Ciphers for MPC and FHE

Martin Albrecht[1] Christian Rechberger[2] Thomas Schneider[3] Tyge Tiessen[2] Michael Zohner[3]

ICMS, Security of symmetric ciphers in network protocols, Edinburgh

[1]Royal Holloway, University of London, UK

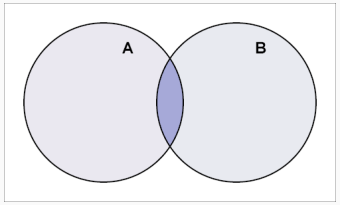
[2]DTU Compute, Technical University of Denmark, Denmark

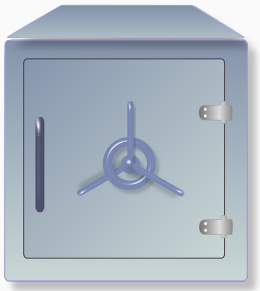
[3]TU Darmstadt, Darmstadt, Germany

Introduction

MPC Motivations

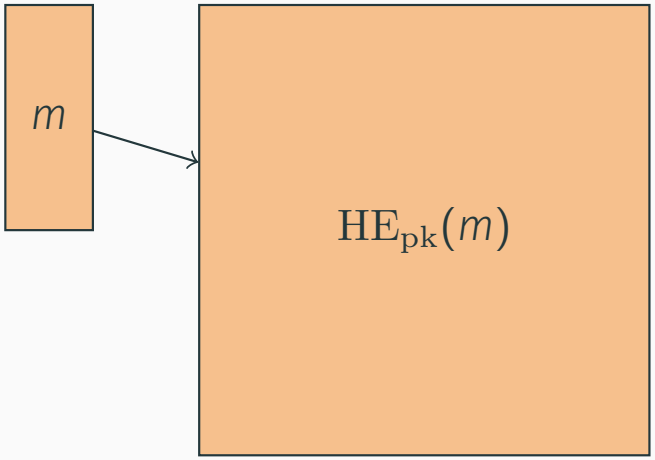
Block ciphers have various applications in MPC



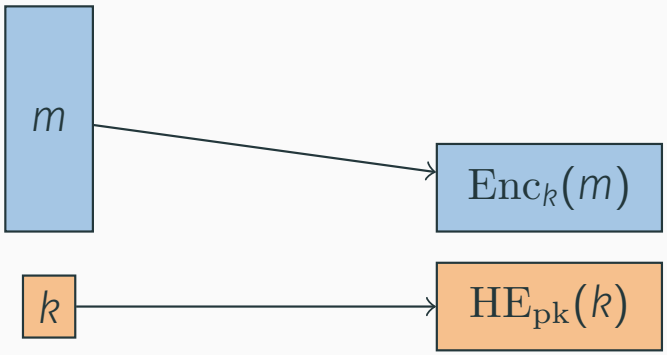


- Server-side one-time passwords, commercialized by Dyadic Security (server-side derivation of one-time passwords via MPC)
- Oblivious Pseudorandom Functions (OPRFs) for privacy-preserving keyword search, private set intersection, secure database join, etc.
- Secure storage: store symmetrically encrypted intermediate MPC values in untrusted storage

FHE Motivation

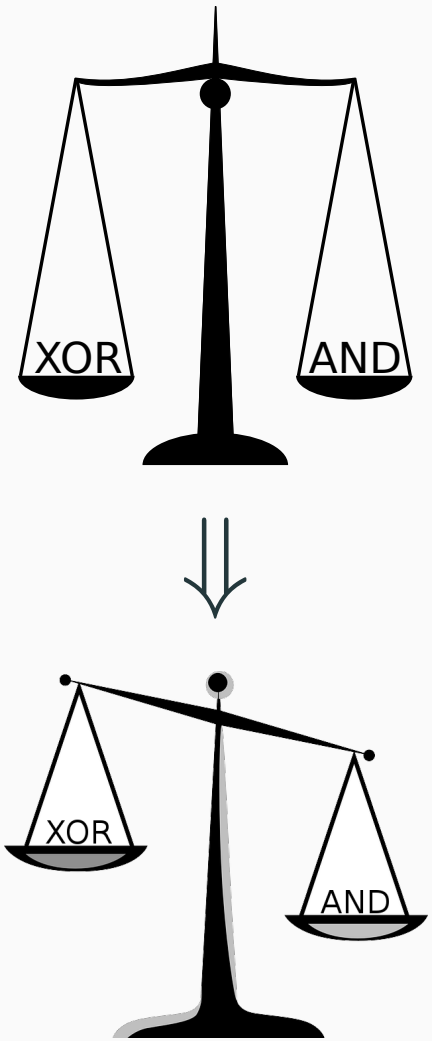


FHE schemes typically come with a ciphertext expansion in the order of 1000s to 1000000s.



