SCREAM minus iSCREAM

Side-Channel Resistant Authenticated Encryption with Masking

Vincent Grosso[1] Gaëtan Leurent[2]
François-Xavier Standert[1] Kerem Varici[1]

Anthony Journault[1] François Durvaux[1]
Lubos Gaspar[1] Stéphanie Kerckhof[1]

[1]UCL, Belgium & [2]Inria, France

scream@uclouvain.be

DIAC 2014

Authenticated Encryption

Many different ways to build authenticated encryption

- Block cipher based
  - 2-pass: GCM, CCM, ...
  - 1-pass: OCB, ...
  - Nonce-misuse resistant: SIV, COPA, POET, ...
- Permutation based
  - SpongeWrap, DuplexWrap, MonkeyWrap, APE, ...
- Stream cipher + MAC
  - Encrypt-then-MAC, MAC-then-Encrypt, Encrypt-and-MAC
- Dedicated
  - Helix/Phelix, ALE, ...

Block cipher modes

- Block ciphers are very popular primitives
  - Efficient: lightweight block ciphers as small as stream ciphers
  - Versatile: modes for encryption, authentication, authenticated encryption, hashing, key derivation, ...
  - We (mostly) know how to build them: AES is a trusted standard
  - No attacks against AES with less than 2^{128} data and time
- Need a mode of operation
  - To deal with messages of arbitrary length
  - To achieve specific security goals
  - Encryption: CBC, CFB, OFB, CTR
  - Authenticated Encryption: OCB, SILC, CLOC, OTR, COPA, JAMBU, ELmD, POET, ...
  - Most modes offer birthday security: there are attacks with 2^{64} data

Birthday-bound security

Birthday bound security

Most modes based on an n-bit primitive can only encrypt $2^{n/2}$ blocks securely

- Because collisions in the internal state reveal information
  - *E.g.*, CBC collisions reveal plaintext XOR
- Because proofs require the PRF-PRP switching lemma
  - *E.g.* CTR mode is distinguishable after $2^{n/2}$ blocks

Birthday-bound security

Birthday bound security

Most modes based on an *n*-bit primitive can only encrypt $2^{n/2}$ blocks securely

- Modes with a 128-bit primitive (AES) have limited security
  - Google stores about 15EB (2^{60} 128-bit blocks)
  - Internet traffic is about 1ZB/year (2^{66} 128-bit blocks)

Solutions

- Use a larger primitive: OMD, Minalpher, Sponges, ...
- Use beyond-birthday modes: CENC, SHELL
- Use a tweakable block cipher: Deoxys, Joltik, SCREAM

Birthday-bound security

Birthday bound security

Most modes based on an n-bit primitive can only encrypt $2^{n/2}$ blocks securely

- Modes with a 128-bit primitive (AES) have limited security
  - Google stores about 15EB (2^{60} 128-bit blocks)
  - Internet traffic is about 1ZB/year (2^{66} 128-bit blocks)

Solutions

- Use a larger primitive: OMD, Minalpher, Sponges, ...
- Use beyond-birthday modes: CENC, SHELL
- Use a tweakable block cipher: Deoxys, Joltik, SCREAM

Birthday-bound security

Birthday bound security

Most modes based on an n-bit primitive can only encrypt $2^{n/2}$ blocks securely

- Modes with a 128-bit primitive (AES) have limited security
  - Google stores about 15EB (2^{60} 128-bit blocks)
  - Internet traffic is about 1ZB/year (2^{66} 128-bit blocks)

Solutions

- Use a larger primitive: OMD, Minalpher, Sponges, ...
- Use beyond-birthday modes: CENC, SHELL
- Use a tweakable block cipher: Deoxys, Joltik, SCREAM

Birthday-bound security

Birthday bound security

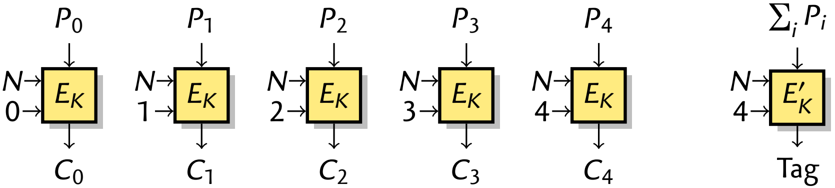
Most modes based on an n-bit primitive can only encrypt $2^{n/2}$ blocks securely

