Radbound University

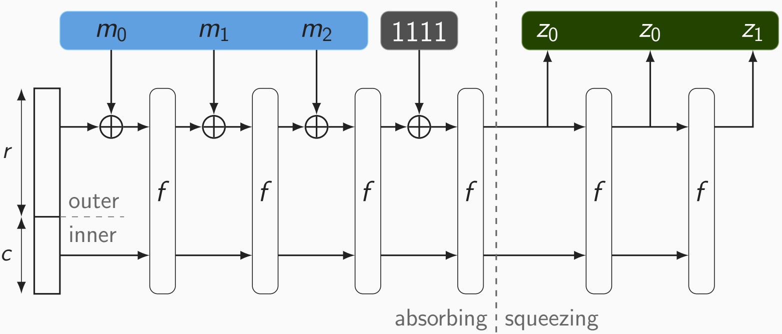
KangarooTwelve
draft-viguier-kangarootwelve-01

Benoît Viguier[1]

CFRG Meeting, March 19, 2018

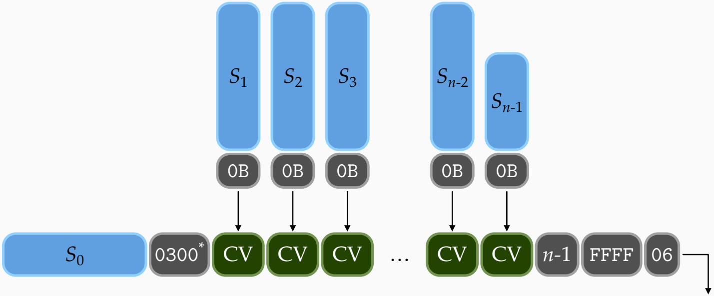
[1]Radboud University, Nijmegen, The Netherlands

What is KangarooTwelve?



- SHAKE128
  - eXtendable Output Function
  - Sponge construction
  - Uses Keccak-$p[1600, n_\mathrm{r} = 24]$
  - BUT no parallelism

What is KANGAROOTWELVE?



- KangarooTwelve
  - eXtendable Output Function
  - Tree on top of sponge construction
  - KECCAK-p reduced from 24 to 12 rounds
  - Parallelism grows automatically with input size
  - No penalty for short messages

How secure is KangarooTwelve?

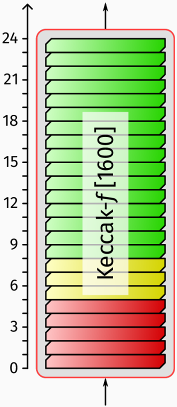
- Same security claim as SHAKE128: 128 bits of security
- Sponge generic security

  [EuroCrypt 2008] – On the Indifferentiability of the Sponge Construction
- Parallel mode with proven generic security

  [IJIS 2014] – Sufficient conditions for sound tree and sequential hashing modes
  [ACNS 2014] – Sakura: A Flexible Coding for Tree Hashing
- Sponge function on top of Keccak-$p[1600, n_\mathrm{r} = 12]$
  - Round function unchanged

    $\Rightarrow$ cryptanalysis since 2008 still valid
  - Safety margin: from *rock-solid* to *comfortable*

Status of KECCAK cryptanalysis



- Collision attacks up to 5 rounds
  - Also up to 6 rounds, but for non-standard parameters ($c = 160$)

  [Song, Liao, Guo, CRYPTO 2017]
- Stream prediction
  - in 8 rounds (2^{128} time, prob. 1)
  - in 9 rounds (2^{256} time, prob. 1)

  [Dinur, Morawiecki, Pieprzyk, Srebrny, Straus, EUROCRYPT 2015]
- Lots of third party cryptanalysis available at:

  https://keccak.team/third_party.html

How fast is KangarooTwelve?

