Linear (Hull) and Algebraic Cryptanalysis of the Block Cipher PRESENT

Jorge Nakahara Jr[1], Pouyan Sepehrdad[1], Bingsheng Zhang[2], Meiqin Wang[3]

[1]EPFL, Lausanne, Switzerland
[2]Cybernetica AS, Estonia and University of Tartu, Estonia
[3]Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, China

Estonian Theory Days, 2-4.10.2009

Why cryptanalysis?!

Outline

- Contributions
- The PRESENT Block Cipher
- Revisited Algebraic Cryptanalysis of PRESENT
- Linear Cryptanalysis of PRESENT
- Linear Hulls of PRESENT
- Conclusions
- Acknowledgements

Outline

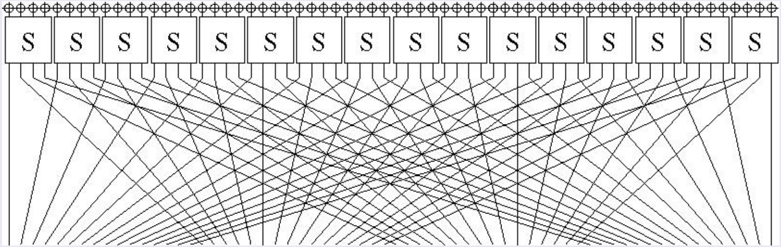
Contributions

- we performed linear analysis of reduced-round PRESENT
  - exploiting fixed-points (and other symmetries) of pLayer
  - exploiting low Hamming Weight bitmasks
  - using iterative linear relations
  - first linear hull analysis of PRESENT: 1st and 2nd best trails
  - known-plaintext and ciphertext-only attack settings
- revisited algebraic analysis of 5-round PRESENT in less than 3 min
- best attacks on up to 26-round PRESENT (out of 31 rounds)

The PRESENT Block Cipher

- block cipher designed by Bogdanov *et al.* at CHES'07
- aimed at RFID tags, sensor networks (hardware environments)
- SPN structure
- 64-bit block size, 80- or 128-bit key size, 31 rounds
- **one full round**: xor with round subkey, S-box layer, bit permutation (pLayer)
- **key schedule**: 61-bit left rotation, S-box application, and xor with counter

Computational graph of one full round of PRESENT



Previous attack complexities on reduced-round PRESENT

#Rounds	Time	Data	Memory	Key Size	Comments
5	1.82 h	64 KP	—	80	KR, AC
7	$2^{100.1}$	$2^{24.3}$ CP	2^{77}	128	IC
15	$2^{35.6}$	$2^{35.6}$ CP	2^{32}	all	KR, SSC
16	2^{62}	2^{62} CP	1Gb	all	KR, AC + DC
16	2^{64}	2^{64} CP	2^{32}	all	KR, DC
17	2^{104}	2^{63} CP	2^{53}	128	KR, RKR
17	2^{93}	2^{62} CP	1Gb	128	KR, AC + DC
18	2^{98}	2^{62} CP	1Gb	128	KR, AC + DC
19	2^{113}	2^{62} CP	1Gb	128	KR, AC + DC
24	2^{57}	2^{57} CP	2^{32}	all	KR, SSC

KR: Key Recovery attack; LC: Linear Cryptanalysis; AC: Algebraic Crypt;
DC: Differential Cryptanalysis; RKR: Related-Key Rectangle;
SSC: Statistical Saturation analysis; IC: Integral Cryptanalysis;
CP: Chosen Plaintext; KP: Known Plaintext;

Outline

Our attacks on reduced-round PRESENT

#Rounds	Time	Data	Memory	Key Size	Comments
5	2.5 min	5 KP	—	80	KR†, AC
5	2.5 min	5 KP	—	128	KR†, AC
14	2^{61}	2^{61} CO	—	all	DR* + KR, LC
17	$2^{69.50}$	2^{64} KP	2^{12}	80	KR, LC
17	$2^{73.91}$	2^{64} KP	2^{16}	80	KR, LC
24	$2^{63.42}$	2^{64} KP	2^{8}	all	KR, LH
25	$2^{98.68}$	2^{64} KP	2^{40}	128	KR, LH
26	$2^{98.62}$	2^{64} KP	2^{40}	128	KR, LH

