CAESAR candidate SCREAM

Side-Channel Resistant Authenticated Encryption with Masking

Vincent Grosso[1] Gaëtan Leurent[1,2]
François-Xavier Standert[1] Kerem Varıcı[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, ...

Authenticated Encryption

Many different ways to build authenticated encryption

Birthday bound security

Most block cipher-based and permutation-based modes only have birthday bound security

They need a $2n$-bit primitive to resist attacks with 2^n data and 2^n time

Side question: is this n-bit security or $2n$-bit security?

- Use a 128-bit primitive: low security
- Design a larger primitive: larger hardware

Authenticated Encryption

Many different ways to build authenticated encryption

Birthday bound security

Most block cipher-based and permutation-based modes
only have birthday bound security

They need a $2n$-bit primitive to resist attacks with 2^n data and 2^n time

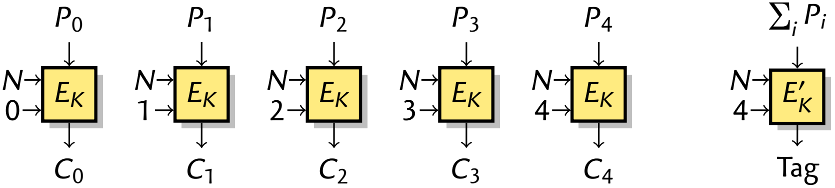
Side question: is this n-bit security or $2n$-bit security?

Beyond birthday security

Tweakable Block Ciphers provide security beyond the birthday bound.

Modes with an n-bit TBC resist attacks with 2^n data and 2^n time.

Tweakable block cipher based AE modes



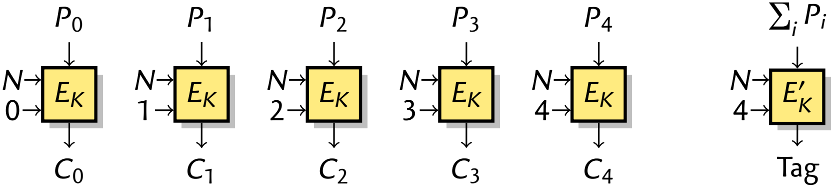
Definition (Tweakable block cipher – Liskov, Rivest, Wagner)

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 idenpendant PRF

- TAE: Tweakable Authenticated Encryption (Liskov, Rivest, Wagner)
  - Nonce-based AEAD, 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 (1TBC); good for small messages
- 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

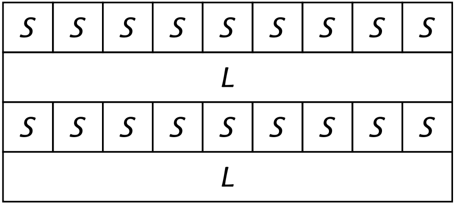
Important remark

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. 2^{nd} 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

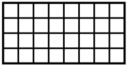
- Mathematical description: SPN network
  - S-boxes
    - With simple gate representation
  - Linear diffusion layer
    - Mixes the full state
    - Binary coefficients
  - Good design criterion: wide-trail



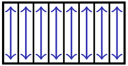
- Bitslice implementation:
  - S-box as a series of bitwise operations with CPU words
  - L-box tables for diffusion layer
  - Easy to mask (simple non-linear ops., complex linear ops.)

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

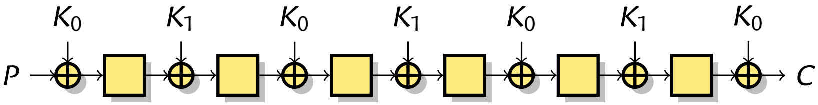
SCREAM S-box and L-box

For SCREAM, we reuse the components of Robin/Fantomas:

- 8-bit S-box
  - Built from 3 smaller S-boxes (Feistel-like structure)
  - $\text{Pr}_{\text{lin}} = 2^{-2}$, $\text{Pr}_{\text{diff}} = 2^{-4}$, 11/12 non-linear gates
- 16-bit L-box
  - Branch number 8 (optimal for a binary matrix)
  - Orthogonal matrix: differential and linear properties equivalent
  - Built from $RM(2,5)$ or $QR[32,16,8]$

Tweak/Key schedule

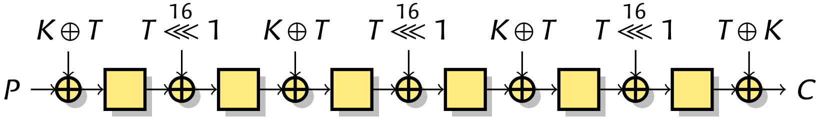
- Robin/Fantomas with a tweak/key schedule
  - 128-bit block
  - 128-bit key
  - 128-bit tweak
- Tweak and key have a similar role (cf. TWEAKEY framework)
- Must be secure against chosen-tweak attacks (≈ related-key)
- Use ideas from LED:



