	ρ	Inputs	Outputs
AES	Hash*	AES-Hash	256	N/A	14		$\|M\|, M$	H
	MAC	CMAC	128	128	10		$\|M\|, M, K, IV$	T
	AEAD	GCM	128	128	10		$\|M\|, M, K, IV, \|AD\|, AD$	T, C
	PRNG	Fortuna	128	N/A	14		S	R
Keccak	Hash	Sponge	1600	N/A	24	1088	$\|M\|, M$	H
	MAC	Sponge	1600	128	24	1088	$\|M\|, M, K, IV$	T
	AEAD	Duplex	1600	128	12	1344	$\|M\|, M, K, IV, \|AD\|, AD$	T, C
	PRNG	Duplex	1600	N/A	12	1344	S	R

M–Message, K–Key, AD–Associated Data, S–Seed, IV–Initialization Value, H–Hash, T–Tag, C–Cipher-text, R–Random Number, $|X|$–Length of X

Design Decisions

- One high speed (HS) and one low-area (LA) all-in-one design each.
- All-in-one supports Hash, MAC, AEAD, and PRNG.
- One HS and one LA dedicated AES-GCM and Keyak design each.
- HS design of Keccak uses full width datapath of 1600 bits.
- HS design of AES uses 2 cores of AES-128/256 that can be combined to a single Rijndael with 256 block size.
- LA design AES 32-bit datapath (width of MixColumns).
- LA design Keccak 64-bit (width of a word in Keccak).
- All padding is performed in hardware.

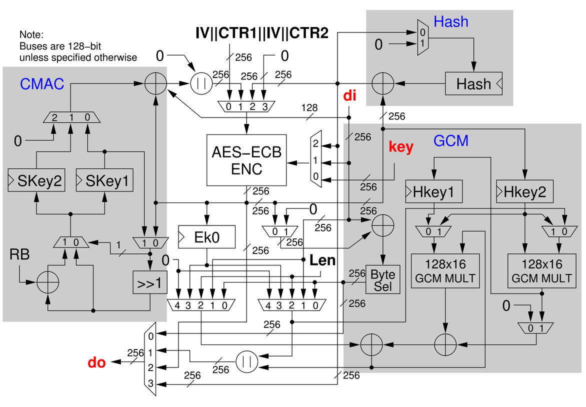
High-Speed Implementations

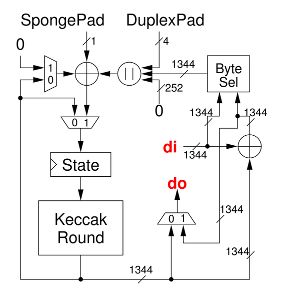
AES

- Contains two AES-128 cores that can support key sizes of 128- and 256-bit.
- These cores also have the capability to form a single Rijndael-256 core for AES-Hash.
- Two 128x16 Galois field multipliers are used in AES-GCM mode.
- Dedicated AES-GCM design contains only one AES-128 bit.

Keccak

- Based on single round iterative architecture.
- DuplexPad signal enables injection of input into the state via an XOR.
- SinglePad provides padding to injected input.
- Due to non-uniform block sizes for Keccak modes, the controller for loading module is complex.





