Fixslicing - Application to some NIST LWC round 2 candidates

Alexandre Adomnicai Thomas Peyrin

Nanyang Technological University, Singapore

Temasek Laboratories, Singapore

Lightweight Cryptography Workshop 2020

What this talk is about

- **Constant-time** software implementations on 32-bit platforms
- Application of the **fixslicing implementation strategy** to some NIST LWC round 2 candidates built upon AES-128, GIFT-128 and Skinny-128 primitives
- Benchmarking results on **ARM Cortex-M3** for payloads up to 256 bytes

The fixslicing implementation strategy

- Initially introduced as **a new representation** for the GIFT block ciphers [ANP20]

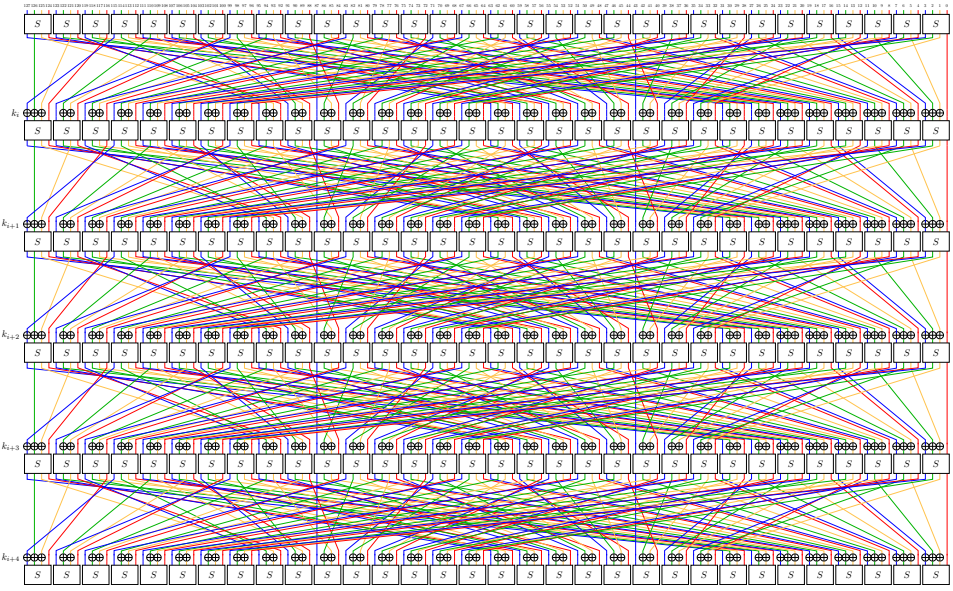
The fixslicing implementation strategy

- Initially introduced as **a new representation** for the GIFT block ciphers [ANP20]
- Fixsliced GIFT-128 **runs about 7x faster** on ARM Cortex-M3 compared to a naive bitsliced implementation

