Block Ciphers that are Easier to Mask How Far Can we Go ?

Benoît Gérard, Vincent Grosso,
María Naya-Plasencia, **François-Xavier Standaert**

DGA & UCL Crypto Group & INRIA

CHES 2013
Santa Barbara, USA

Block ciphers

- Trojan horses of modern cryptography
  - Used for encryption, authentication, hashing

Block ciphers

- Trojan horses of modern cryptography
  - Used for encryption, authentication, hashing

- Well known standards (DES, AES)

Block ciphers

- Trojan horses of modern cryptography
  - Used for encryption, authentication, hashing
- Well known standards (DES, AES)
- Active research in lightweight designs
  - TEA, NOEKEON, SERPENT, ICEBERG, HIGHT, mCrypton, SEA, PRESENT, KATAN, MIBS, LED, Piccolo, Lblock, KLEIN, PRINCE, …

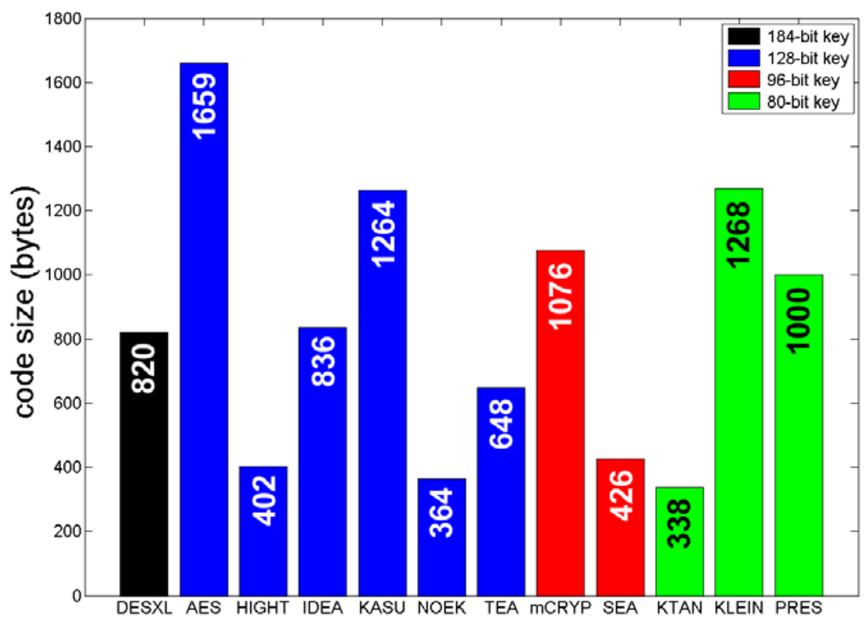
Block ciphers

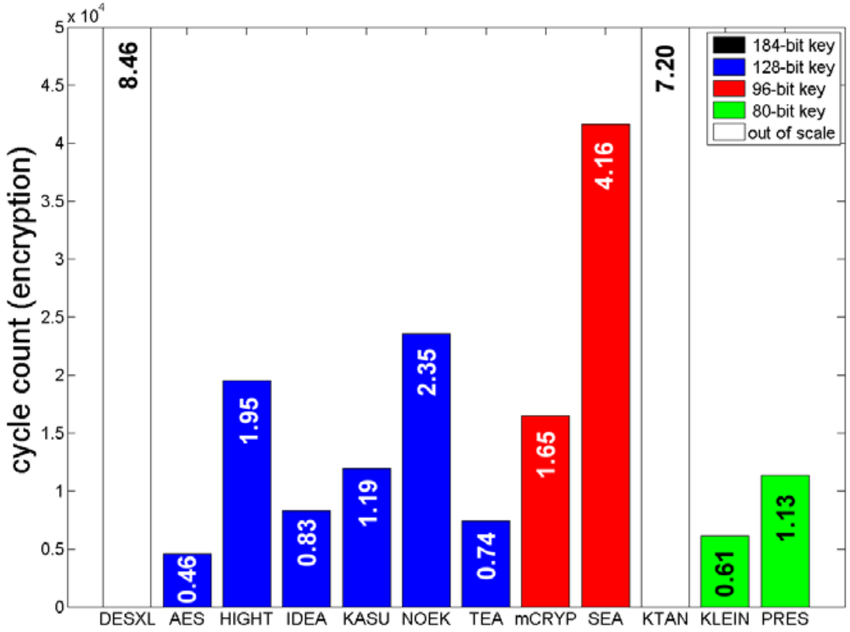
- Trojan horses of modern cryptography
  - Used for encryption, authentication, hashing

- Well known standards (DES, AES)