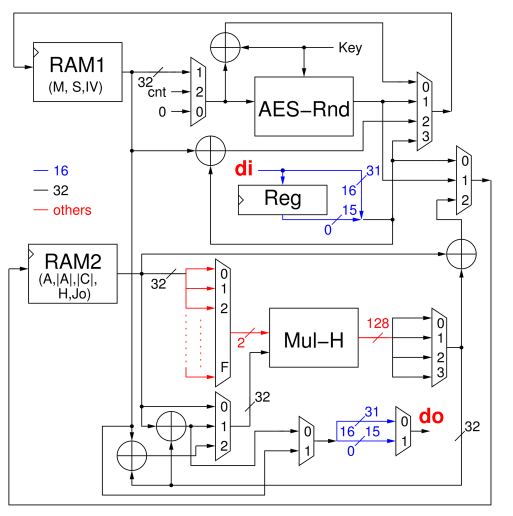
Low-Area Implementations

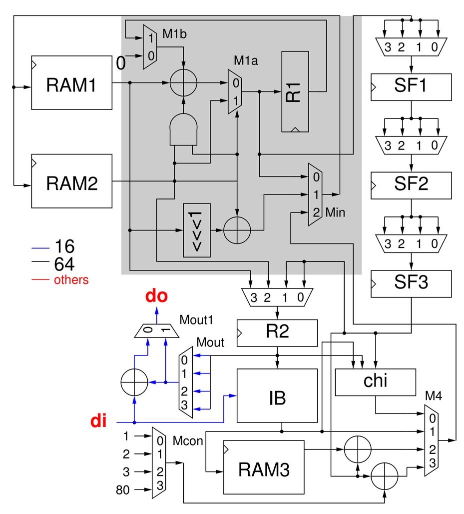
AES

- Two 32-bit wide dual-port RAMs are used to store inputs and state variables.
- It takes 4 clock cycles for one round of AES.
- Multiplications in AES-GCM mode are performed using a 128x2 multiplier.
- Dedicated AES-GCM design is a reduced version of multi-purpose core.

Keccak

- Two 64-bit wide dual-port RAMs are used to store state variables.
- Additionally a single-port RAM is for storing Key, IV, and Seed.
- It takes 58 clock cycles for one round of Keccak.
- Only 6-bits of each round constant are stored to conserve space.
- Dedicated Keyak is derived from the multi-purpose Keccak with minimal changes.





Implementation Results and Comparisons

Comparison of our designs with other implementations on Xilinx Virtex-5 (TW = This Work)

	Mode	Design	Area (Slices)	Freq. (MHz)	TP (Gbps)	TP/Area (Mbps/Slices)
High-Speed	Hash	Multi-AES[TW]	2871	203.29	3.470	1.208
		Multi-Keccak[TW]	2805	163.92	7.431	2.649
		Keccak[1]	1395	281.84	12.777	9.16
	MAC	Multi-AES[TW]	2871	203.29	2.366	0.824
		Multi-Keccak[TW]	2805	163.92	7.431	2.649
		GMAC[2]	9405	120.17	15.382	1.636
	Dedicated AEAD	AES-GCM[TW]	1089	283.53	3.299	3.030
		AES-GCM[3]	678	335.00	2.250	3.319
		AES-CCM[4]	490	274.00	1.525	3.112
		Grøestl/AES[5]	3102	233.00	3.848	1.240
		Keyak[TW]	2357	243.96	27.324	11.593
Low-Area	Hash	Multi-AES[TW]	478	131.23	0.262	0.549
		Multi-Keccak[TW]	318	257.00	0.211	0.665
		Keccak[6]	275	251.25	0.118	0.430
		Keccak[7]	393	159.0	0.864	2.198
	Dedicated AEAD	AES-GCM[TW]	351	130.87	0.116	0.331
		AES-GCM[3]	247	393.00	0.230	0.931
		AES-CCM[4]	214	272.00	0.363	1.696
		Keyak[TW]	259	281.29	0.506	1.954

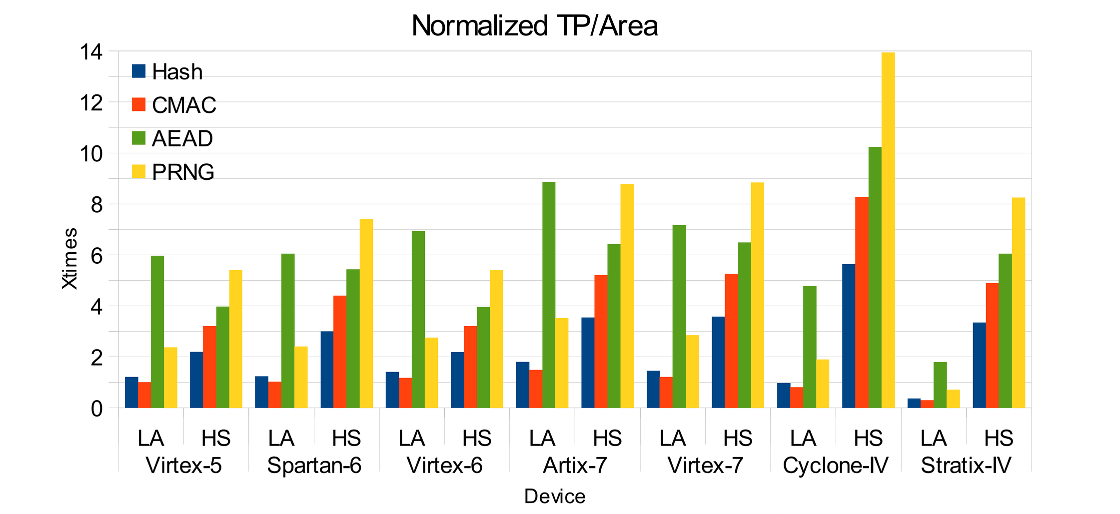
- All comparisons are made with dedicated design from literature.
- Our high-speed multi-Keccak performance degrades by more than 1/3 compared to dedicated Keccak [1] due to additional hardware for modes and padding.
- High-speed multi-AES trails behind GMAC[2], however multi-Keccak can compete in MAC mode.
- Our dedicated high-speed AES-GCM TP/Area closely matches those of single message AEAD designs [3] and [4].
- Keyak outperforms [5] by a factor of 9 and other designs by almost a factor of 4 due to larger block size with similar number of rounds.
- Even our low-area Keccak suffers performance degradation due to additional hardware for modes.
- Implementation details of [3] and [4] are unknown.