- Modes with a 128-bit primitive (AES) have limited security
  - Google stores about 15EB (2^{60} 128-bit blocks)
  - Internet traffic is about 1ZB/year (2^{66} 128-bit blocks)

Solutions

- Use a larger primitive: OMD, Minalpher, Sponges, ...
- Use beyond-birthday modes: CENC, SHELL
- Use a tweakable block cipher: Deoxys, Joltik, SCREAM

Tweakable block cipher based AE modes



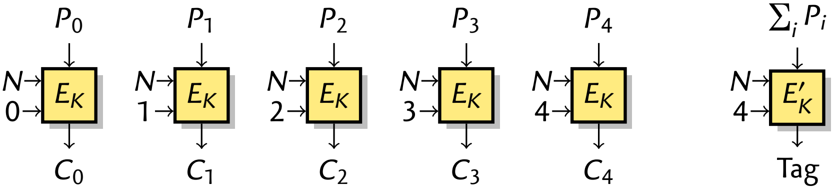
Definition (Tweakable block cipher – Liskov, Rivest, Wagner)

Family of permutation indexed by a key K (secret) and a tweak T (public)

For each tweak T, $x \mapsto E_K(T, x)$ is an independent PRF

- TAE: Tweakable Authenticated Encryption (Liskov, Rivest, Wagner)
  - Inspired by OCB
  - Tweak is Nounce+Counter
  - Full n-bit security

Tweakable block cipher based AE modes



TAE Features

- Fully parallelizable
- 128-bit security with 128-bit state
  - + key, nounce, checksum
- Low overhead for authentication (1TBC)
- Minimal extension
- Patent-free?

TBC design

We want to design a tweakable block cipher that is efficient on wide range of platform and secure.

- Side-channel resistance necessary in many lightweight settings
  - Avoid your car keys / credit card being cloned
- Usual approach:
  1. Design a secure cipher (AES, PRESENT, Noekeon, ...)
  2. Implement with side-channel countermeasures
- We use LS-Designs, with a reverse approach:
  1. Use operations that are easy to mask
  2. In order to design a secure cipher
- Previous work: Zorro, PICARO

TBC design

We want to design a tweakable block cipher that is efficient on wide range of platform and secure.

- Side-channel resistance necessary in many lightweight settings
  - Avoid your car keys / credit card being cloned
- Usual approach:
  1. Design a secure cipher (AES, PRESENT, Noekeon, ...)
  2. Implement with side-channel countermeasures
- We use LS-Designs, with a reverse approach:
  1. Use operations that are easy to mask
  2. In order to design a secure cipher
- Previous work: Zorro, PICARO

Choice of operations

Important remark

Logic gates are easier to mask than table-based S-boxes
(If we target Boolean masking)

- Use bitsliced S-boxes (SERPENT, Noekeon, ...)
  - One word contains the msb (resp. 2nd bit, ...) of every S-box
  - Bitwise operations: 8 S-boxes in parallel using 8-bit words
  - Use a small number of non-linear gates
- We can use tables for the diffusion layer!
  - Efficient, good diffusion
  - Easy to mask (linear)

Choice of operations

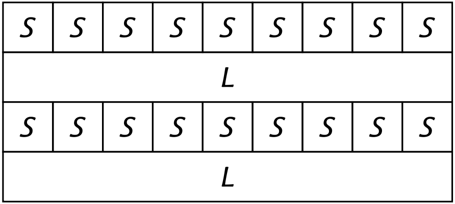
Important remark

Logic gates are easier to mask than table-based S-boxes
(If we target Boolean masking)

- Use bitsliced S-boxes (SERPENT, Noekeon, ...)
  - One word contains the msb (resp. 2nd bit, ...) of every S-box
  - Bitwise operations: 8 S-boxes in parallel using 8-bit words
  - Use a small number of non-linear gates
- We can use tables for the diffusion layer!
  - Efficient, good diffusion
  - Easy to mask (linear)

LS-designs

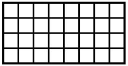
- Mathematical description: SPN network
  - S-boxes
    - With simple gate representation
  - Linear diffusion layer
    - Mixes the full state
    - Binary coefficients
  - Good design criterion: wide-trail



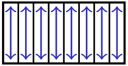
- Bitslice implementation:
  - S-box as a series of bitwise operations with CPU words
  - L-box tables for diffusion layer
  - Easy to mask (simple non-linear ops., complex linear ops.)

