Radbound University

Koala: A Low-Latency Pseudorandom Function

Joint work with Parisa Amiri Eliasi, Yanis Belkheyar, Joan Daemen, Santosh Ghosh, Daniël Kuijsters, Alireza Mehrdad, Silvia Mella, Shahram Rasoolzadeh and Gilles Van Assche

SAC, August, 2024, Montréal, Canada

ESCADA

Outline

Goal

First attempt at doing better

Strengthening LoLaSub: KOALA

Extending it

Building it

Goal

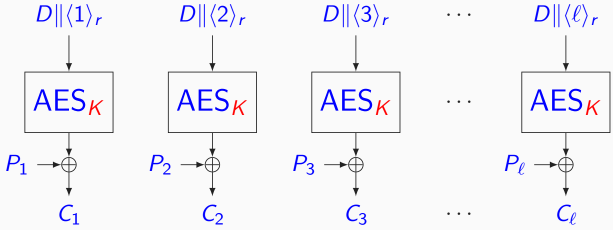
Goals

1. Protect **confidentiality** of relatively short blocks of data
   - e.g., encryption of external memory, fields in databases, ...
   - authentication is not required or solved with a separate MAC
   - ...so stream encryption will do $C \leftarrow P + Z$ with $Z = \mathrm{SC}_K(D)$
2. With **low latency**: in particular
   - short time between availability of diversifier (nonce) D
   - keystream block Z
   - should be just a few cycles with a multi GHz clock
   - aproximate the latency with the gate type and depth
3. With **128 bits of security** for any (plausibly limited) adversary

Contenders

- AES [Daemen/Rijmen, 1998] in counter mode

Standard stream encryption: AES [Daemen/Rijmen, 1998] in counter mode



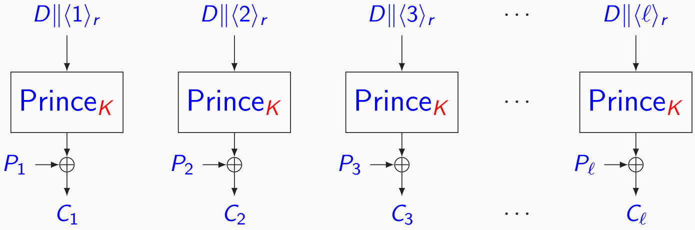
Problem: it is not low latency

- Due to long critical path in AES
- AES 8-bit Sbox has a gate depth of 16
- Mainly due to heavy S-box

Contenders

- AES [Daemen/Rijmen, 1998] in counter mode
- Prince[Borghoff et al., 2012] in counter mode

Low-latency alternative: Prince[Borghoff et al., 2012] in counter mode

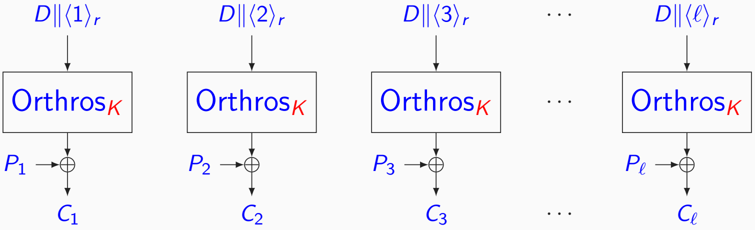


Problem: it does not offer 128 bits of security

- Distinguishing blocks $Z_{D,i} = \text{Prince}_K(D || \langle i \rangle)$ from fully random blocks:
- Random blocks will have a collision when $M \approx 2^{n/2}$, blocks $Z_{D,i}$ never collide
- Prince has block length $n = 64$ so security strength is about 32

Contenders

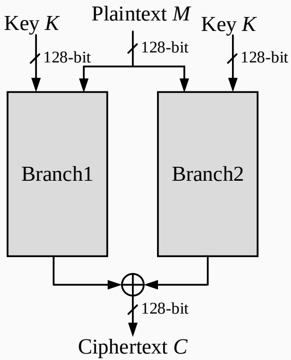
- AES [Daemen/Rijmen, 1998] in counter mode
- PRINCE[Borghoff et al., 2012] in counter mode
- ORTHROS[Banik et al. ToSC 2021] or
- GLEEOK[Anand et al. CHES 2024]



This is it!