Comparison of AES vs. Keccak

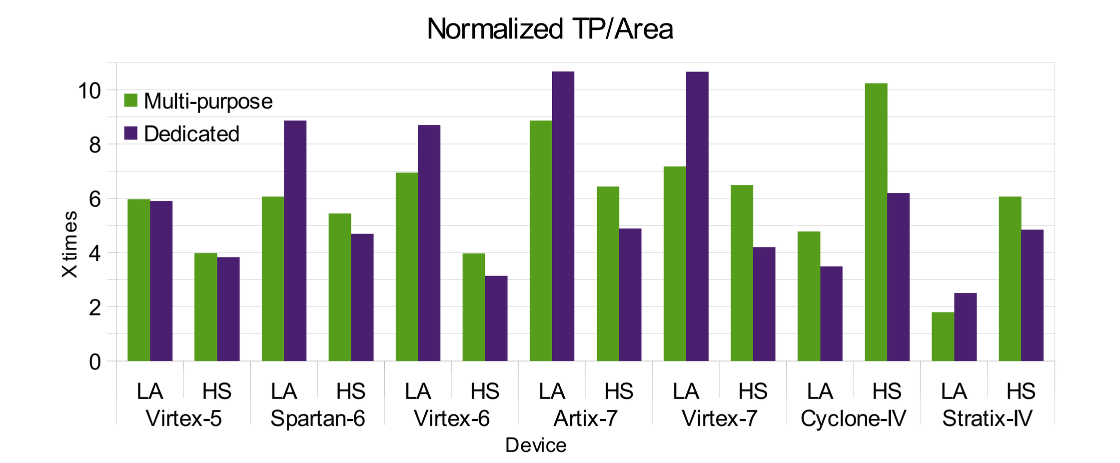
Results of AES and Keccak Implementations

Mode	Design	Area (Slices)	Freq. (MHz)	TP (Gbps)	TP/Area (Mbps/Slices
High-Speed Designs on Xilinx Virtex-7 FPGA					
Hash	Multi-AES	3061	188.18	3.212	1.049
	Multi-Keccak	2495	206.70	9.370	3.756
MAC	Multi-AES	3061	188.18	2.190	0.715
	Multi-Keccak	2495	206.70	9.370	3.756
AEAD	Multi-AES	3061	188.18	4.380	1.431
	Multi-Keccak	2495	206.70	**23.150**	9.279
PRNG	AES-PRNG	3061	188.18	3.212	1.049
	Multi-Keccak	2495	206.70	23.150	9.279
Dedicated AEAD	AES-GCM	1455	352.98	4.107	2.823
	Keyak	2444	258.40	**28.941**	11.841
Low-Area Designs on Xilinx Artix-7 FPGA					
Hash	Multi-AES	629	82.83	0.166	0.263
	Multi-Keccak	264	152.23	0.125	0.474
MAC	Multi-AES	629	82.83	0.189	0.301
	Multi-Keccak	264	152.23	0.119	0.451
AEAD	Multi-AES	629	82.83	0.074	0.117
	Multi-Keccak	264	152.23	0.274	1.037
PRNG	Multi-AES	629	82.83	0.379	0.602
	Multi-Keccak	264	152.23	0.280	1.060
Dedicated AEAD	AES-GCM	548	71.09	0.630	0.115
	Keyak	260	177.87	0.136	1.231

- Multi-Keccak implementations always outperform multi-AES implementations. In some cases up to 14 times.
  - High-speed Keccak datapath width is 12.5 times wider than AES.
  - Low-area Keccak datapath width is 2 times wider than AES.
  - This increase in width is not translated into increase in area.
  - Adding mode to Keccak requires minimal additional resources whereas AES modes have vastly different underlying characteristics.
- Keyak outperforms AES-GCM in a similar way as multi-Keccak outperforms multi-AES.
  - Transforming Keccak into Keyak requires only minimal additional hardware.
  - Transforming AES into AES-GCM adds costly GCM multipliers and increases number of clock cycles.
- The performance of Keyak relative to AES-GCM is higher than multi-Keccak in AEAD mode relative to multi-AES in the same more for low-area designs.
  - The is not true for high-speed design as multi-AES employs two cores but not AES-GCM.