- Active research in lightweight designs
  - TEA, NOEKEON, SERPENT, ICEBERG, HIGHT, mCrypton, SEA, PRESENT, KATAN, MIBS, LED, Piccolo, Lblock, KLEIN, PRINCE, …
  - Optimized for various performance criteria
    - Code size, throughput, gate count, energy, …

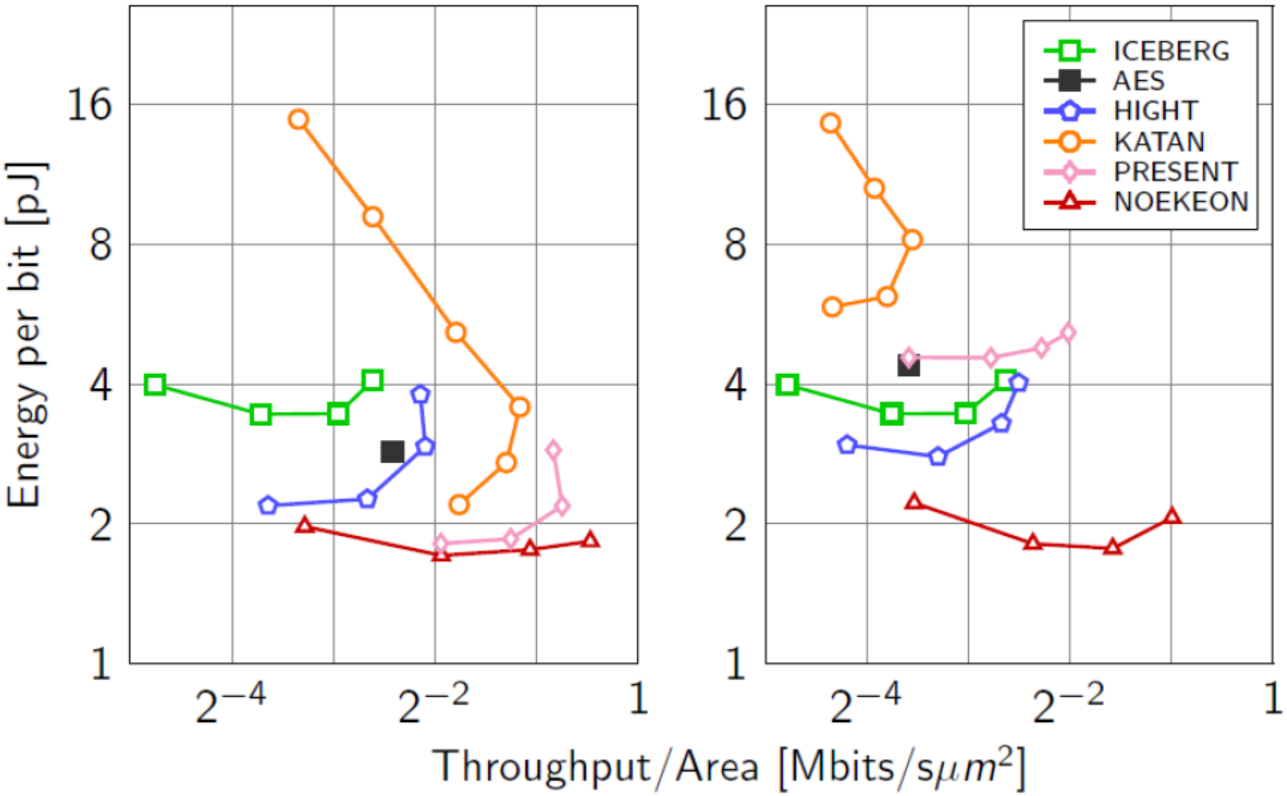
Lessons learned (Atmel AVR case)





- Different designs ≈ different tradeoffs

Lessons learned (ASIC case)



- Different designs ≈ different tradeoffs
- Similar design principles (e.g. wide-trail strategy) lead to similar “efficiencies” (security is the limit)

Masking

- aka secret sharing – see previous talk
  - Most investigated countermeasure against SCAs
  - Main idea: split the sensitive data in r shares