*: time complexity is number of parity computations; †: recover half of the user key;
KR: Key Recovery attack; LC: Linear Cryptanalysis; AC: Algebraic Crypt;
DC: Differential Cryptanalysis; LH: Linear Hull;
CP: Chosen Plaintext; KP: Known Plaintext; CO: Ciphertext Only

Outline

Revisited Algebraic Cryptanalysis of PRESENT

Algebraic Cryptanalysis

- Attributed to C. Shannon: breaking a good cipher should require “as much work as solving a system of simultaneous equations in a large number of unknowns of a complex type”.
- Small number of plaintext / ciphertext pairs.
- Realistic compare to traditional statistical attacks like linear and differential cryptanalysis.
- Not simple to predict the complexity of the attack priori to provoking the experiments.

Revisited Algebraic Cryptanalysis of PRESENT

Distinct Stages

- Writing the cipher as a polynomial system of equations of low degree (often over $\mathbb{GF}(2)$ or $\mathbb{GF}(2^k)$).
- Solve the corresponding system of equations.
  - Gröbner basis (Buchberger, F4 or F5), ElimLin, XL and its family, Raddum-Semaev algorithm or SAT solvers.

Focus: ElimLin and Gröbner basis algorithms (F4 under PolyBori framework)

Revisited Algebraic Cryptanalysis of PRESENT

ElimLin Algorithm

- Proposed by N. Courtois to attack DES (breaks 5-round DES)
  - Gaussian Elimination: all the linear equations in the span of initial equations are found.
  - Substitution: one of the variables nominated in each linear equation and is substituted in the whole system.
  - Iteration: repeat up to the time no new linear equation is found.

ElimLin recovers half of the bits of the key of 5-round PRESENT in less than 3 mins using a 2Ghz CPU with 1GB RAM. (both key sizes) [First attack by N. Courtois against PRESENT-80.]

Revisited Algebraic Cryptanalysis of PRESENT

Our Goal: comparison between ElimLin and F4 algorithm under PolyBori framework.

PolyBori

- The most efficient implementation of F4 known to us.
- A C++ library designed to compute Gröbner basis applied to Boolean polynomials.
- A Python interface surrounding the C++ core.
- Zero-suppressed binary decision diagrams (ZDD) as a high level data structure to store Boolean polynomials.
  - Less memory and speculated to be faster!

We used polybori-0.4 in our attacks.

Algebraic expression of the S-box

$$\begin{cases} y_0 = x_1x_2 + x_0 + x_2 + x_3 \\ y_1 = x_0x_1x_3 + x_0x_2x_3 + x_0x_1x_2 + x_1x_3 + x_2x_3 + x_1 + x_3 \\ y_2 = x_0x_1x_3 + x_0x_2x_3 + x_0x_1 + x_0x_3 + x_1x_3 + x_2 + x_3 + 1 \\ y_3 = x_0x_1x_2 + x_0x_1x_3 + x_0x_2x_3 + x_1x_2 + x_0 + x_1 + x_3 + 1 \end{cases}$$

$$\begin{cases} x_0 = y_1y_3 + y_0 + y_2 + 1 \\ x_1 = y_0y_1y_2 + y_0y_1y_3 + y_0y_2y_3 + y_0y_2 + y_1y_3 + y_2y_3 + y_0 + y_1 + y_3 \\ x_2 = y_0y_1y_2 + y_0y_1y_3 + y_0y_2y_3 + y_0y_1 + y_0y_2 + y_1y_2 + y_0y_3 + y_1y_3 \\ \quad + y_3 + 1 \\ x_3 = y_0y_1y_2 + y_0y_2y_3 + y_0y_1 + y_0 + y_1 + y_2 + y_3 \end{cases}$$

Revisited Algebraic Cryptanalysis of PRESENT

Comparison

# R	#key bits	#key bits fixed	full key (hours)	# plaintexts	notes
5	80	40	0.04	5 KP	ElimLin
5	80	40	0.07	5 KP	PolyBori
5	80	37	0.61	10 KP	ElimLin
5	80	37	0.52	10 KP	PolyBori
5	80	36	3.53	16 KP	ElimLin
5	80	36	Crashed!	16 KP	PolyBori
5	80	35	4.47	16 KP	ElimLin
5	80	35	Crashed!	16 KP	PolyBori
5	128	88	0.05	5 KP	ElimLin
5	128	88	0.07	5 KP	PolyBori

Revisited Algebraic Cryptanalysis of PRESENT

Results specifically for PRESENT

- nothing better to use PolyBori compared to ElimLin.
- still PolyBori uses a large amount of memory (Gröbner basis approach of increasing the polynomial degree in intermediate stages), where the one of ElimLin is negligible.
- time complexity is closely comparable to ElimLin.
- faced multiple instances in which PolyBori crashed! (probably due to running out of memory) but ElimLin outputs the result.