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

iSCREAM: involutive components

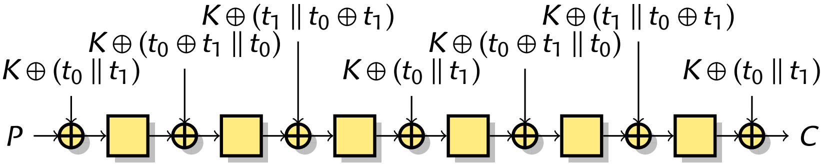
- Tweak every step; key every second step



- Rotation avoids optimal trails with tweak difference
  - $\Delta \to \Delta$: 8 active S-Boxes (involution)
  - $\Delta \to \Delta \overset{16}{\lll} 1$: 12 active S-Boxes

SCREAM: non-involutive components

- Key-schedule based on a $[3,2,2]_4$ code.
  - Two consecutive subkeys cannot be inactive (with related key).
  - Tweak difference gives the same *truncated* difference in all subkeys.

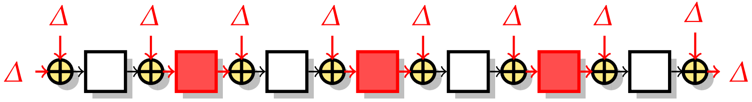


- Optimize L-box to avoid specific trails
  - 1-R trails $\Delta \to \Delta$ have at least 14 active S-boxes
  - RK trails with consecutive active steps are equivalent to SK trails
    - 4-R trail `-xx-` with tweak difference δ
    - $\delta \rightsquigarrow a$, $b \rightsquigarrow \delta$ gives $b \rightsquigarrow \delta \rightsquigarrow a$; at least 20 active S-boxes

Outline

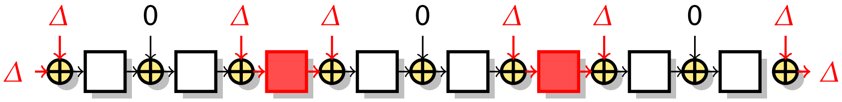
Security against Differential and Linear Cryptanalysis

- Fixed key ⊕ Chosen tweak ≈ Related key
  At least one half of the steps active
- Related key ⊕ Chosen tweak ≈ Related key with more freedom
  At least one half/one third of the steps active (iScream/Scream)
- Wide-trail strategy:
  each active 2-round step has at least 8 active S-boxes.



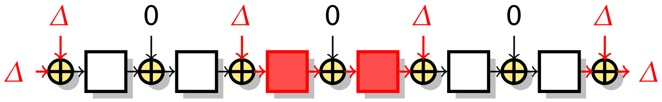
Security against Differential and Linear Cryptanalysis

- Fixed key ⊕ Chosen tweak ≈ Related key
  At least one half of the steps active
- Related key ⊕ Chosen tweak ≈ Related key with more freedom
  At least one half/one third of the steps active (iScream/Scream)
- Wide-trail strategy:
  each active 2-round step has at least 8 active S-boxes.



Security against Differential and Linear Cryptanalysis

- Fixed key ⊕ Chosen tweak ≈ Related key
  At least one half of the steps active
- Related key ⊕ Chosen tweak ≈ Related key with more freedom
  At least one half/one third of the steps active (iScream/Scream)
- Wide-trail strategy:
  each active 2-round step has at least 8 active S-boxes.



Security against Differential and Linear Cryptanalysis

- Fixed key ⊕ Chosen tweak ≈ Related key
  At least one half of the steps active
- Related key ⊕ Chosen tweak ≈ Related key with more freedom
  At least one half/one third of the steps active (iScream/Scream)
- Wide-trail strategy:
  each active 2-round step has at least 8 active S-boxes.

Minimum number of active S-Boxes

Setting	Steps:	1	2	3	4	5	6	7	8	9	10	11	12
Single Key	Scream-10	0	0	8	8	16	16	24	24	**32**			
	iScream-12	0	0	8	8	16	16	24	24	**32**			
Related Key	Scream-12	0	0	8	8	8	16	16	16	24	24	24	**32**
	iScream-14	0	0	8	16	16	16	24	32	32	32	**40**	**40**

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

Minimum number of active S-Boxes

Setting	Steps:	1	2	3	4	5	6	7	8	9	10	11	12
Single Key	Scream-10	0	8	14	20	28	**35**						
	iScream-12	0	8	12	16	24	28	**32**	**40**				
Related Key	Scream-12	0	0	8	14	14	22	28	28	**36**			
	iScream-14	0	0	8	16	16	16	24	**32**	**32**	**32**	**40**	**48**

SCREAM v1 problem

In SCREAM v1, we tried to optimize the use of counters in TAE...