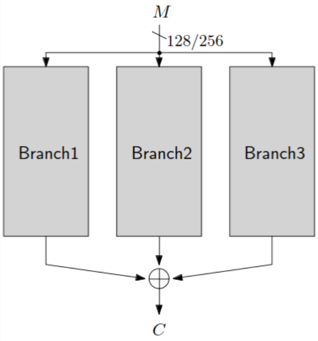
- Dedicated low-latency design, like Prince
- Security objective: instead of pseudorandom permutation (PRP) like block ciphers
- . . . pseudorandom function (PRF)

Orthros block diagram



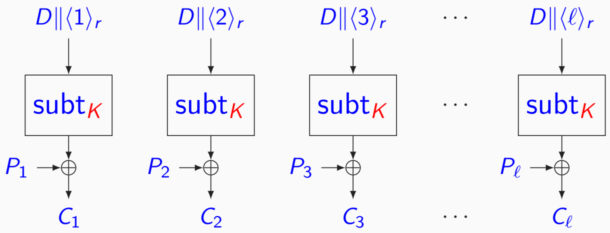
- 128-bit output is sum of two 128-bit block ciphers each applied to the same input
- Paradigm: *sum of (pseudo)random permutations*
- Kind of suboptimal: two block cipher computations per keystream block

Gleeok block diagram



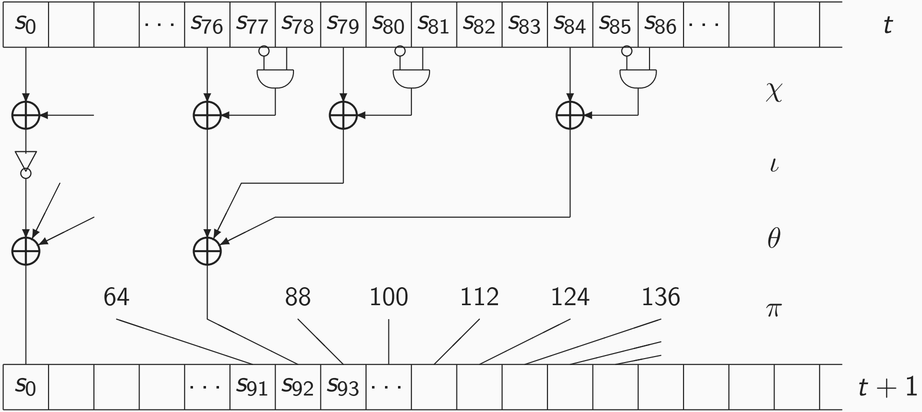
- 128/256-bit output is sum of three 128/256-bit block ciphers each applied to the same input
- Paradigm: *sum of (pseudo)random permutations*
- Kind of suboptimal: three block cipher computations per keystream block

First attempt at doing better



- subt is 257-bit permutation of Subterranean 2.0 [Daemen et al., 2019] with 8 rounds
- subt_K is subt in the Even-Mansour construction
- subt_K is invertible: security limited by $s \leq 258 - \log_2 M_{\max}$, so no worries
- But is it low-latency?

The Subterranean 2.0 round function



Critical path: 3 XORs and 1 (N)AND

Attack vectors for LoLaSub

- Differential cryptanalysis (DC)
  - try to find differentials over 7 or 6 rounds with high DP
  - ...via differential trails with low weight (DP $\leq 2^{-78}$ for 6 round trails)
  - exploit the differential as a distinguisher to determine bits of whitening keys
- Linear cryptanalysis (LC)
  - try to find linear approximations over 7 or 6 rounds with high correlation
  - ...via linear trails with low weight
  - exploit the linear approximation to determine bits of whitening keys
- Refinements and combinations of DC and LC
- Integral attacks AKA cube attack AKA higher-order differential attacks

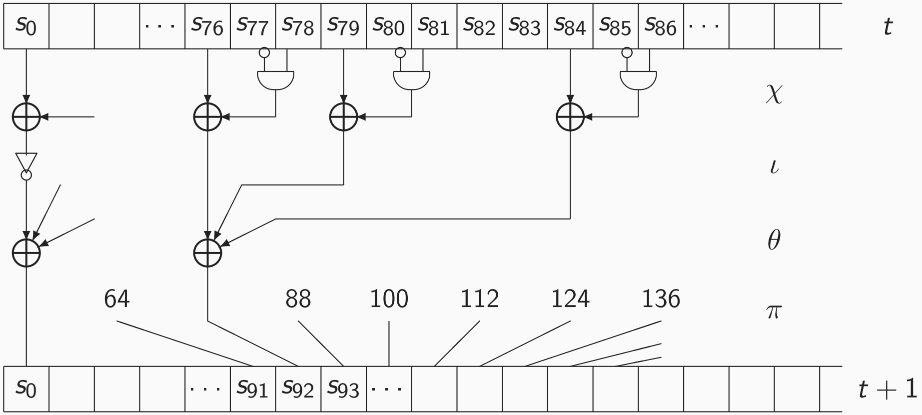
  ...**this appear to be the most powerful attacks against LoLaSub**

