Radboud University

Optimising masking costs of CAESAR candidates

Ko Stoffelen & Lejla Batina

Digital Security, Radboud University
k.stoffelen@cs.ru.nl

DIAC 2015
September 29, 2015

Masking

- Countermeasure against side-channel analysis
- Arithmetic vs. Boolean
- Costs factor 2–8 in terms of cycles [Mes01]
- Costs grow quadratically for nonlinear gates, e.g.:

$$
\begin{aligned}
z = x \wedge y \quad \rightarrow \quad & [x' = x \oplus x_m] \\
& [y' = y \oplus y_m] \\
& z' = x' \wedge y' \\
& z_m = (x_m \wedge y') \oplus (y_m \wedge x') \oplus (x_m \wedge y_m)
\end{aligned}
$$

Goal

- How can the costs of applying masking countermeasures to ciphers be reduced?

Goal

- How can the costs of applying masking countermeasures to ciphers be reduced?
  - By reducing nonlinear operations?
  - By design?

Context – CAESAR competition

ACORN	++AE	AEGIS	AES-CMCC	AES-COBRA
AES-COPA	AES-CPFB	AES-JAMBU	AES-OTR	AEZ
Artemia	Ascon	AVALANCHE	Calico	CBA
CBEAM	CLOC	Deoxys	ELmD	Enchilada
FASER	HKC	HS1-SIV	ICEPOLE	iFeed[AES]
Joltik	Julius	Ketje	Keyak	KIASU
LAC	Marble	McMambo	Minalpher	MORUS
NORX	OCB	OMD	PAEQ	PAES
PANDA	π-Cipher	POET	POLAWIS	PRIMATEs
Prøst	Raviyoyla	Sablier	SCREAM	SHELL
SILC	Silver	STRIBOB	Tiaoxin	TriviA-ck
Wheesht	YAES			

Context – CAESAR competition

ACORN	++AE	AEGIS	AES-CMCC	AES-COBRA
AES-COPA	AES-CPFB	AES-JAMBU	AES-OTR	AEZ
Artemia	Ascon	AVALANCHE	Calico	CBA
CBEAM	CLOC	Deoxys	ELmD	Enchilada
FASER	HKC	HS1-SIV	ICEPOLE	iFeed[AES]
Joltik	Julius	Ketje	Keyak	KIASU
LAC	Marble	McMambo	Minalpher	MORUS
NORX	OCB	OMD	PAEQ	PAES
PANDA	π-Cipher	POET	POLAWIS	PRIMATEs
Prøst	Raviyoyla	Sablier	SCREAM	SHELL
SILC	Silver	STRIBOB	Tiaoxin	TriviA-ck
Wheesht	YAES			

Context – CAESAR competition

ACORN	++AE	AEGIS	AES-CMCC	AES-COBRA
AES-COPA	AES-CPFB	AES-JAMBU	AES-OTR	AEZ
Artemia	Ascon	AVALANCHE	Calico	CBA
CBEAM	CLOC	Deoxys	ELmD	Enchilada
FASER	HKC	HS1-SIV	ICEPOLE	iFeed[AES]
Joltik	Julius	Ketje	Keyak	KIASU
LAC	Marble	McMambo	Minalpher	MORUS
NORX	OCB	OMD	PAEQ	PAES
PANDA	π-Cipher	POET	POLAWIS	PRIMATEs
Prøst	Raviyoyla	Sablier	SCREAM	SHELL
SILC	Silver	STRIBOB	Tiaoxin	TriviA-ck
Wheesht	YAES			

Context – CAESAR competition

(S-boxes of)

8x8	5x5	4x4
AES	Ascon	Joltik
AES^{-1}	ICEPOLE	Joltik^{-1}
iSCREAM	Ketje/Keyak	LAC
SCREAM	PRIMATE	Minalpher
SCREAM^{-1}	PRIMATE^{-1}	Prøst
		RECTANGLE
		RECTANGLE^{-1}

Optimising masking costs

Nonlinear operations

Multiplicative complexity (MC)

- Most nonlinear operations in the nonlinear part of the primitive: the S-box

Multiplicative complexity (MC)

- Most nonlinear operations in the nonlinear part of the primitive: the S-box
- MC: minimal number of `AND`/`OR` gates required to implement function

Multiplicative complexity (MC)

- Most nonlinear operations in the nonlinear part of the primitive: the S-box
- MC: minimal number of `AND`/`OR` gates required to implement function
- Goal is to compute the MC of CAESAR S-boxes

Minimizing AND/OR gates

- Existing logic synthesis tools not very helpful
  - E.g. Espresso, SIS, misII, Logic Friday, ABC, ...
- Instead: convert to SAT and let SAT solvers do the work
- Converting problem to SAT nontrivial, but feasible [CHM11, Mou15]