LS-designs

$x \leftarrow P \oplus K$
for $0 \le r < N_r$ **do**
  ▷ S-box layer:
  for $0 \le i < l$ **do**
    $x[i,\star] = S[x[i,\star]]$
  ▷ L-box layer:
  for $0 \le j < s$ **do**
    $x[\star,j] = L[x[\star,j]]$
  ▷ Key addition:
  $x \leftarrow x \oplus k_r$
return x



State as a bit-matrix



S-box layer

L-box layer

Changes in SCREAM v3

1. SCREAM v3 uses the original TAE mode
   Mistakes in the initial mode (TAE variant)
   [Lei & Siang]
2. iSCREAM removed
   Problems with iSCREAM S-Box and L-Box
   [Leander, Minaud & Rønjom]
3. Improved S-Box
   Better difference probability
   [Canteaut, Duval & Leurent]

Ch

1. SCREAM v3 uses
   Mistakes in the in
   [Lei & Siang]

2. iSCREAM removed
   Problems with iSCREA
   nder, Minaud & Rønjom]

3. Improved S-Box
   Better difference probability
   [Canteaut, Duval & Leurent]

Changes in SCREAM v3

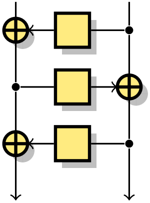
1. SCREAM v3 uses the original TAE mode
   Mistakes in the initial mode (TAE variant)
   [Lei & Siang]
2. iSCREAM removed
   Problems with iSCREAM S-Box and L-Box
   [Leander, Minaud & Rønjom]
3. Improved S-Box
   Better difference probability
   [Canteaut, Duval & Leurent]

Changes in SCREAM v3

1. SCREAM v3 uses the original TAE mode
   Mistakes in the initial mode (TAE variant)
   [Lei & Siang]
2. iSCREAM removed
   Problems with iSCREAM S-Box and L-Box
   [Leander, Minaud & Rønjom]
3. Improved S-Box
   Better difference probability
   [Canteaut, Duval & Leurent]

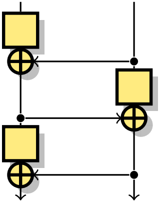
Constructing S-Boxes from smaller ones

Trade-off between S-Box properties and implementation cost



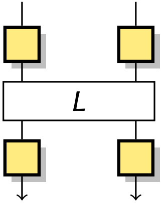
Feistel

- Crypton v0.5
- ≈ Zorro
- Robin



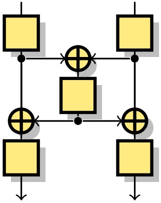
Misty

- Fantomas



SPN

- Crypton v1.0
- Iceberg
- Khazad



Lai-Massey

- Whirlpool

8-bit Feistel & MISTY S-Boxes

Results from [CDL SAC'15]:

Feistel

- $\delta(F) \geq 8$, tight
  - Requires S_1, S_3 APN, S_2 perm. with $\delta(S_2) = 4$
- $\mathcal{L}(F) \geq 48$
  - $\mathcal{L}(F) \geq 64$ if $\delta(F) < 32$

MISTY

- $\delta(F) \geq 8$, tight
  - Requires S_2, S_3 APN, S_1 perm. with $\delta(S_1) = 4$
  - F is not a permutation!
- $\mathcal{L}(F) \geq 48$
  - $\mathcal{L}(F) \geq 64$ if $\delta(F) < 32$
- Permutation: $\delta(F) \geq 16$, tight

- Exhaustive search over small implem. for APN function & perm. with $\delta = 4$

8-bit Feistel & MISTY S-Boxes

Results from [CDL SAC'15]:

Feistel

- $\delta(F) \geq 8$, tight
  - Requires S_1, S_3 APN, S_2 perm. with $\delta(S_2) = 4$
- $\mathcal{L}(F) \geq 48$
  - $\mathcal{L}(F) \geq 64$ if $\delta(F) < 32$

MISTY

- $\delta(F) \geq 8$, tight
  - Requires S_2, S_3 APN, S_1 perm. with $\delta(S_1) = 4$
  - F is not a permutation!
- $\mathcal{L}(F) \geq 48$
  - $\mathcal{L}(F) \geq 64$ if $\delta(F) < 32$
- Permutation: $\delta(F) \geq 16$, tight