...and failed :-(

In SCREAM v2 we stick to the original TAE.

Thanks

Thanks to Wang Lei and Sim Siang for finding out the mistake!

iSCREAM problem

iSCREAM uses an involutive S-Box and L-Box...

...with some unexpected properties :-(

The strong structure of the involutive L-Box,
combined with low-weight round constants,
allows a self-similarity attack with weak keys or related keys.

We focus on SCREAM at the moment
We plan to redesign iSCREAM in the future

Simple tweak: add full constants

Thanks

Thanks to Henry Gilbert, Gregor Leander, Brice Minaud, Sondre Rønjom for finding out!

Outline

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	Scream	Fantomas	AES-GCM	AES
ARM Cortex A15	23.5	21.8	14.2	31.1	17.8
Atom	56	55	33.3	28.8	17
Nehalem	10.8	9.4	6.3	9.9	6.9
Ivy Bridge ~~AES-NI~~	8.0	7.1	4.2	8.3	5.4
Ivy Bridge AES-NI				2.5	1.3

More detailed benchmarks soon in eBASH...

Implementation: High-end CPUs

Software performance for long messages (cycles/byte)

	SCREAM	Scream	Fantomas	AES-GCM	AES
ARM Cortex A15	23.5	21.8	14.2	31.1	17.8
Atom	56	55	33.3	28.8	17
Nehalem	10.8	9.4	6.3	9.9	6.9
Ivy Bridge ~~AES-NI~~	8.0	7.1	4.2	8.3	5.4
Ivy Bridge AES-NI				2.5	1.3
Haswell ~~AES-NI~~	5.7?	4.7?		??	??
Haswell AES-NI				1.0	0.75

WORK IN PROGRESS

More detailed benchmarks soon in eBASH...

Implementation: High-end CPUs

Software performance for long messages (cycles/byte)

	SCREAM	Scream	Fantomas	AES-GCM	AES
ARM Cortex A15	23.5	21.8	14.2	31.1	17.8
Atom	56	55	33.3	28.8	17
Nehalem	10.8	9.4	6.3	9.9	6.9
Ivy Bridge ~~AES-NI~~	8.0	7.1	4.2	8.3	5.4
Ivy Bridge AES-NI				2.5	1.3
Haswell ~~AES-NI~~	5.7?	4.7?		??	??
Haswell AES-NI				1.0	0.75
Future Intel CPU	AVX512, VPTERNLOG, ...				

WORK IN PROGRESS

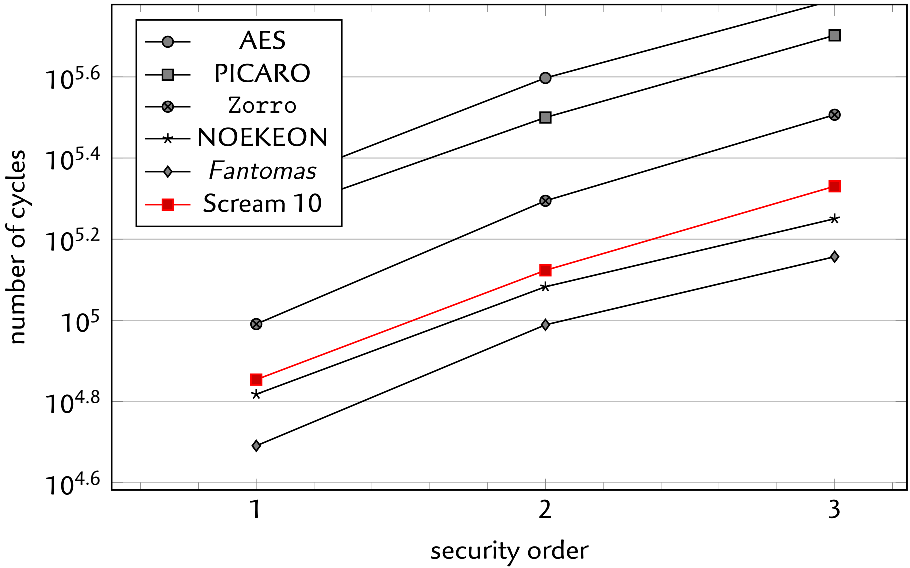
More detailed benchmarks soon in eBASH...