Solution:
Encrypt message symmetrically, transfer key homomorphically.
Cloud then decrypts homomorphically.

New computational models require new designs



- Since 1970s: balance between linear and non-linear operations because AND gates and XOR gates are roughly the same in most hardware.
- But cost of XOR gate is (almost) negligible compared to AND gate in MPC or FHE setting
- Idea: Explore extreme trade-offs

Our Guiding Question

What would an efficient cipher look like if linear operations were for free?

Possible metrics for optimisation

There are three possible metrics to minimise:

1. ANDs per bit of encrypted text (ANDs/bit)
2. multiplicative depth of the encryption circuit (ANDdepth)
3. total number of ANDs per encryption (ANDs)

Refined Guiding Question

Can we design a cipher that can be optimized with regard to any combination of these metrics?

Related work

Minimisation of multiplicative complexity also relevant in side-channel countermeasures. However, such designs are much less extreme:

- Noekeon
- Fantomas
- Robin

Joan Daemen, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen. Nessie proposal: Noekeon. In *First Open NESSIE Workshop*, 2000.

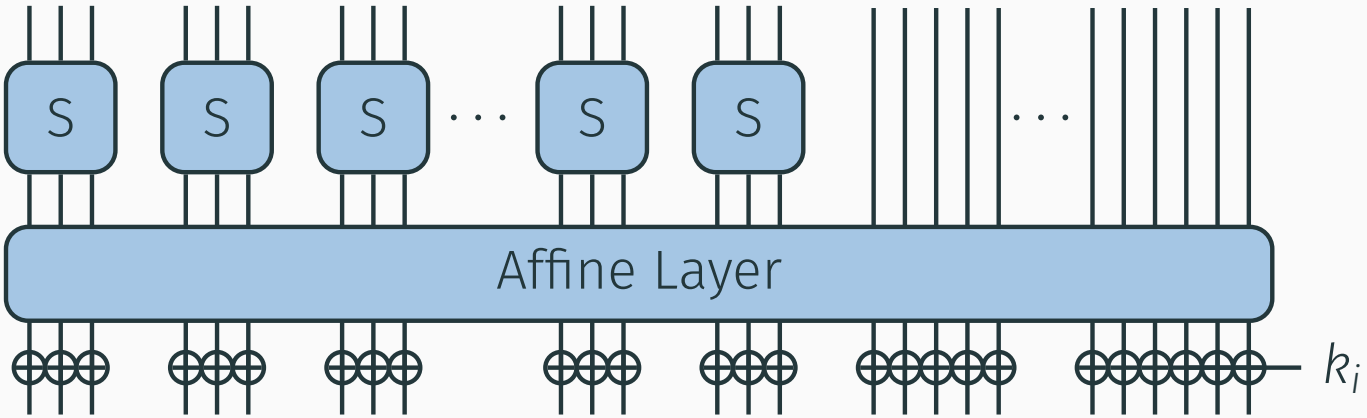
Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, and Kerem Varici. LS-designs: Bitslice encryption for efficient masked software implementations. In *Fast Software Encryption (FSE 2014)*, LNCS. Springer.

Design Ideas

Minimise ANDs needed for confusion, maximise diffusion.

- Use a Substitution-Permutation network (SPN)
- Use small S-boxes with low multiplicative complexity
- Utilise a partial substitution layer
- Maximise diffusion in affine layer

The LowMC round function and parameters



Size parameters

- block size n bits
- number of S-boxes m in substitution layer

Security parameters

- key size k
- permitted data complexity d

Number of rounds r is calculated as a function of the above.

S-box Properties

- maximum differential probability 2^{-2}
- maximum linear probability 2^{-2}
- circuit needs only 3 AND gates and has ANDdepth 1
- any combination of output bits has algebraic degree 2

Algebraic Normal Form of S-box:

$$S_0(A, B, C) = A \oplus BC$$
$$S_1(A, B, C) = A \oplus B \oplus AC$$
$$S_2(A, B, C) = A \oplus B \oplus C \oplus AB$$

Maximise diffusion in affine layer

How do we maximise diffusion in affine layer?

- Choose most general affine layer: multiplication with $n \times n$ matrix over $\mathbb{F}_2$ and addition of constant $\mathbb{F}_2$ vector of length n.

How do we choose good matrices and vectors?

- Unfortunately, determining branch number of a binary matrix is hard in practice and theory.

We thus

- choose random matrices uniformly from all invertible $n \times n$ matrices over $\mathbb{F}_2$.
- choose random constant vectors uniformly from $\mathbb{F}_2^n$.