- Exhaustive search over small implem. for APN function & perm. with $\delta = 4$

Exhaustive search over small implementations

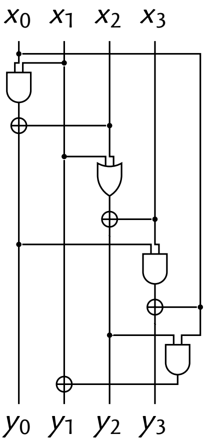
Permutation with $\delta = 4$

- Easy search
  - Re-use results from Üllrich *et al.*
- 9-instruction solutions
  - 4 non-linear
  - 4 XOR
  - 1 copy
- 4 NL gates is optimal

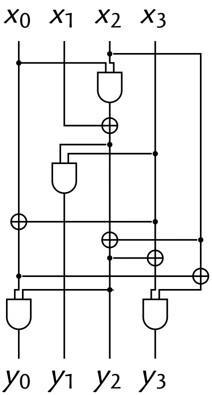
APN function

- Expensive search
  - No permutation filtering
  - 6k core-hours
- 10-instruction solutions
  - But 6 non-linear
- 11-instruction solutions
  - 4 non-linear
  - 5 XOR
  - 2 copy
- 4 NL gates is optimal

Exhaustive search over small implementations



Permutation with $\delta = 4$



APN function

Efficient combined implementation

function $\textsc{Sbox}(W_0, \ldots, W_7)$

$t_0 = (W_1 \wedge W_2) \oplus W_0$
$t_1 = (W_1 \oplus W_3)$
$t_2 = W_2 \oplus t_0$
$W_4 = W_4 \oplus ((W_3 \oplus t_2) \wedge (W_2 \oplus t_1))$
$W_5 = W_5 \oplus t_2$
$W_6 = W_6 \oplus (W_3 \wedge t_0)$
$W_7 = W_7 \oplus (t_1 \wedge t_2)$
$t_0 = (W_4 \wedge W_5) \oplus W_6$
$t_1 = (W_5 \vee W_6) \oplus W_7$
$t_2 = (W_7 \wedge t_0) \oplus W_4$
$t_3 = (W_4 \wedge t_1) \oplus W_5$
$W_0 = W_0 \oplus t_0$
$W_2 = W_2 \oplus t_1$
$W_1 = W_1 \oplus t_2$
$W_3 = W_3 \oplus t_3$
$t_0 = \neg((W_1 \wedge W_2) \oplus W_0)$
$t_1 = (W_1 \oplus W_3)$
$t_2 = W_2 \oplus t_0$
$W_4 = W_4 \oplus ((W_3 \oplus t_2) \wedge (W_2 \oplus t_1))$
$W_5 = W_5 \oplus t_2$
$W_6 = W_6 \oplus (W_3 \wedge t_0)$
$W_7 = W_7 \oplus (t_1 \wedge t_2)$

function $\textsc{InvSbox}(W_0, \ldots, W_7)$

$t_0 = \neg((W_1 \wedge W_2) \oplus W_0)$
$t_1 = (W_1 \oplus W_3)$
$t_2 = W_2 \oplus t_0$
$W_4 = W_4 \oplus ((W_3 \oplus t_2) \wedge (W_2 \oplus t_1))$
$W_5 = W_5 \oplus t_2$
$W_6 = W_6 \oplus (W_3 \wedge t_0)$
$W_7 = W_7 \oplus (t_1 \wedge t_2)$
$t_0 = (W_4 \wedge W_5) \oplus W_6$
$t_1 = (W_5 \vee W_6) \oplus W_7$
$t_2 = (W_7 \wedge t_0) \oplus W_4$
$t_3 = (W_4 \wedge t_1) \oplus W_5$
$W_0 = W_0 \oplus t_0$
$W_2 = W_2 \oplus t_1$
$W_1 = W_1 \oplus t_2$
$W_3 = W_3 \oplus t_3$
$t_0 = (W_1 \wedge W_2) \oplus W_0$
$t_1 = (W_1 \oplus W_3)$
$t_2 = W_2 \oplus t_0$
$W_4 = W_4 \oplus ((W_3 \oplus t_2) \wedge (W_2 \oplus t_1))$
$W_5 = W_5 \oplus t_2$
$W_6 = W_6 \oplus (W_3 \wedge t_0)$
$W_7 = W_7 \oplus (t_1 \wedge t_2)$