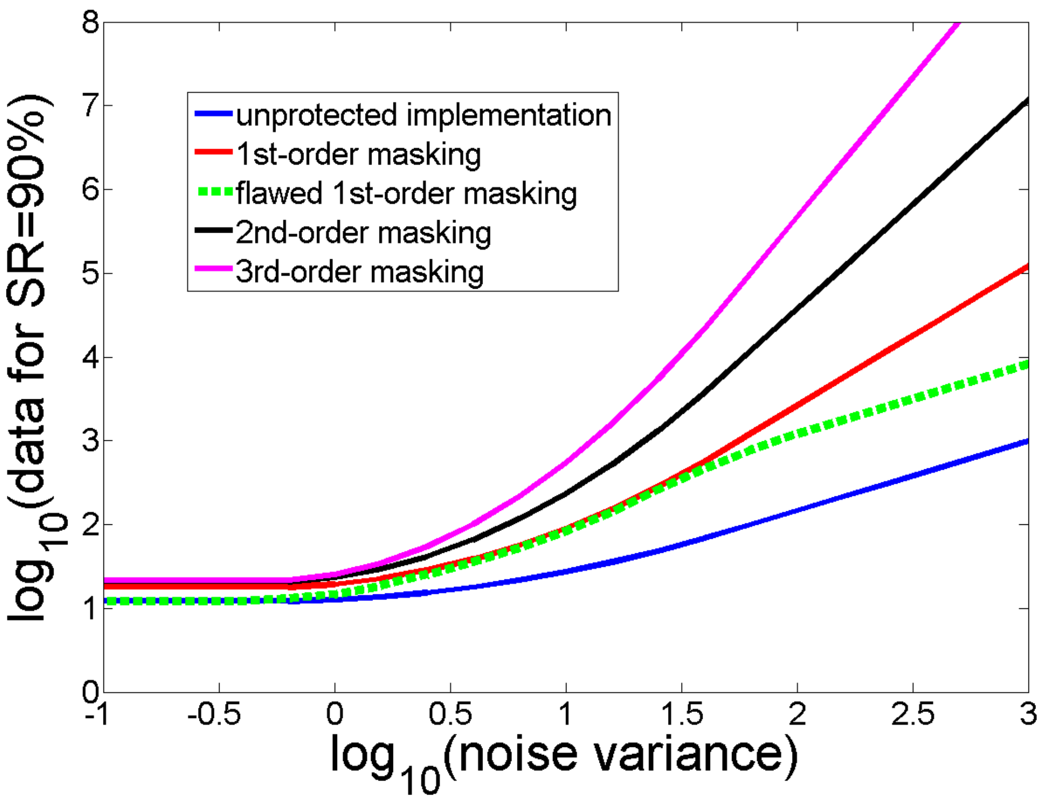
Masking

- aka secret sharing – see previous talk
  - Most investigated countermeasure against SCAs
  - Main idea: split the sensitive data in *r* shares

- If perfect implementation, the data complexity to break masking is proportional to $(\sigma_n^2)^r$
  - Perfect ~ if the smallest-order key-dependent moment in the leakage distribution is *r*
  - Essentially depends on physical assumptions
    - Difficult in hardware (glitches, …)
    - Easier in software (time separation)

Masking

- aka secret sharing – see previous talk
  - Most investigated countermeasure against SCAs
  - Main idea: split the sensitive data in *r* shares



Lessons learned

- Goals are similar to (but not the same as) MPC
  - Linear operations are easy to perform
    - Masks can be propagated independently

Lessons learned

- Goals are similar to (but not the same as) MPC
  - Linear operations are easy to perform
    - Masks can be propagated independently

  - Non-linear operations are more expensive
    - Need interaction (and randomness)
    - Implementation cost increases with r^2

Lessons learned

- Goals are similar to (but not the same as) MPC
  - Linear operations are easy to perform
    - Masks can be propagated independently

  - Non-linear operations are more expensive
    - Need interaction (and randomness)
    - Implementation cost increases with r^2

- Given a block cipher (e.g. the AES), it is usually possible to implement masking “quite” efficiently
  - *By finding the best representation*
    - e.g. [RP10,PR11]: AES S-box ≈ 4 multiplications

Research problem

- Does it make sense to “reverse” the question, i.e. design a block cipher that is efficient to mask?

Research problem

- Does it make sense to “reverse” the question, i.e. design a block cipher that is efficient to mask?

- Previous work: PIretCArletROche (ACNS 2011)
  - Mostly focused in the S-box selection
    - Feistel structure + non-bijective S-box

Research problem

- Does it make sense to “reverse” the question, i.e. design a block cipher that is efficient to mask?

- Previous work: PIretCArletROche (ACNS 2011)
  - Mostly focused in the S-box selection
    - Feistel structure + non-bijective S-box

- Interesting approach but…
  - Non-bijective S-boxes are bad choice for SCA-resistance (because they allow generic attacks)

Can we do better?

Can we do better?

- Re-using the AES rounds as much as possible
  - Most investigated cipher for physical attacks

Can we do better?

- Re-using the AES rounds as much as possible
  - Most investigated cipher for physical attacks

- Keeping bijective S-boxes
  - That can be represented with less multiplications

Can we do better?