Integral attacks on LoLaSub

- Round function has degree 2, r rounds have degree (at most) 2^r
- Basic attack on r-round version recovering input whitening key
  - Find a set leading to zero sum dependent of t-bit key
  - Sum over the set for all 2^t possible key
  - Guess the right t-bit key
  - Find another set depending on different key bit and restart.
- Basic attack on r-round version has data complexity 2^{2^r-1} blocks
  - 5 round LoLaSub: pratical 2-bit key-recovery attack (data comlpexity 2^{32})
  - 6-round LoLaSub: attack with data complexity 2^{63} blocks and maybe 7-round
  - 8 rounds: thin security margin
- And there are other attack variants ...

Strengthening LoLaSub: Koala

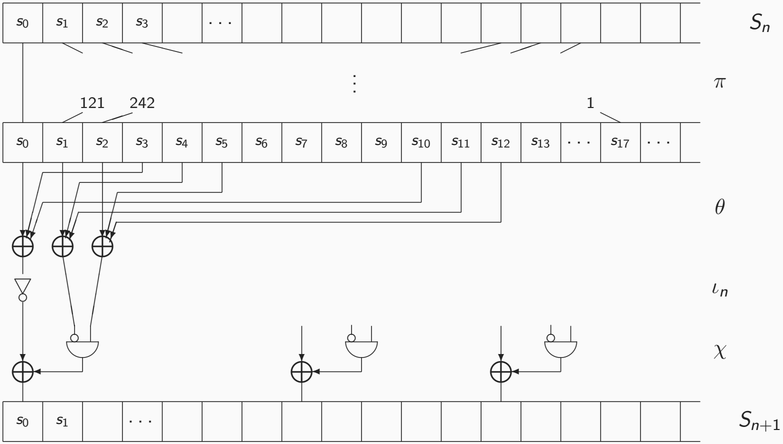
The Subterranean 2.0 round function



Strengthening LoLaSub via the permutation

1. Changing the parameters of the linear layer
   - different offsets in θ and different multiplication factor in π
2. Rephasing the round function moving the non-linear layer to the end
   - linear layer between last non-linear layer and key addition has no added value
   - linear layer before the first non-linear layer does have added value (against integral attacks)
3. We call the result KOALA-**P**

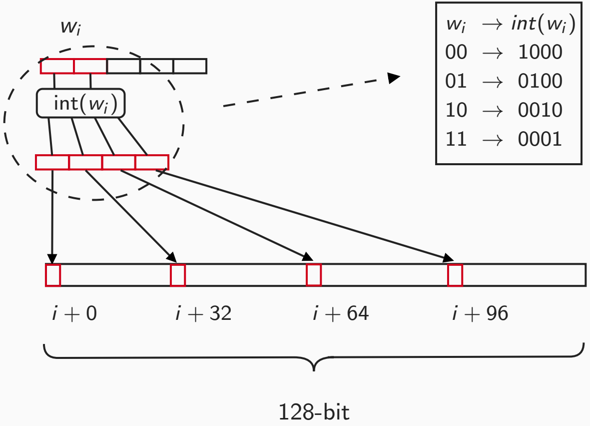
These changes reduce the probability that the basic attack can be extended by one (or) two rounds


s_0 s_1 s_2 s_3 . . .
S_n
π
121
242
1
s_0 s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 s_9 s_{10} s_{11} s_{12} s_{13} . . . s_{17} . . .
θ
ι_n
χ
s_0 s_1 . . .
S_{n+1}

Strengthening LoLaSub via the input

- Reducing the input length from 257 bits to 64 bits
- Demultiplexer-like input injection
  - 64-bit input D parses into a sequence of 2-bit integers $e_i = d_{2i} + 2d_{2i+1}$
  - each index i has 4 associated positions in the state p_0, p_1, p_2 and p_4
  - an input e_i complements the bit in position p_{e_i}
- Properties of demultiplexer
  - input set after injection has affine subspaces of dimension at most 32
  - demultiplexer layer has latency only 1 (N)AND and algebraic degree 2
- These changes strongly reduce the degrees of freedom of the attacker

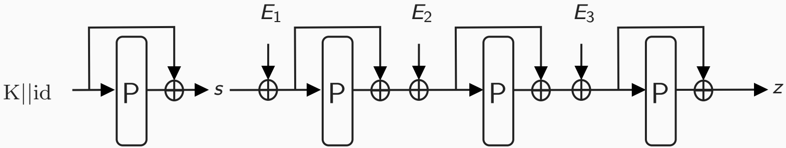
Our analysis suggests there is no exploitable integral distinguisher based on algebraic representation above 5 rounds