S-Box comparison

		Implem.		Properties	
S-Box	Construction	$\wedge, \vee$	$\oplus$	$\mathcal{L}$	δ
AES	Inversion	32	83	32	4
Whirlpool	Lai-Massey	36	58	64	8
CRYPTON	3-r. Feistel	49	12	64	8
iSCREAM v1	3-r. Feistel	12	24	64	16
SCREAM v1	3-r. MISTY (3/5 bits)	11	25	64	16
LS (unnamed)	Whirlpool-like	16	41	64	10
SCREAM v3	3-r. Feistel	12	27	64	8

SCREAM v3 S-box

- Only 3 extra operations (1 non-linear)
- Improved differential probability, no fixed points
- Inverse S-box almost for free

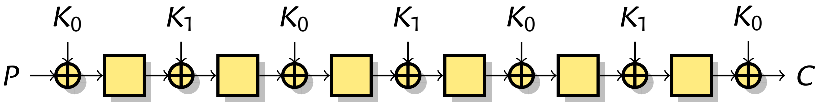
SCREAM S-box and L-box

Choice of components:

- 8-bit S-box
  - Built from 3 smaller S-boxes (Feistel-like structure)
  - $\mathrm{Pr}_{\mathrm{lin}} = 2^{-2}$, $\mathrm{Pr}_{\mathrm{diff}} = 2^{-5}$, 12 non-linear gates
  - Differential trails must have less than 26 active S-Boxes
- 16-bit L-box
  - Branch number 8 (optimal for a binary matrix)
  - Orthogonal matrix: differential and linear properties equivalent
  - Built from $QR[32, 16, 8]$

Tweak/Key schedule

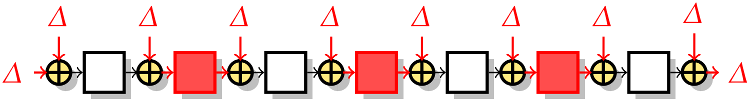
- Add a tweak/key schedule to turn block cipher into tweakable block
  - 128-bit block
  - 128-bit key
  - 128-bit tweak
- Tweak and key have a similar role (cf. TWEAKEY framework)
- Must be secure against chosen-tweak attacks (≈ related-key)
- Use ideas from LED:



  - One step is two rounds: $\mathcal{B}$ active S-Boxes
  - At least half the steps are active with related-key

Security against Differential and Linear Cryptanalysis

- Fixed key ⊕ Chosen tweak ≈ Related key
  At least one half of the steps active
- Wide-trail strategy:
  every 2-round step has at least 8 active S-boxes.

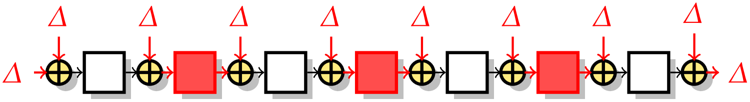


Minimum number of active S-Boxes

Setting	1	2	3	4	5	6	7	8	9	10	11	12
Single Key	0	8	8	16	16	24	24	**32**				
Related Key	0	0	8	8	8	16	16	16	24	24	24	**32**

Security against Differential and Linear Cryptanalysis

- Fixed key ⊕ Chosen tweak ≈ Related key
  At least one half of the steps active
- Wide-trail strategy:
  every 2-round step has at least 8 active S-boxes.



Minimum number of active S-Boxes

Setting	1	2	3	4	5	6	7	8	9	10	11	12
Single Key	0	8	8	16	16	24	24	**32**				
Related Key	0	0	8	8	8	16	16	16	24	24	24	**32**

Improved Security Analysis

- Components designed to make those simple trails expensive.
  - Combine analysis at step level, and analysis at S-box level
- Optimal trails have two third of the steps active (fixed key).
  - See submission for more details
- We recommend
  - 10 rounds for single key security
  - 12 rounds for related key security

Minimum number of active S-Boxes

Setting	1	2	3	4	5	6	7	8	9	10	11	12
Single Key	0	8	14	20	**28**	35						
Related Key	0	0	8	14	14	22	**28**	28	36			