Performance improvement of multi-Keccak over multi-AES for specific modes of operation

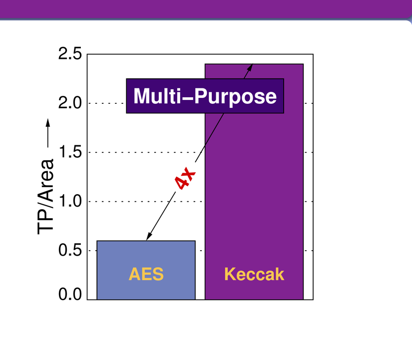


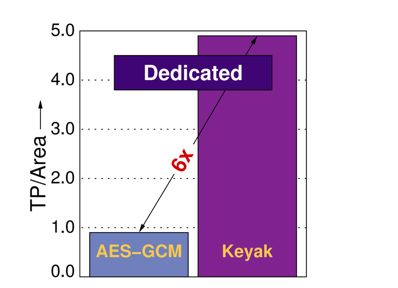
Performance improvement of dedicated and multi- purpose Keccak over corresponding AES cores for AEAD

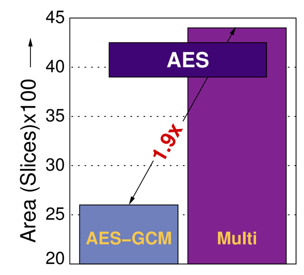
Conclusions

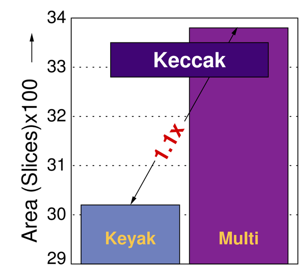
- Overall, our Multi-Keccak design outperforms our Multi-AES design by a factor of 4 on average across all functions and FPGAs in terms of throughput over area.
- Keccak in AE-mode (Keyak) achieves a TP of 23.2 Gbps on Xilinx Virtex-7 and 28.7 Gbps on Altera Stratix-IV.
- Dedicated Keyak outperforms AES-GCM by a factor of 6 on average across all devices.

⇒Keccak is more flexible than AES.









- Adding modes to AES-GCM increases the area by a factor of 1.9 and for Keyak only by 1.1.

References

[1] E. Homsirikamol, M. Rogawski, and K. Gaj, "Throughput vs area trade-offs architectures of five round 3 SHA-3 candidates implemented using Xilinx and Altera FPGAs," in *CHES*, ser. LNCS, B. Preneel and T. Takagi, Eds., vol. 6917. Springer, Sep 2011, pp. 491–506.

[2] Y. Lu, G. Shou, Y. Hu, and Z. Guo, "The research and efficient fpga implementation of ghash core for gmac," in *E-Business and Information System Security, 2009. EBISS '09*, 2009, pp. 1–5.

[3] *AES-GCM Core family for Xilinx FPGA*, Helion Technology, Fulbourn, Cambridge CB21 5DQ, England, 2011, http://www.heliontech.com/downloads/aes_gcm_8bit_xilinx_datasheet.pdf.

[4] *AES-CCM Core family for Xilinx FPGA*, Helion Technology Limited, Fulbourn, Cambridge CB21 5DQ, England, 2011, http://www.heliontech.com/downloads/Helion_PB_-_AES-CCM_8-bit_FPGA.pdf.

[5] M. Rogawski, "Development and Benchmarking of New Hardware Architectures for Emerging Cryptographic Transformations," Ph.D. dissertation, George Mason University, July 2013.

[6] J.-P. Kaps, P. Yalla, K. K. Surapathi, B. Habib, S. Vadlamudi, and S. Gurung, "Lightweight implementations of SHA-3 finalists on FPGAs," Mar 2012, third SHA-3 Candidate Conference.

[7] B. Jungk and J. Apfelbeck, "Area-efficient FPGA implementations of the SHA-3 finalists," in *International Conference on ReConfigurable Computing and FPGAs*. IEEE: ReConfig'11, DEC 2011.