Why not to use a simple algorithm like ElimLin, comparing to a complex one such as Gröbner basis approach!

Linear Cryptanalysis of PRESENT

- ideas related to LC dates back to A. Shamir (1985) observations on DES S-boxes
- LC technique was developed by M. Matsui against FEAL (1992)
- later Matsui applied LC to DES (1993, 1994)
- known-plaintext and ciphertext-only (ASCII plaintext) attack settings
- main concept: linear relation

$$P \cdot \Gamma_P \oplus C \cdot \Gamma_c = K \cdot \Gamma_k$$

  with probability p
- bias: $\epsilon = |p - \frac{1}{2}|$, with $0 \leq \epsilon \leq \frac{1}{2}$

Linear Cryptanalysis of PRESENT

Analysis Strategy for PRESENT

- exploit poor diffusion in pLayer (branch number: 2)
- exploit single-bit trails (low HW bitmasks)
- look for iterative linear relations
- minimize number of active S-boxes per round
- study linear hull effect (multiple trails sharing the same fixed plaintext/ciphertext masks)

Linear Cryptanalysis of PRESENT

Example of linear relation with high bias

$$8000000000000000_x \xrightarrow{1r} 8000000000000000_x$$

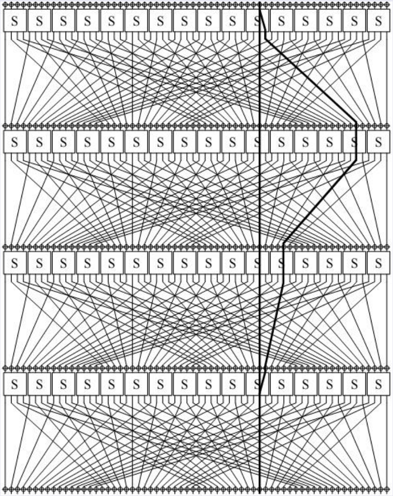
- translates into $p_0 \oplus c_0 = k_0^1 \oplus k_0^2$ for one round
- 1-round **iterative** linear relation
- bias: 2^{-3} (one active S-box)
- based on **fix-point** of pLayer
- ASCII plaintext: ciphertext-only (CO) distinguish-from-random attack
- on 14 rounds: 2^{61} CO and equivalent parity computations, negligible memory

Linear Hulls of PRESENT

- concept due to K. Nyberg (1994)
- linear hull is the collection of all linear relations sharing the same plaintext and ciphertext masks (across a certain number of rounds of a cipher)
- bias of linear hull: $\epsilon^2 = \sum_{i=1}^{t} \epsilon_i^2$, where ϵ_i^2 are the bias of individual linear trails
- the **linear hull effect** accounts for a clustering of linear trails, ie. several distinct paths across the linear distinguisher
- in PRESENT, this effect was observed in practice (even for a small number of rounds)
- linear hulls $\neq$ multiple linear relations

Linear Hulls of PRESENT

Example of branching and merging back to one S-box



Linear Hulls of PRESENT

Software Tools

- We have developed software tools to search for linear trails inside a hull
- recursive depth-first search with optimization strategies:
  - **minimize** number of active S-boxes
  - **single-bit trails**
  - 1st and 2nd best trails: r or $r+2$ active S-boxes across r rounds
  - **upper bound** the bias of individual trails
  - trails are probably not 'linearly independent' (not a problem c.f. Kaliski-Robshaw)