Implementation: overview

- Hardware:
  - The tweakable block cipher costs about the same as AES
  - Low overhead for TAE mode
  - Parallelism can be leveraged in a pipeline implementation
- Micro-controller:
  - Good performance of the TBC ($< 8k$ cycles)
  - Very good when masking is needed
- High-end CPU:
  - Parallelism exploited with SIMD
    - Vector permute for the L-box
  - Performance similar to AES-GCM *(excluding hardware AES)*

skip details

Implementation: High-end CPUs

- Use large registers (128-bit) for bitsliced S-box
- Use vector permute instructions for L-box
  - 4-bit to 8-bit table with `pshufb` in SSSE3, `vtbl` in NEON
  - 16-bit to 16-bit table as 8 small tables
  - Constant time (no cache timing side-channel)

Results

- Fantomas has performances close to AES *(excluding hardware AES)*
- Tweak gives more security, requires more rounds (20 vs. 12)

- The TAE mode has a very small overhead
- Performances similar to AES-GCM *(excluding hardware AES)*

Implementation: High-end CPUs

Software performance for long messages (cycles/byte)

	SCREAM v3	SCREAM v2	AES-GCM	AES
ARM Cortex A15	-	23.5	31.1	17.8
Atom	57	56	28.8	17
Nehalem	10.8	10.7	9.9	6.9
Ivy Bridge ~~AES-NI~~	7.9	7.7	8.3	5.4
Ivy Bridge AES-NI			2.5	1.3

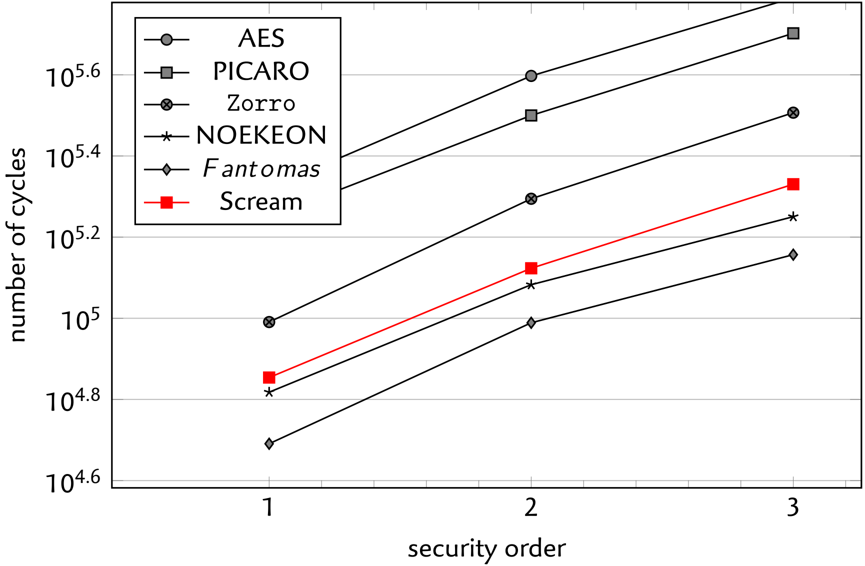
TODO

- AVX2 implementation (Haswell): currently 5.7 c/B
- AVX512-BW implementation (Xeon Skylake?)

Implementation: AVR micro-controller

- TBC performance: ≈ 7700 cycles
  - Using 1kB table
  - Smaller tables possible with more cycles
- In many cases, side-channel protection will be required
  - Scream is very efficient with masking
  - Noekeon also very good (similar components)

Implementation: AVR micro-controller



Implementation: Hardware

- We consider implementations with 128-bit datapath
  - Reasonable cost/performance trade-off
- Low amount of logic in one round
  - We can unroll one full step (2 cycles)
  - One step ≈ one AES round
  - Scream TBC ≈ AES
- Low overhead for TAE mode
  - Few extra state variables

SCREAM Features

TAE Mode

- Fully parallelizable
- 128-bit security / 128-bit state
  - + key, nounce, checksum
- Low overhead (1TBC)
- Minimal extension
- Patent-free?

LS Tweakable Block Cipher

- Clean and simple design
  - SPN, Wide-trail
  - Simple bounds for trails
- Scalable
  - Hardware: small state
  - Microcontrollers: masking
  - High-end CPUs: vectorized

- High security, high performances
- Improved security margin in SCREAM v3
- The tweakable block cipher is also a useful primitive in itself.
  - Can be used with SCT mode for nonce-misuse resistance