- Re-using the AES rounds as much as possible
  - Most investigated cipher for physical attacks
- Keeping bijective S-boxes
  - That can be represented with less multiplications
- Reducing the total number of S-boxes
  - Taking advantage of strong diffusion

Can we do better?

- Re-using the AES rounds as much as possible
  - Most investigated cipher for physical attacks
- Keeping bijective S-boxes
  - That can be represented with less multiplications
- Reducing the total number of S-boxes
  - Taking advantage of strong diffusion
- Excluding related keys for now
  - As most lightweight ciphers

Outline: the cipher Zorro

1. Which S-boxes?
2. How many S-boxes?
3. Key scheduling
4. Putting things together

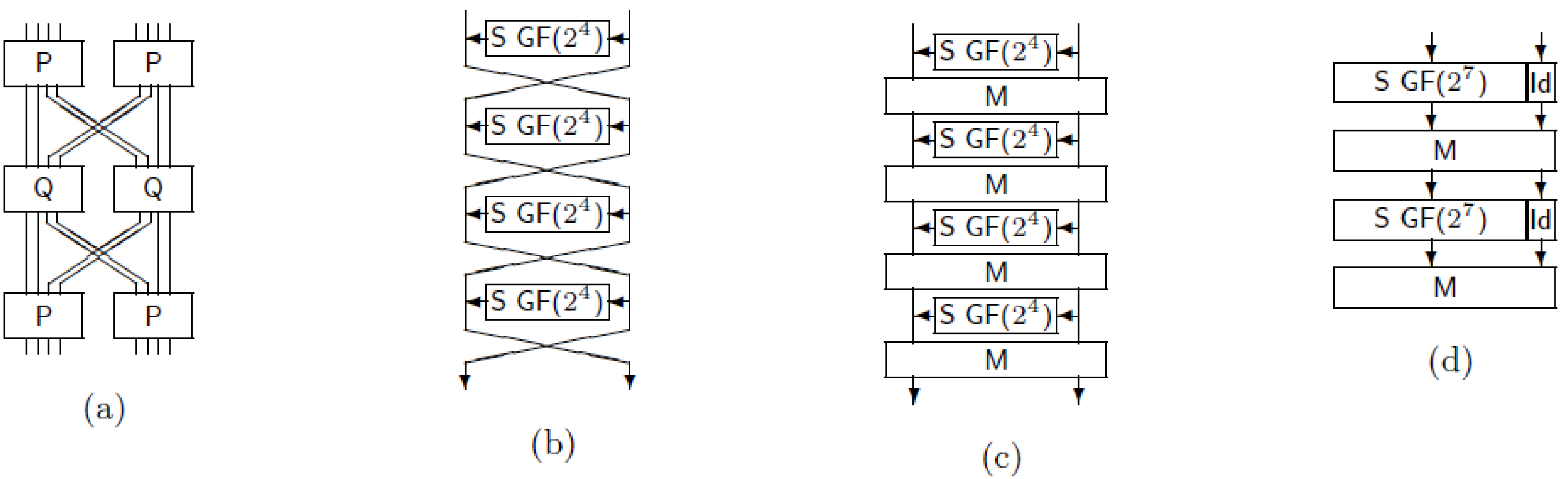
1. Which S-boxes?

- Goal: reduce the number of multiplications (keeping decent linear/differential/algebraic properties)
  - AES S-box: 4 multiplications, max(WS)=32, max(DS) = 4, algebraic degree = 7

1. Which S-boxes?

- Goal: reduce the number of multiplications (keeping decent linear/differential/algebraic properties)
  - AES S-box: 4 multiplications, max(WS)=32, max(DS) = 4, algebraic degree = 7

- Monomials/binomials in GF(2^8): exhaustive search
- Others S-boxes: “informed search”, e.g.



Results

	required randomness (bit)			# sec. mult.	additional operations	security properties		
	$d = 1$	$d = 2$	d			$deg(S)$	$\max \Delta_S$	$\max \Omega_S$
AES [33]	48	128	$16d^2 + 32d$	4 (GF(2^8))	7 squ. + 1 Diff. matrix	7	4	32
AES [19]	32	84	$10d^2 + 22d$	5 (GF(2^4))	3 squ. + 5 Diff. matrix	7	4	32
PICARO	16	48	$8d^2 + 8d$	4 (GF(2^4))	2 squ.	4	4	68
X^7	24	64	$8d^2 + 16d$	2 (GF(2^8))	2 squ. + 1 Diff. matrix	3	6	64
X^{29}	32	88	$12d^2 + 20d$	3 (GF(2^8))	4 squ. + 1 Diff. matrix	4	10	64
X^{37}	24	64	$8d^2 + 16d$	2 (GF(2^8))	5 squ. + 1 Diff. matrix	3	6	64
$8X^{97} + X^{12}$	32	80	$8d^2 + 24d$	2 (GF(2^8))	6 squ. + 1 Diff. matrix	3	6	48
$155X^7 + X^{92}$	40	104	$12d^2 + 28d$	3 (GF(2^8))	8 squ. + 1 Diff. matrix	4	6	48
Ex. 1	32	80	$8d^2 + 24d$	4 (GF(2^4))	4 squ. + 4 Diff. matrix	7	10	64
Ex. 2	48	112	$8d^2 + 40d$	4 (GF(2^4))	28 squ. + 4 Diff. matrix	6	8	64
Ex. 3	28	70	$7d^2 + 21d$	2 (GF(2^7))	2 squ. + 2 Diff. matrix	4	10	64