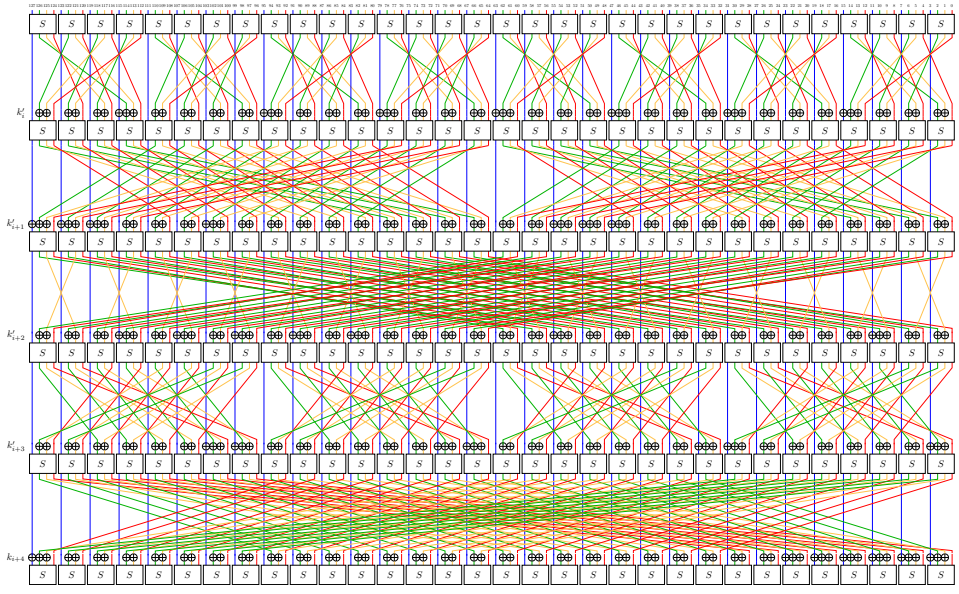
The fixslicing implementation strategy

- Initially introduced as **a new representation** for the GIFT block ciphers [ANP20]
- Fixsliced GIFT-128 **runs about 7x faster** on ARM Cortex-M3 compared to a naive bitsliced implementation
- Consists in **fixing a slice to never move** and adjusting the others for the S-box layer

Classical representation of GIFT-128



Fixsliced representation of GIFT-128



Genericity of the fixslicing technique

Actually, the fixslicing technique is a particular case for permutations which ensures that, from a bitsliced perspective, all bits within a slice remain in the same one through the permutation. Therefore, it can be applied to all permutations that verify this property, and the number of rounds to consider for the decomposition equals $min(\mathsf{order}(P_i))$ for all i.

Figure: Extract from [ANP20]

- ▷ So, only of interest for Substitution-bitPermutation Networks (SbPN)?

Genericity of the fixslicing technique

Actually, the fixslicing technique is a particular case for permutations which ensures that, from a bitsliced perspective, all bits within a slice remain in the same one through the permutation. Therefore, it can be applied to all permutations that verify this property, and the number of rounds to consider for the decomposition equals $min(\mathsf{order}(P_i))$ for all i.

Figure: Extract from [ANP20]

- ▷ So, only of interest for Substitution-bitPermutation Networks (SbPN)? **NOPE!**
- ▷ Many ciphers spend cycles to move bits within the slices to achieve better diffusion ⇒ **alternative representations might be valuable even for more complex linear layers**

Application to AES-like ciphers

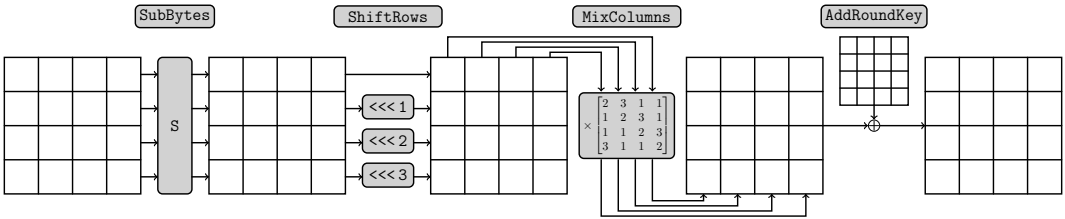


Figure: AES round function

Application to AES-like ciphers

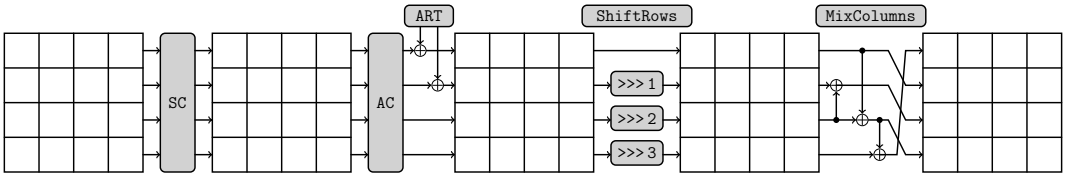


Figure: Skinny round function

Application to AES-like ciphers

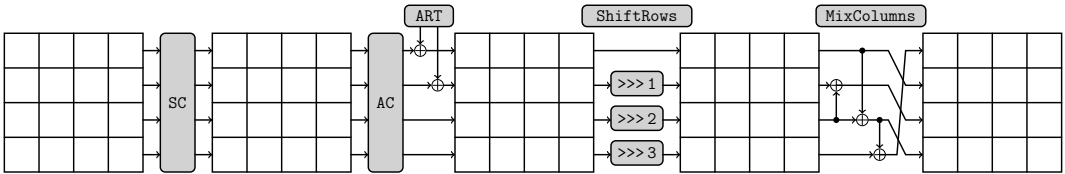
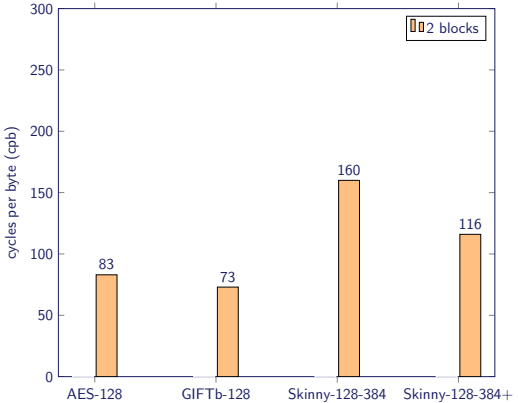