Reducing decisional MC to SAT

$$
\begin{aligned}
q_0 &= a_0 + a_1 \cdot x_0 + a_2 \cdot x_1 + a_3 \cdot x_2 + a_4 \cdot x_3 \\
q_1 &= a_5 + a_6 \cdot x_0 + a_7 \cdot x_1 + a_8 \cdot x_2 + a_9 \cdot x_3 \\
t_0 &= q_0 \cdot q_1 \\
q_2 &= a_{10} + a_{11} \cdot x_0 + a_{12} \cdot x_1 + a_{13} \cdot x_2 + a_{14} \cdot x_3 + a_{15} \cdot t_0 \\
q_3 &= a_{16} + a_{17} \cdot x_0 + a_{18} \cdot x_1 + a_{19} \cdot x_2 + a_{20} \cdot x_3 + a_{21} \cdot t_0 \\
t_1 &= q_2 \cdot q_3 \\
q_4 &= a_{22} + a_{23} \cdot x_0 + a_{24} \cdot x_1 + a_{25} \cdot x_2 + a_{26} \cdot x_3 + a_{27} \cdot t_0 + a_{28} \cdot t_1 \\
q_5 &= a_{29} + a_{30} \cdot x_0 + a_{31} \cdot x_1 + a_{32} \cdot x_2 + a_{33} \cdot x_3 + a_{34} \cdot t_0 + a_{35} \cdot t_1 \\
t_2 &= q_4 \cdot q_5 \\
y_0 &= a_{36} x_0 + a_{37} \cdot x_1 + a_{38} \cdot x_2 + a_{39} \cdot x_3 + a_{40} \cdot t_0 + a_{41} \cdot t_1 + a_{42} \cdot t_2 \\
y_1 &= a_{43} x_0 + a_{44} \cdot x_1 + a_{45} \cdot x_2 + a_{46} \cdot x_3 + a_{47} \cdot t_0 + a_{48} \cdot t_1 + a_{49} \cdot t_2 \\
y_2 &= a_{50} x_0 + a_{51} \cdot x_1 + a_{52} \cdot x_2 + a_{53} \cdot x_3 + a_{54} \cdot t_0 + a_{55} \cdot t_1 + a_{56} \cdot t_2 \\
y_3 &= a_{57} x_0 + a_{58} \cdot x_1 + a_{59} \cdot x_2 + a_{60} \cdot x_3 + a_{61} \cdot t_0 + a_{62} \cdot t_1 + a_{63} \cdot t_2
\end{aligned}
$$

Our work

- Generate logic formulas in ANF for given S-box and MC
- Convert ANF to CNF
- Let MiniSAT, CryptoMiniSAT, Plingeling, Treengeling do the work on big machine
- Translate back to S-box implementation

Results

S-box	MC	S-box	MC
AES	≤ 32 [BP10]	PRIMATE^{-1}	$\in \{6,7,8,9,10\}$*
AES^{-1}	≤ 32 [BP10]	Joltik	4
iSCREAM	≤ 12 [GLSV14]	Joltik^{-1}	4*
SCREAM	≤ 12 [GLS^{+}15]	LAC	4*
SCREAM^{-1}	≤ 12 [GLS^{+}15]	Minalpher	5*
Ascon	5	Prøst	4
ICEPOLE	6*	RECTANGLE	4
Ketje/Keyak	5	RECTANGLE^{-1}	4*
PRIMATE	$\in \{6,7\}$*		

Intermezzo – bitslice gate complexity

- Minimal number of AND/OR/XOR/NOT operations
- Largely been done for 4x4 S-boxes [UDCI+11]
- Provably optimal bitsliced implementations using provably minimal nonlinear operations

Intermezzo – work in progress...

S-box	BGC	Mine	Authors
Ascon			5 AND, 11 XOR, 6 NOT
ICEPOLE			
Ketje/Keyak	≤ 15	5 AND, 5 XOR, 5 NOT	5 AND, 5 XOR, 5 NOT
PRIMATE		6 AND, 1 OR, 37 XOR, 3 NOT	
PRIMATE^{-1}			
Joltik	11	4 OR, 4 XOR, 3 NOT	4 NOR, 3 XOR, 1 XNOR
Joltik^{-1}	11	4 OR, 4 XOR, 3 NOT	
LAC	13	2 AND, 2 OR, 6 XOR, 3 NOT	
Minalpher			
Prøst			4 AND, 4 XOR
RECTANGLE	≤ 12	2 AND, 2 OR, 7 XOR, 1 NOT	1 AND, 3 OR, 7 XOR, 1 NOT
RECTANGLE^{-1}			

Disclaimer: not optimal in number of NOT

Intermezzo – Joltik

1. $y_0 = x_0 | x_1$
2. $t_0 = \neg x_3$
3. $y_0 = y_0 \oplus t_0$
4. $t_0 = x_1 | x_2$
5. $t_0 = \neg t_0$
6. $y_1 = x_0 \oplus t_0$
7. $t_0 = y_0 | y_1$
8. $t_0 = \neg t_0$
9. $y_3 = t_0 \oplus x_2$
10. $t_0 = x_2 | y_0$
11. $y_2 = t_0 \oplus x_1$