Results

	required randomness (bit)			# sec. mult.	additional operations	security properties		
	$d=1$	$d=2$	d			$deg(S)$	max Δ_S	max Ω_S
AES [33]	48	128	$16d^2+32d$	4 (GF(2^8))	7 squ. + 1 Diff. matrix	7	4	32
AES [19]	32	84	$10d^2+22d$	5 (GF(2^4))	3 squ. + 5 Diff. matrix	7	4	32
PICARO	16	48	$8d^2+8d$	4 (GF(2^4))	2 squ.	4	4	68
X^7	24	64	$8d^2+16d$	2 (GF(2^8))	2 squ. + 1 Diff. matrix	3	6	64
X^{29}	32	88	$12d^2+20d$	3 (GF(2^8))	4 squ. + 1 Diff. matrix	4	10	64
X^{37}	24	64	$8d^2+16d$	2 (GF(2^8))	5 squ. + 1 Diff. matrix	3	6	64
$8X^{97}+X^{12}$	32	80	$8d^2+24d$	2 (GF(2^8))	6 squ. + 1 Diff. matrix	3	6	48
$155X^7+X^{92}$	40	104	$12d^2+28d$	3 (GF(2^8))	8 squ. + 1 Diff. matrix	4	6	48
Ex. 1	32	80	$8d^2+24d$	4 (GF(2^4))	4 squ. + 4 Diff. matrix	7	10	64
Ex. 2	48	112	$8d^2+40d$	4 (GF(2^4))	28 squ. + 4 Diff. matrix	6	8	64
Ex. 3	28	70	$7d^2+21d$	2 (GF(2^7))	2 squ. + 2 Diff. matrix	4	10	64

Our choice: same # of multiplications as PICARO

2. How many S-boxes?

- AES (very) strong against statistical attacks

2. How many S-boxes?

- AES (very) strong against statistical attacks
- Can we reduce the total # of S-boxes (taking advantage of strong diffusion properties)?

2. How many S-boxes?

- AES (very) strong against statistical attacks
- Can we reduce the total # of S-boxes (taking advantage of strong diffusion properties)?
- Answer: mainly depends on the permutation layer
  - e.g. not possible with wire crossings (see paper)

2. How many S-boxes?

- AES (very) strong against statistical attacks
- Can we reduce the total # of S-boxes (taking advantage of strong diffusion properties)?
- Answer: mainly depends on the permutation layer
  - e.g. not possible with wire crossings (see paper)
- What can we do with MixColumns?

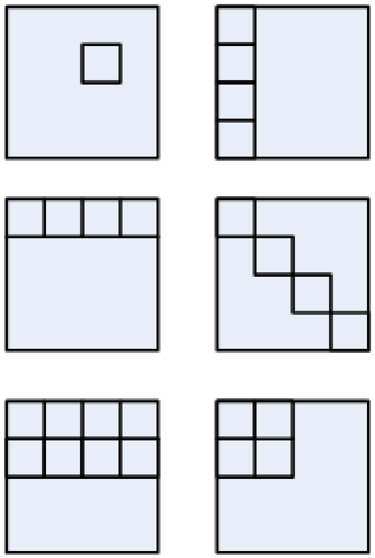
2. How many S-boxes?

- AES (very) strong against statistical attacks
- Can we reduce the total # of S-boxes (taking advantage of strong diffusion properties)?
- Answer: mainly depends on the permutation layer
  - e.g. not possible with wire crossings (see paper)
- What can we do with MixColumns?