Figure: Skinny round function

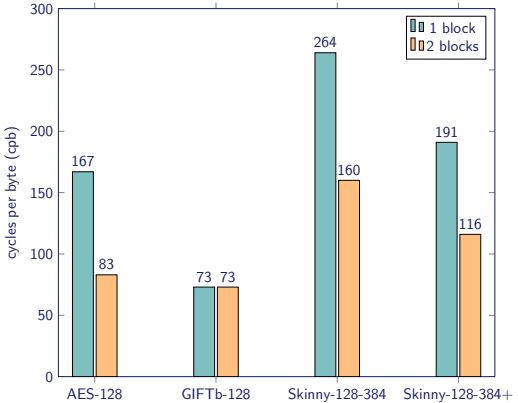
- Performance improvements for AES and Skinny-128 on ARM Cortex-M and E31 RISC-V processors [AP20]

Implementation results on ARM Cortex-M3



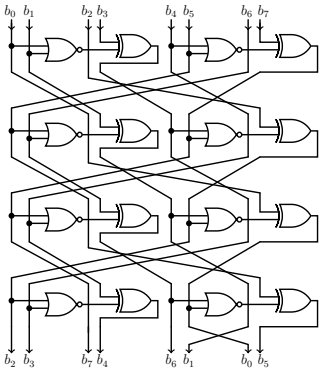
Performance for constant-time implementations on ARM Cortex-M3

Implementation results on ARM Cortex-M3

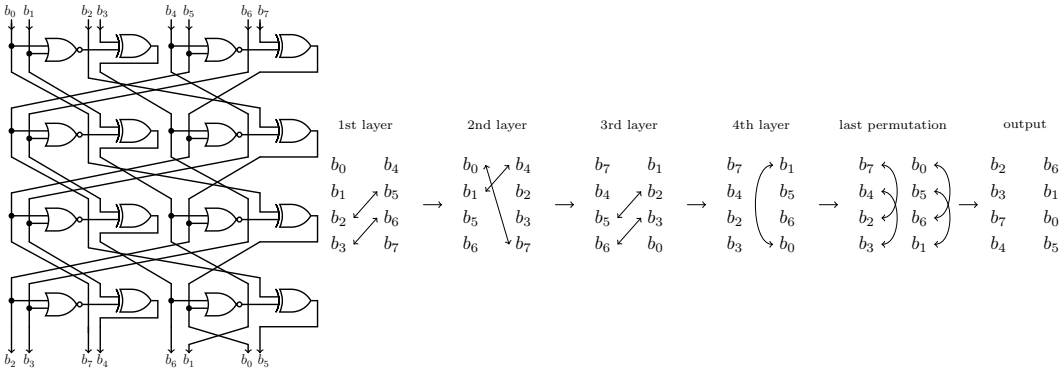


Performance for constant-time implementations on ARM Cortex-M3

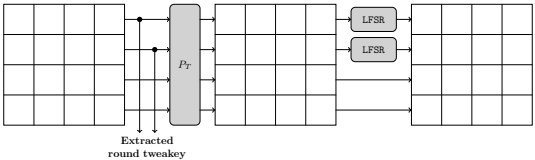
Bitslicing a single block for Skinny-128