- At least twice as fast as SHAKE128 on short inputs
- Much faster when parallelism is exploited on long inputs

	Short input	Long input
Intel Core i5-4570 (Haswell)	3.68 c/b	1.44 c/b
Intel Core i5-6500 (Skylake)	2.89 c/b	1.22 c/b
Intel Core i7-7800X (Skylake-X)	2.35 c/b	0.55 c/b

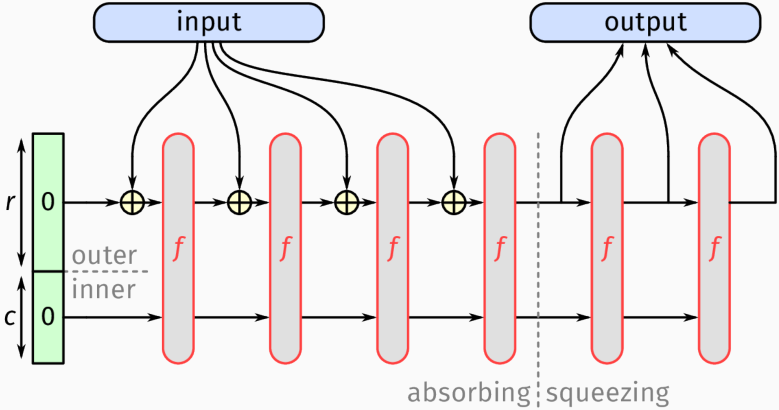
Single core only.

Why is it interesting for the IETF?

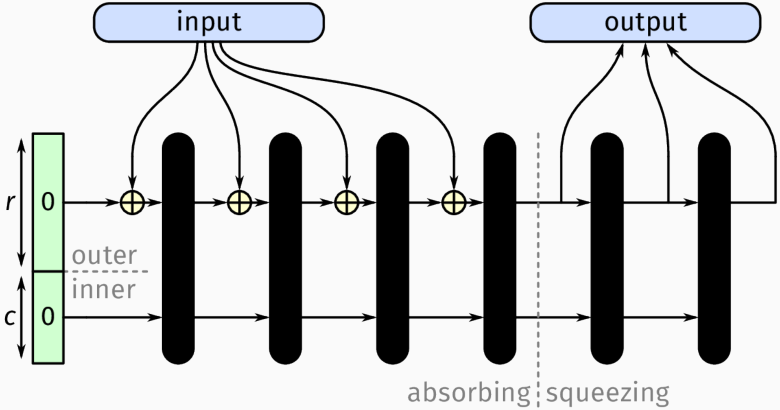
- KECCAK/KANGAROOTWELVE is an open design
  - Public design rationale
  - Result of an open international competition
  - Long-standing active scrutiny from the crypto community
- Best security/speed trade-off
  - Speed-up w/o wasting cryptanalysis resources (no tweaks)
  - Proven generic security
- Scalable parallelism
  - As much parallelism as the implementation can exploit
  - Without parameter

https://tools.ietf.org/html/draft-viguier-kangarootwelve-01

Analyzing the sponge construction



Analyzing the sponge construction



Generic security of the sponge construction

Theorem 2. *A padded sponge construction calling a random permutation, $\mathcal{S}'[\mathcal{F}]$, is (t_D, t_S, N, ϵ)-indistinguishable from a random oracle, for any t_D, $t_S = O(N^2)$, $N < 2^c$ and and for any ϵ with $\epsilon > f_{\mathrm{P}}(N)$.*

If N is significantly smaller than 2^c, $f_P(N)$ can be approximated closely by:

$$f_P(N) \approx 1 - e^{-\frac{(1-2^{-r})N^2+(1+2^{-r})N}{2^{c+1}}} < \frac{(1-2^{-r})N^2+(1+2^{-r})N}{2^{c+1}}. \qquad (6)$$

[EuroCrypt 2008]

http://sponge.noekeon.org/SpongeIndifferentiability.pdf

Generic security of the sponge construction

Theorem 2. *A padded sponge construction calling a random permutation, $\mathcal{S}'[\mathcal{F}]$, is (t_D, t_S, N, ϵ)-indistinguishable from a random oracle, for any t_D, $t_S = O(N^2)$, $N < 2^c$ and and for any ϵ with $\epsilon > f_{\mathrm{P}}(N)$.*

If N is significantly smaller than 2^c, $f_P(N)$ can be approximated closely by:

$$f_P(N) \approx 1 - e^{-\frac{(1-2^{-r})N^2+(1+2^{-r})N}{2^{c+1}}} < \frac{(1-2^{-r})N^2+(1+2^{-r})N}{2^{c+1}}. \qquad (6)$$

[EuroCrypt 2008]

http://sponge.noekeon.org/SpongeIndifferentiability.pdf

Theorem, explained

$$\Pr[\text{attack}] \le \frac{N^2}{2^{c+1}} \text{ (or so)}$$

$\Rightarrow$ if $N \ll 2^{c/2}$, then the probability is negligible

Two pillars of security in cryptography

- Generic security
  - Strong mathematical proofs

Two pillars of security in cryptography

- Generic security
  - Strong mathematical proofs

    $\Rightarrow$ scope of cryptanalysis reduced to primitive

Two pillars of security in cryptography

- Generic security
  - Strong mathematical proofs

    $\Rightarrow$ scope of cryptanalysis reduced to primitive
- Security of the primitive
  - No proof!