- Informal tests: how many rounds for
  - At least going through one S-box
  - All output bytes having a non-linear term
  - Input diffs. with non-linear effect on output bytes

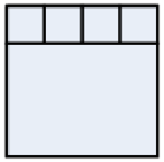
Testing different configurations

	NrSbox	NrNlin	NrDiff
1 S-box	3	2	4
4 S-boxes, 1 line	2	1	3
8 S-boxes, 2 lines	2	1	3
4 S-boxes, 1 column	3	1	3
4 S-boxes, 1 diagonal	2	2	3
4 S-boxes, 1 per column	2	2	3
4 S-boxes, Square	3	2	4



Testing different configurations

	NrSbox	NrNlin	NrDiff
1 S-box	3	2	4
4 S-boxes, 1 line	2	1	3
8 S-boxes, 2 lines	2	1	3
4 S-boxes, 1 column	3	1	3
4 S-boxes, 1 diagonal	2	2	3
4 S-boxes, 1 per column	2	2	3
4 S-boxes, Square	3	2	4



Our choice: 4 S-boxes on the first state line

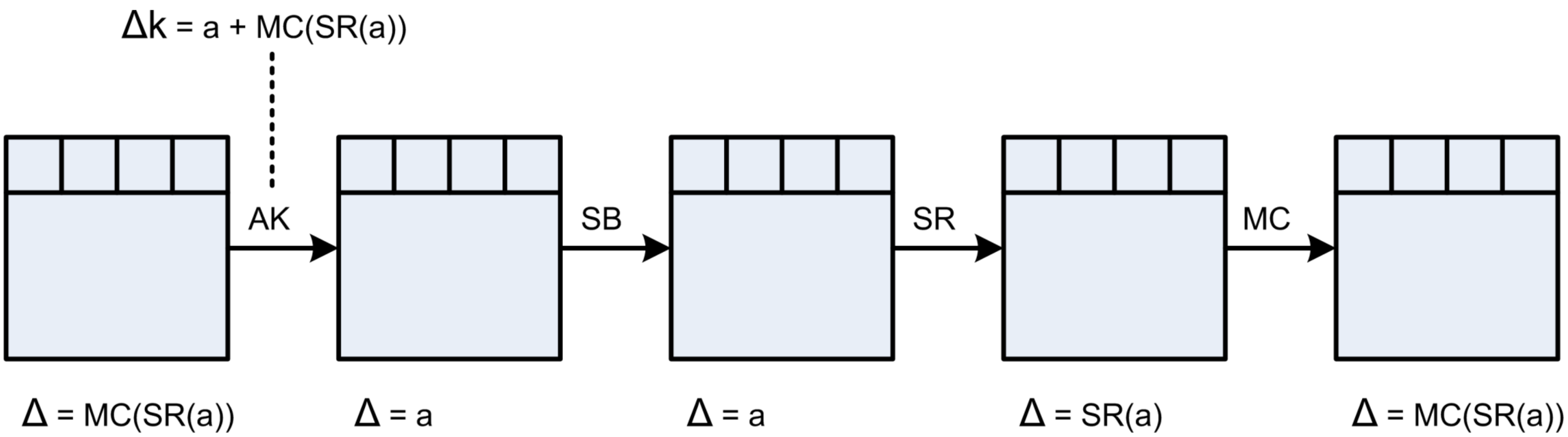
3. Key scheduling

- Minimalism (Mutliple Even-Mansour, LED, …)
- Main question: key addition every *???* rounds

3. Key scheduling

- Minimalism (Mutliple Even-Mansour, LED, ...)
- Main question: key addition every ??? rounds

- Example: every single round => related-key issue



Intuition

- Property so strong that it leads to non-related-key attacks with 2^64 data and 2^64 time
  - (*thanks to Dmitry Khovratovich*!)

Intuition

- Property so strong that it leads to non-related-key attacks with 2^64 data and 2^64 time
  - (*thanks to Dmitry Khovratovich*!)

=> Key addition should be performed after a “complex enough” function of the state (we choose 4 rounds)

Intuition

- Property so strong that it leads to non-related-key attacks with 2^64 data and 2^64 time
  - (*thanks to Dmitry Khovratovich*!)