Bitslicing a single block for Skinny-128

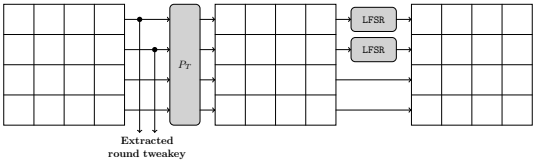


Speed optimized Skinny tweakey schedule

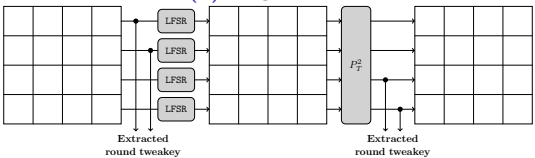


(a) Single round

Speed optimized Skinny tweakey schedule



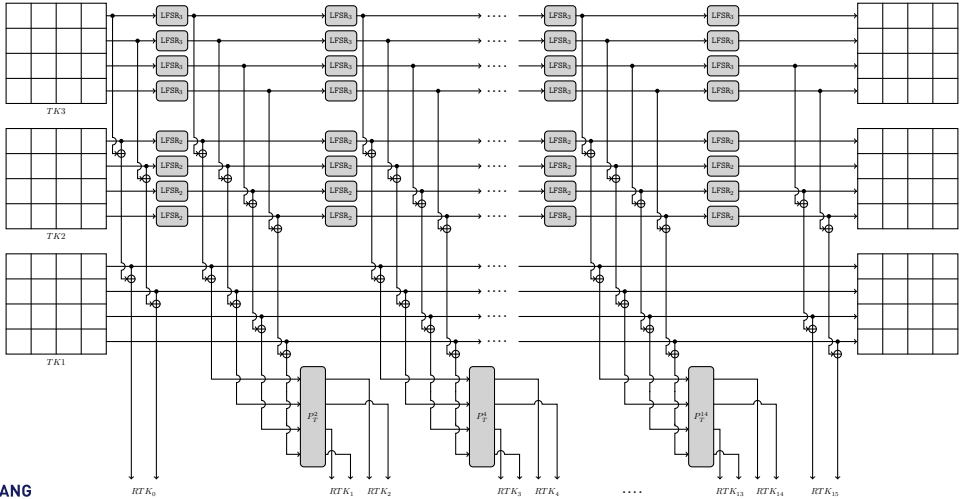
(a) Single round



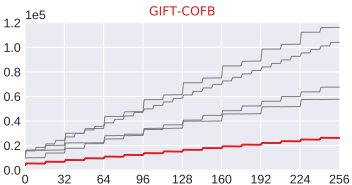
(b) Double round

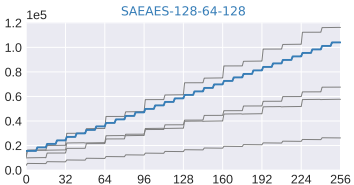
Figure: Skinny tweakey schedule round function

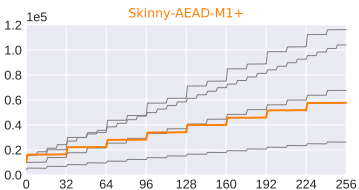
Speed optimized Skinny tweakey schedule

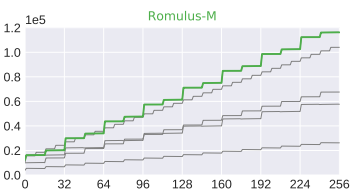


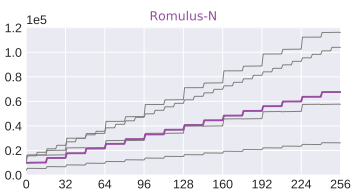
Benchmark results on ARM Cortex-M3











What about other candidates?

- Fixslicing may be **valuable for other candidates**!
  - PHOTON-Beetle? (AES-like primitive)
  - Elephant? (Spongent is an SbPN)
  - ...

What about other candidates?

- Fixslicing may be **valuable for other candidates**!
  - PHOTON-Beetle? (AES-like primitive)
  - Elephant? (Spongent is an SbPN)
  - ...
- Some primitives are **fixsliced by design** (e.g. Ascon-p)

Thanks for your attention!

Questions?

Feel free to contact us at firstname.lastname@ntu.edu.sg

References

Alexandre Adomnicai, Zakaria Najm, and Thomas Peyrin, Fixslicing: A New GIFT Representation: Fast Constant-Time Implementations of GIFT and GIFT-COFB on ARM Cortex-M, IACR Transactions on Cryptographic Hardware and Embedded Systems 2020 (2020), no. 3, 402–427.

Alexandre Adomnicai and Thomas Peyrin, Fixslicing AES-like Ciphers: New bitsliced AES speed records on ARM-Cortex M and RISC-V, Cryptology ePrint Archive, Report 2020/1123, 2020, https://eprint.iacr.org/2020/1123.