Implementation: AVR micro-controller

- TBC performance: 7650 cycles
  - Using 1kB table
  - Smaller tables if needed

- For many embedded devices, side-channel attack are a real threat
- SCREAM has very good performances for masked implementations
  - Noekeon also very good (similar components)

Implementation: AVR micro-controller



Implementation: Hardware

- We study implementations with a 128-bit datapath
  - Reasonable price/performance ration
- Low amount of logic in one round
  - We can unroll one full step per clock cycle
  - One step ≈ one AES round
  - SCREAM TBC ≈ AES
- Low overhead for TAE mode
  - Limited extra memory: small total state

Implementation: Hardware

Hardware performance of the TBC: ASIC

	Cycle	Mode E,D,ED	Area [μm^2]	f_{max} [MHz]	Latency [cycles]	Throughput [Mbps]
AES	1R	E	17921	444	12	4740
		D	20292	377	22	2195
		ED	24272	363	≈17	≈2997
Scream-10	1R	E	12951	751	21	4577
		D	*12951*	*751*	21	*4577*
		ED	*17292*	*751*	21	*4577*
Scream-10	2R	E	17292	446	11	5190
		D	*17292*	*446*	11	*5190*
		ED	*25974*	*446*	11	*5190*

Implementation: Hardware

Hardware performance of the TBC / full mode: Virtex 6 FPGA

	Cycle	Slices [slices]	BRAM [$\times 18k$]	f_{max} [MHz]	Latency [cycles]	Throughput [Mbps]
`AES`	1R	562	–	211	11	2450
		136	10	308	11	3585
Scream-10	1R	251	–	321	20	2050
		167	16	287	20	1836
	2R	416	–	193	10	2470
		190	16	278	10	2965
SCREAM-10	1R	512	–	302	$20 \cdot (\ell + 1)$	1932
	2R	571	–	146	$10 \cdot (\ell + 1)$	1870

Implementation: overview

- Hardware:
  - The tweakable block cipher costs about the same as AES
  - Low overhead for TAE mode (limited extra memory)
  - Parallelism can be leveraged in a pipelined implementation
- Micro-controller:
  - Good performance ($<$ 8k cycles)
  - Very good if masking is needed
- High-end CPU
  - Parallelism exploited with SIMD
  - Performance similar to AES-GCM

    (excluding hardware AES instructions)

SCREAM Features

TAE Mode

- Nonce-based AEAD
- Fully parallelizable
- 128-bit security
- 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

Small tweaks to fix initial mistakes

- The tweakable block cipher is also a useful primitive in itself.

Extra Slides

FPGA implementation results

FPGA implementation results

Tweakable Block Cipher:

For **Virtex 6** (XC6 VLX 240T - 3 FF1156):

	DP size	BRAMs[1]	UNROLL[1]	REG_O[1]	Cycles[3]	Timing performance strategy				Area reduction strategy			
						Regs/LUTs	Slices	BRAMs	F_{max}	Regs/LUTs	Slices	BRAMs	F_{max}
SCREAM 128 bit	**128**	**F**	**F**	**---**	**20**	**404/823**	**251**	**0**	**321**	**400/640**	**187**	**0**	**286**
	128	**F**	**T**	**---**	**10**	**399/1520**	**416**	**0**	**193**	**398/1033**	**282**	**0**	**153**
	128	T	F	T	20	401/629	205	8x18k	287	400/479	147	8x18k	261
	128	**T**	**F**	**F**	**20**	**273/609**	**167**	**8x18k**	**287**	**273/460**	**126**	**8x18k**	**261**
	128	T[2]	T	T	10	398/670	177	16x18k	277	398/665	204	16x18k	252
	128	**T[2]**	**T**	**F**	**10**	**271/667**	**190**	**16x18k**	**278**	**271/643**	**201**	**16x18k**	**252**
SCREAM 16b	16	F	F	---	320	780/643	222	0	400	260/359	107	0	237
AES1	128	F	F	---	11	686/2317	815	0	211	526/1431	398	0	154
AES2	128	F	F	---	11	619/1712	562	0	211	398/1430	392	0	154
AES3	128	T	F	---	11	398/481	136	10x18k	308	398/468	152	10x18k	284
AES4	128	T	F	---	11	398/476	163	10x18k	308	270/450	133	10x18k	285

Notes:

[1] Parameter settings: T = True; F = False; --- = not applicable

[2] BRAMs operate on 2x higher clock frequency than the rest of the core

[3] Key initialization requires extra 1 clock cycle for 128b version or 8 clock cycles for 16b version

FPGA implementation results

Authenticated Encryption (full mode)

	DP size	TRUNC	PADD	UNROLL	Cycles	Timing performance strategy				Area reduction strategy			
						Regs/LUTs	Slices	BRAMs	F_{max}	Regs/LUTs	Slices	BRAMs	F_{max}
SCREAM TAE 128 bit	128	T	T	T	X	917/2193	571	0	146	917/1755	459	0	154
	128	T	T	F	Y	920/1932	512	0	302	919/1392	363	0	289
	128	T	F	T	X	918/2109	567	0	150	917/1766	458	0	149
	128	T	F	F	Y	920/1588	414	0	286	919/1392	362	0	312

X = (A + P + 1)*10 + 2; Y = (A + P + 1)*20 + 2; A - number of 128b blocks of associated data, P - number of 128b blocks of the plaintext