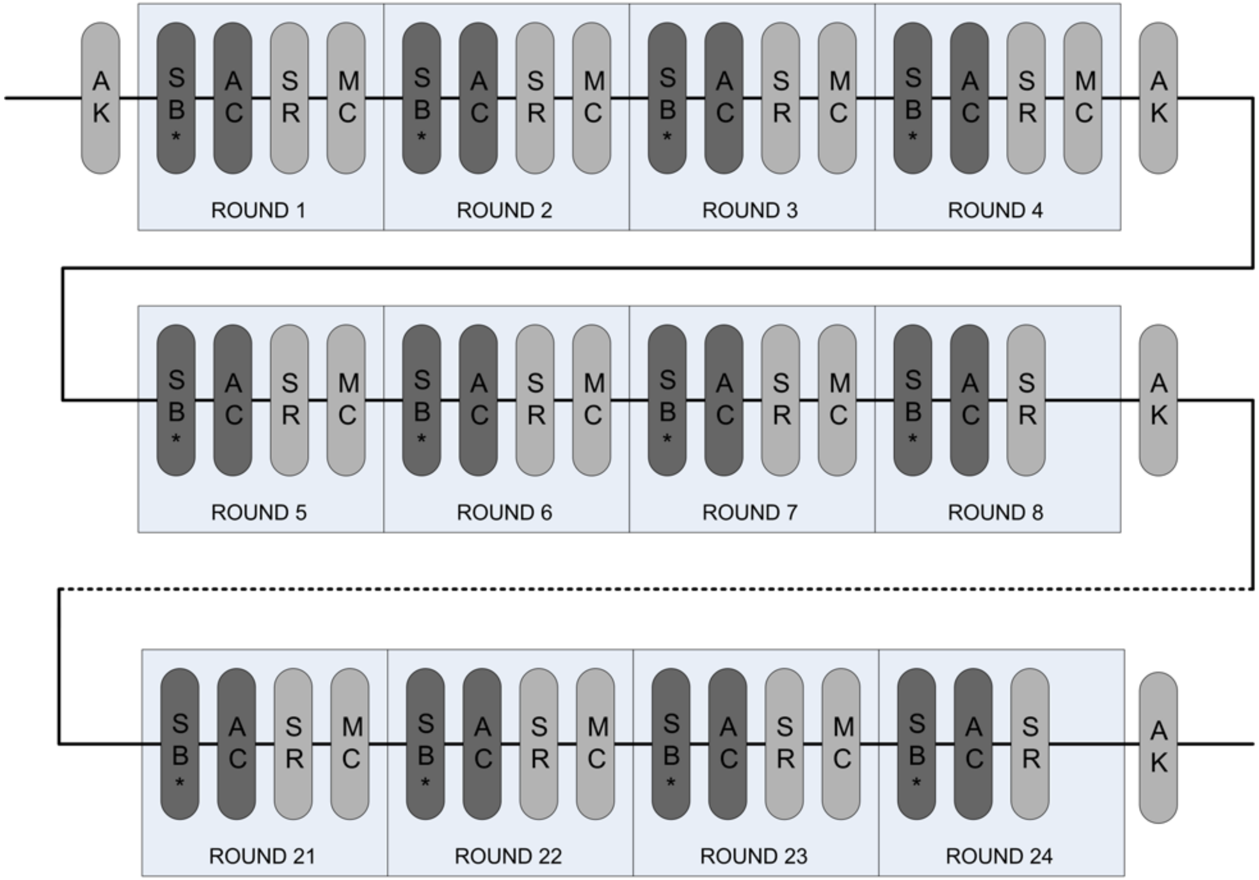
=> Key addition should be performed after a "complex enough" function of the state (we choose 4 rounds)

… and a sufficient number of times to avoid generic attacks against Even-Mansour schemes (we choose 7)

- cfr. Asiacrypt 2012 and 2013
  - (*thanks to Orr Dunkelman*!)

4. Putting things together

- Number of rounds: 24 (6 steps of 4 rounds)
  - Roughly divides the total # of multiplications by 4!



Security analysis (ePrint version)

- Non trivial (frequently exploiting results from hash function cryptanalysis and SHA3 competition)

Security analysis (ePrint version)

- Non trivial (frequently exploiting results from hash function cryptanalysis and SHA3 competition)

- Linear/differential cryptanalysis: bounds on the best characteristics for 16/14 rounds
  - By exploiting degrees of freedom
    - # active S-boxes = 4 Nr - 31

Security analysis (ePrint version)

- Non trivial (frequently exploiting results from hash function cryptanalysis and SHA3 competition)

- Linear/differential cryptanalysis: bounds on the best characteristics for 16/14 rounds
  - By exploiting degrees of freedom
    - # active S-boxes = 4 Nr - 31

- Impossible differential attack for 10 rounds
- Rebound *distinguisher* for 12 rounds

Security analysis (ePrint version)

- Non trivial (frequently exploiting results from hash function cryptanalysis and SHA3 competition)