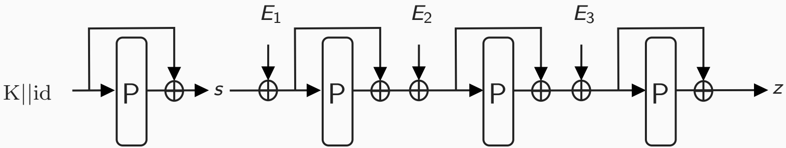
w_i
$\text{int}(w_i)$
$w_i \rightarrow int(w_i)$
$00 \rightarrow 1000$
$01 \rightarrow 0100$
$10 \rightarrow 0010$
$11 \rightarrow 0001$
$i + 0$
$i + 32$
$i + 64$
$i + 96$
128-bit

Extending it

- Limitation to 64-bit input may be restrictive
- Therefore we introduce iteration to support multiple 64-bit blocks
  - we replace Even-Mansour by secret initial state and feedforward
  - we impose prefix-free encoding
- We relax the low-latency requirement to last input block



We call it **Kirby** and in combination with KOALA-P and input encoding: KOALA

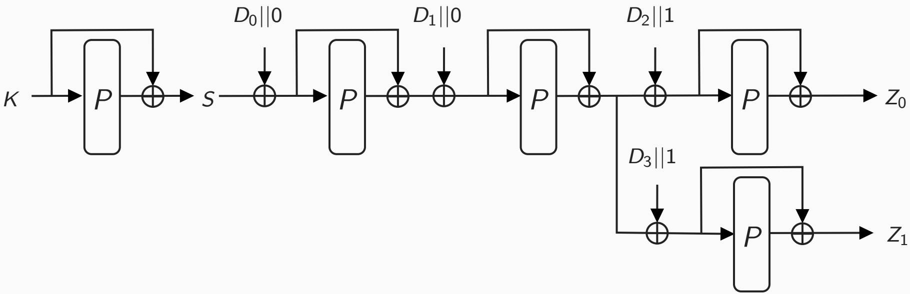


- Arbitrary number of input blocks
- Feedforward gives some level of leakage resilience
- We prove an upper bound on PRF advantage in random permutation model
- If inputs E form prefix-free set (and if ids are unique also multi-user):

$$\mathrm{Adv}_{\mathsf{PRF}} < \frac{3M^2}{2^b} + \frac{NM}{2^b} + \frac{N}{2^{|K|}}.$$

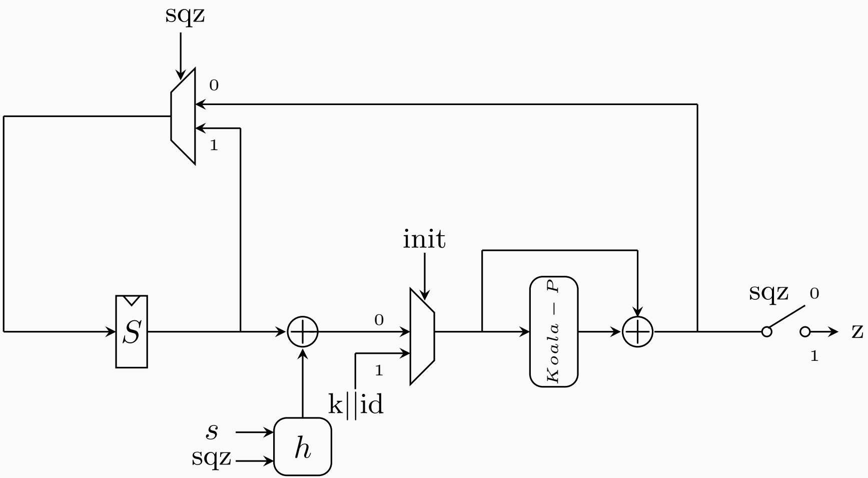
Kirby: example of prefix-free encoding

Solution that costs 1 bit per block:



Building it

Hardware architecture for Koala



Performance comparison

Table: Synthesis results for the Nangate 15nm library.

Cipher	Output width	Area		Latency	MaxTp	MaxTp/Area
	[bits]	[μm^2]	[GE]	[ps]	[Gbits/s]	[Mbits/(s $\times$ μm^2)]
Koala	257	4175	21236	395	651	156
Kirby+sub	257	4167	21196	399	644	155
Prince	64	1696	8627	482	133	78.4
Orthros	128	5993	30482	400	320	53.4
Gleeok-128	128	9887	50291	400	320	32.4
Gleeok-256	256	26043	132462	550	465	17.8

Thanks for your attention!