Intermezzo – Joltik^{-1}

1. $y_2 = x_0 | x_1$
2. $t_0 = \neg x_3$
3. $y_2 = y_2 \oplus t_0$
4. $t_0 = x_0 | y_2$
5. $y_1 = t_0 \oplus x_2$
6. $t_0 = y_1 | y_2$
7. $t_0 = \neg t_0$
8. $y_0 = t_0 \oplus x_1$
9. $t_0 = y_0 | y_1$
10. $t_0 = \neg t_0$
11. $y_3 = t_0 \oplus x_0$

Intermezzo – LAC

1. $t_0 = \neg x_1$
2. $t_1 = t_0 | x_0$
3. $t_1 = x_2 \oplus t_1$
4. $t_2 = x_0 \oplus x_3$
5. $t_3 = \neg t_2$
6. $t_2 = t_3 | t_1$
7. $y_3 = t_3 \oplus t_1$
8. $y_0 = x_0 \oplus t_2$
9. $t_2 = t_0 \& y_3$
10. $y_2 = t_1 \oplus t_2$
11. $t_2 = \neg x_0$
12. $t_2 = t_2 \& y_2$
13. $y_1 = x_1 \oplus t_2$

Optimising masking costs

Comparing designs

High-level operations

- Table lookups
- Bitwise Boolean functions
- Shifts and rotates
- Modular addition/multiplication
- Modular polynomial multiplication

Results

Operation	Table lookups	Bitwise Boolean	Shifts/ rotates	Mod. add. and mult.	Mod. poly. mult.
AES	256 bytes	XOR	Fixed		✓
AES tables	4096 bytes	XOR	Fixed		
AES bitsliced		AND,OR,XOR	Fixed		✓
iSCREAM	512 bytes	AND,OR,XOR	Fixed	$\times$ mod 256	
SCREAM	512 bytes	AND,OR,XOR		$\times$ mod 256	
Ascon		AND,OR,XOR	Fixed		
ICEPOLE	96 bytes	AND,XOR	Fixed		
Ketje/Keyak		AND,XOR	Fixed		
PRIMATE	25 bytes	XOR	Fixed		✓
Joltik	64 bytes	XOR	Fixed	$+$ mod 16	✓
LAC	16 bytes	XOR	Fixed		
Minalpher	16 bytes	XOR			
Prøst		AND,XOR	Fixed		
RECTANGLE		AND,OR,XOR	Fixed		

Results

- Expected masking costs less high than in [Mes01]
- Ascon, Ketje, Keyak, LAC, Minalpher, Prøst, and RECTANGLE stand out (at the moment)
- Designers/implementers should use operations that are cheap to mask under a Boolean scheme

Conclusions

- For 4- and 5-bit S-boxes, we can find an implementation with a provably minimum number of AND/OR operations

Conclusions

- For 4- and 5-bit S-boxes, we can find an implementation with a provably minimum number of `AND`/`OR` operations
- Same technique can be used to find provably minimal bitsliced implementations

Conclusions

- For 4- and 5-bit S-boxes, we can find an implementation with a provably minimum number of `AND`/`OR` operations
- Same technique can be used to find provably minimal bitsliced implementations
- Designers and implementers should take masking costs into consideration

Conclusions

- For 4- and 5-bit S-boxes, we can find an implementation with a provably minimum number of `AND`/`OR` operations
- Same technique can be used to find provably minimal bitsliced implementations
- Designers and implementers should take masking costs into consideration
    - CAESAR committee as well
    - Benchmarking possibilities?

Questions

Thank you for your attention

Questions?

References I

Joan Boyar and René Peralta.
A new combinational logic minimization technique with applications to cryptology.
In Paola Festa, editor, *Experimental Algorithms*, volume 6049 of *Lecture Notes in Computer Science*, pages 178–189. Springer Berlin Heidelberg, 2010.

Nicolas Courtois, Daniel Hulme, and Theodosis Mourouzis.
Solving circuit optimisation problems in cryptography and cryptanalysis, 2011.
http://eprint.iacr.org/.

Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, Kerem Varıcı, Anthony Journault, François Durvaux, Lubos Gaspar, and Stéphanie Kerckhof.
SCREAM. CAESAR submissions, 2015.
http://competitions.cr.yp.to/round2/screamv3.pdf.

Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, and Kerem Varıcı.
LS-designs: Bitslice encryption for efficient masked software implementations.
In *Fast Software Encryption – FSE 2014*, 2014.

References II

Thomas S. Messerges.
Securing the AES finalists against power analysis attacks.
In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, and Bruce Schneier, editors, *Fast Software Encryption*, volume 1978 of *Lecture Notes in Computer Science*, pages 150–164. Springer Berlin Heidelberg, 2001.

Theodosis Mourouzis.
Optimizations in Algebraic and Differential Cryptanalysis.
PhD thesis, UCL (University College London), 2015.

Markus Ullrich, Christophe De Canniere, Sebastian Indesteege, Özgül Küçük, Nicky Mouha, and Bart Preneel.
Finding optimal bitsliced implementations of 4x4-bit S-boxes.
In *SKEW 2011 Symmetric Key Encryption Workshop, Copenhagen, Denmark*, pages 16–17, 2011.