Two pillars of security in cryptography

- Generic security
  - Strong mathematical proofs
    $\Rightarrow$ scope of cryptanalysis reduced to primitive
- Security of the primitive
  - No proof!
    $\Rightarrow$ open design rationale

Two pillars of security in cryptography

- Generic security
  - Strong mathematical proofs
    $\Rightarrow$ scope of cryptanalysis reduced to primitive
- Security of the primitive
  - No proof!
    $\Rightarrow$ open design rationale
    $\Rightarrow$ **cryptanalysis!**

Two pillars of security in cryptography

- Generic security
  - Strong mathematical proofs

    $\Rightarrow$ scope of cryptanalysis reduced to primitive
- Security of the primitive
  - No proof!

    $\Rightarrow$ open design rationale

    $\Rightarrow$ third-party **cryptanalysis!**

Two pillars of security in cryptography

- Generic security
  - Strong mathematical proofs

    $\Rightarrow$ scope of cryptanalysis reduced to primitive
- Security of the primitive
  - No proof!

    $\Rightarrow$ open design rationale

    $\Rightarrow$ lots of third-party **cryptanalysis!**

Two pillars of security in cryptography

- Generic security
  - Strong mathematical proofs

    $\Rightarrow$ scope of cryptanalysis reduced to primitive
- Security of the primitive
  - No proof!

    $\Rightarrow$ open design rationale

    $\Rightarrow$ lots of third-party **cryptanalysis!**
  - Confidence

    $\Leftarrow$ sustained cryptanalysis activity and no break

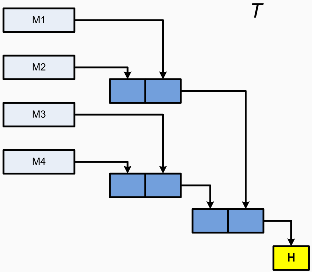
    $\Leftarrow$ proven properties

Impact of parallelism

KECCAK-$f[1600] \times 1$	1070 cycles
KECCAK-$f[1600] \times 2$	1360 cycles
KECCAK-$f[1600] \times 4$	1410 cycles

CPU: Intel Core i5-6500 (Skylake) with AVX2 256-bit SIMD

Tree hashing



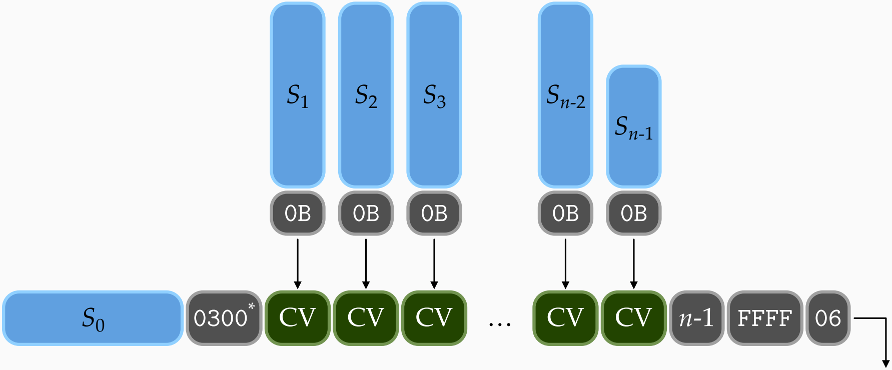
Example: **ParallelHash** [SP 800-185]

function	instruction set	cycles/byte [1]
KECCAK$[c = 256] \times 1$	x86_64	6.29
KECCAK$[c = 256] \times 2$	AVX2	4.32
KECCAK$[c = 256] \times 4$	AVX2	2.31

CPU: Intel Core i5-6500 (Skylake) with AVX2 256-bit SIMD

[1] for long messages.

KangarooTwelve's mode



Final node growing with kangaroo hopping and Sakura coding

[ACNS 2014]