Linear Hulls of PRESENT

ALH vs. Piling-up lemma

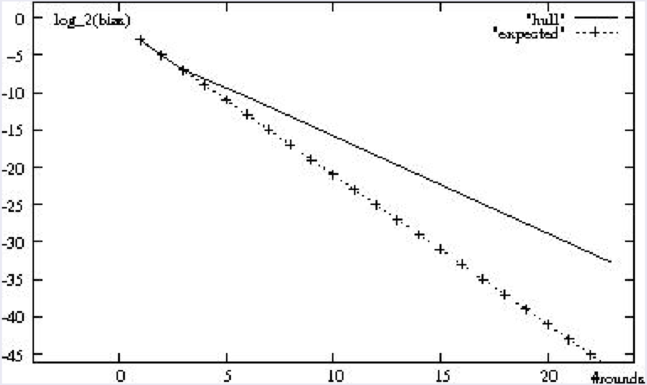
Table: Computed bias (cb) and expected bias (eb) of linear hulls in PRESENT for input/output mask 0000000000200000_x

# rounds	1	2	3	4	5	6	7
# trails	1	1	1	9	9	27	72
(cb)	2^{-3}	2^{-5}	2^{-7}	$2^{-8.20}$	$2^{-9.40}$	$2^{-10.61}$	$2^{-11.90}$
(eb)	2^{-3}	2^{-5}	2^{-7}	2^{-9}	2^{-11}	2^{-13}	2^{-15}
# rounds	8	9	10	11	12	13	14
# trails	192	512	1344	3528	9261	24255	63525
(cb)	$2^{-13.19}$	$2^{-14.48}$	$2^{-15.78}$	$2^{-17.08}$	$2^{-18.38}$	$2^{-19.71}$	$2^{-21.02}$
(eb)	2^{-17}	2^{-19}	2^{-21}	2^{-23}	2^{-25}	2^{-27}	2^{-29}
# rounds	15	16	17	18	19	20	21
# trails	166375	435600	1140480	2985984	7817472	20466576	53582633
(cb)	$2^{-22.33}$	$2^{-23.63}$	$2^{-24.94}$	$2^{-26.25}$	$2^{-27.55}$	$2^{-28.85}$	$2^{-30.16}$
(eb)	2^{-31}	2^{-33}	2^{-35}	2^{-37}	2^{-39}	2^{-41}	2^{-43}
# rounds	22	23					
# trails	140281323	367261713					
(cb)	$2^{-31.47}$	$2^{-32.77}$					
(eb)	2^{-45}	2^{-47}					

Linear Hulls of PRESENT

ALH vs. Piling-up lemma

This graph shows that the linear hull effect (clustering of linear trails) for r-round PRESENT ($1 \leq r \leq 23$).



Outline

Linear Hulls of PRESENT

Number of second best trails

Table: Input/Output bitmask 0000000000200000_x

# rounds	# best trails	# 2nd best trail	bias of 2nd best trails
5	9	18	$2^{-12.915}$
6	27	81	$2^{-13.830}$
7	72	288	$2^{-14.915}$
8	192	960	$2^{-16.046}$
9	512	3072	$2^{-17.207}$
10	1344	9536	$2^{-18.376}$
11	3528	28896	$2^{-19.565}$
12	9261	85995	$2^{-20.771}$
13	24255	252021	$2^{-21.990}$
14	63525	730235	$2^{-23.219}$

Linear Hulls of PRESENT

21 and 22-round linear hull

- ALH (0000000000200000_x,0000000000200000_x) for 21 rounds is $2^{-60.22}$.
- 25-round key-recovery attack: complexity $2^{98.68}$ 25-round computations, memory 2^{40} and success rate 0.61.
- ALH (0000000000200000_x,0000000000200000_x) for 22 rounds is $2^{-62.83}$.
- 26-round key-recovery attack: complexity $2^{98.62}$ 26-round computations, memory 2^{40} (128 key bit only).

Conclusions

Highlights

- Revisited algebraic analysis of 5-round PRESENT
- Linear analysis with ciphertext-only attacks (14 rounds)
- First linear hull analysis of up to 26-round PRESENT (but small success rate)

Acknowledgements

Thanks

- Bingsheng Zhang is supported by Estonian Science Foundation, grant #8058, the European Regional Development Fund through the Estonian Center of Excellence in Computer Science (EXCS), and the 6th Framework Programme project AEOLUS (FP6-IST-15964).
- Meiqin Wang is supported by 973 Program of China (Grant No.2007CB807902) and National Outstanding Young Scientist fund of China (Grant No. 60525201).