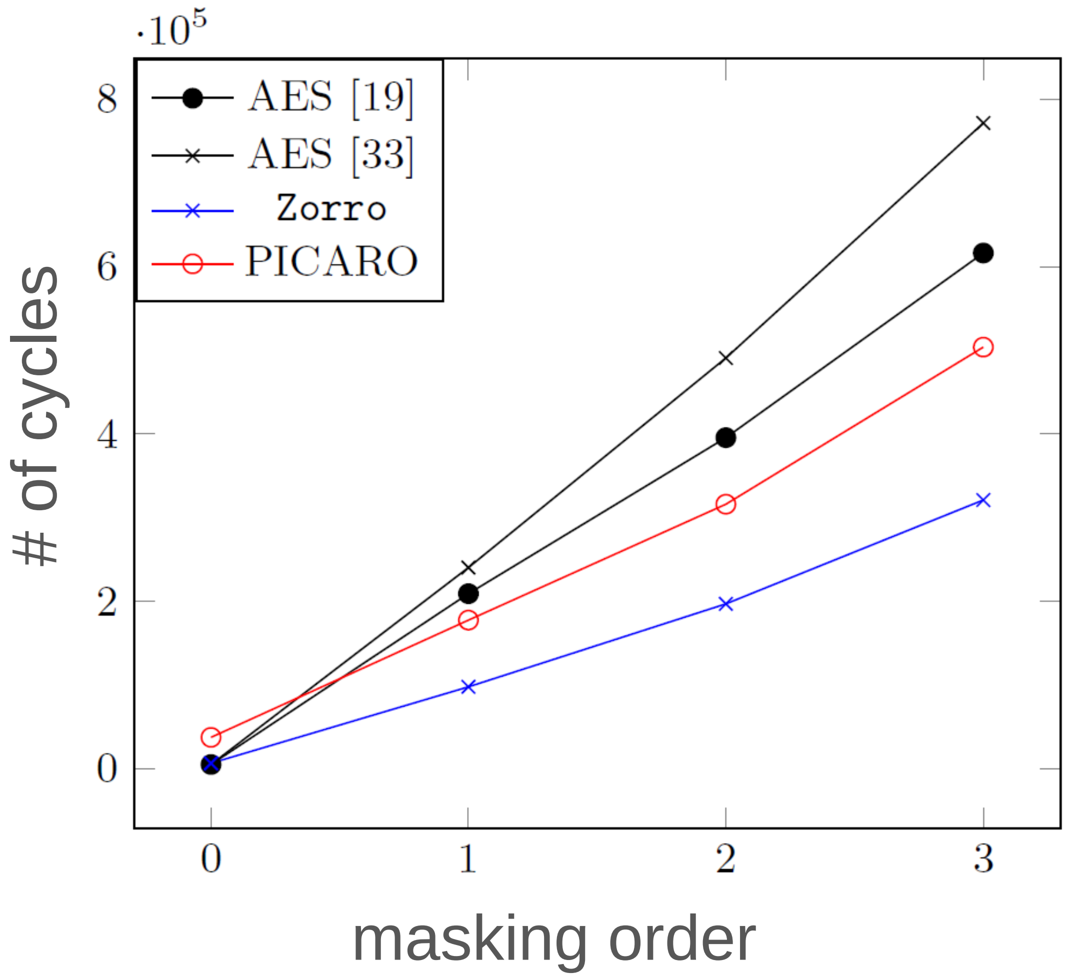
- Linear/differential cryptanalysis: bounds on the best characteristics for 16/14 rounds
  - By exploiting degrees of freedom
    - # active S-boxes = 4 Nr - 31

- Impossible differential attack for 10 rounds
- Rebound *distinguisher* for 12 rounds

- (+ truncated differential, cube testers, MITM, …)

Performance evaluation

- Case study: Atmel AtMega644p



Conclusions

- Significant performance gains compared to AES
    - For Boolean & polynomial masking

Conclusions

- Significant performance gains compared to AES
  - For Boolean & polynomial masking
- “Non-regular” design exploiting strong diffusion
  => New analysis techniques for block ciphers
  => Not only S-boxes matter !

Conclusions

- Significant performance gains compared to AES
  - For Boolean & polynomial masking
- “Non-regular” design exploiting strong diffusion

  => New analysis techniques for block ciphers

  => Not only S-boxes matter !
- Interesting target for cryptanalysis?

Conclusions

- Significant performance gains compared to AES
  - For Boolean & polynomial masking
- "Non-regular" design exploiting strong diffusion
  => New analysis techniques for block ciphers
  => Not only S-boxes matter !
- Interesting target for cryptanalysis?
- Next: moving away from the AES?
  - Stronger diffusion (Khazad-like) or smaller S-boxes (NOEKEON, PRESENT, …)?

Conclusions

- Significant performance gains compared to AES
  - For Boolean & polynomial masking
- “Non-regular” design exploiting strong diffusion
  => New analysis techniques for block ciphers
  => Not only S-boxes matter !
- Interesting target for cryptanalysis?
- Next: moving away from the AES?
  - Stronger diffusion (Khazad-like) or smaller S-boxes (NOEKEON, PRESENT, …)?
- Or specialize to Boolean masking only (=> bitslice)

THANKS

http://perso.uclouvain.be/fstandae/