Bonus: This allows novel security arguments.

Key schedule

Reuse random matrix approach for key schedule:

- Derive round keys from general key by multiplication with $n \times k$ binary matrix.
- Choose matrices uniformly at random from all binary $n \times k$ matrices of rank $\min(n, k)$.

Instantiation of affine layers and round key matrices

Problem: How do you accountably instantiate the random matrices and vectors?

- instance of cipher cannot use ”random” matrices but must use fixed ones
- how choose them in an accountable way (”nothing up the sleeve”)?

Our solution:

- Use Grain LFSR as self-shrinking generator to produce random bit string
- Then use this string to generate the matrices.

“Free XOR”

Ain't no such thing as free XOR

- We started our work assuming that XORs are "essentially free".
- Turns out, "essentially" is not "actually".
- When doing $\approx n^2$ XORs per round this starts to hurt, both in the FHE and the MPC case (in particular in the LAN setting)
- We hence use techniques from efficiently linear algebra over $\mathbb{F}_2$ to reduce the cost of matrix-vector products.

Gray Codes

The Gray code, named after Frank Gray and also known as reflected binary code, is a numbering system where two consecutive values differ in only one digit.

	0	0	⇓
0	0	1	
1	1	1	
	1	0	⇑

0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

Consider $w = A \cdot v$ (A is $\in \mathbb{F}_2^{n \times n}$ and v is $\in \mathbb{F}_2^n$), where operations on v are expensive.

Divide A into n/k vertical "stripes" $A_1 \ldots_{n/k}$ of k columns each. Split v into n/k horizontal "stripes" $v_1 \ldots v_{n/k}$ of k rows each. We have:

$$C = A \cdot v = \sum_{1}^{n/k} A_i \cdot v_i.$$

$$A = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}, v = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

$$A_0 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, A_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}, v_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, v_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$A_0 \cdot v_0 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, A_1 \cdot v_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

M4RM: Algorithm $O(n^2 / \log n)$

```
1 begin
2     w ⟵ all zero vector of length n;
3     k ⟵ ⌊log n⌋;
4     for 0 ≤ i < (n/k) do
          // create table of 2^k − 1 linear combinations
5         T ← MakeTable(v, i × k, 0, k);
6         for 0 ≤ j < n do
              // read index for table T
7             id ⟵ ReadBits(A, j, i × k, k);
8             add row id from T to row j of w;
9     return w;
```

Algorithm 1: M4RM

Security Analysis

To determine round number cryptanalysis necessary

Two factors determine the number of rounds

1. Maximal length of a distinguisher
2. Number of rounds that can be peeled off

We investigated the following distinguishers:

- Statistical distinguishers: linear and differential characteristics
- Low-degree attacks
- Combined attacks, special case: Boomerang attacks

Resistance Against Differential attacks

Standard method to determine probability of best differential characteristic:

1. Determine minimal number of active S-boxes.
2. Combine with maximal differential probability of S-box to determine lower bound on best possible characteristic.

To determine the minimal number of active S-boxes the branch number would be helpful.

Problem

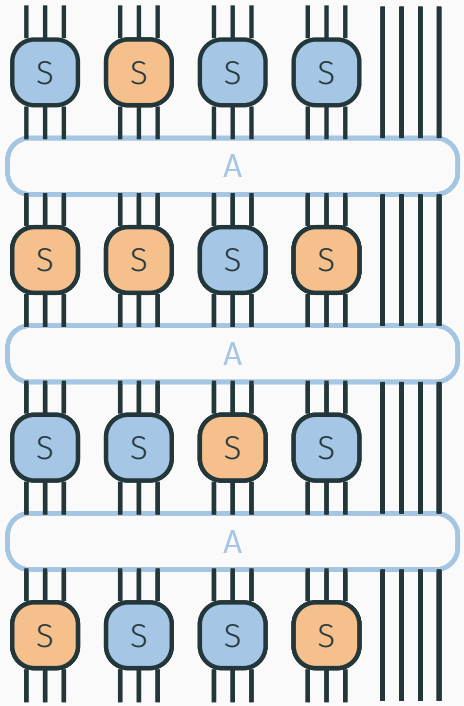
We do not know the branch number of the randomly chosen matrix.

Bounding differential characteristics

Idea

Calculate for each possible good differential characteristic probability that it is realised in instantiation of LowMC. Sum all these probabilities to get upper bound for probability that at least one is realised.

Let C be the set of possible good characteristics.



$$\sum_{c \in C} \Pr(c \text{ exists in cipher}) \geq \Pr(\text{good characteristic exists})$$

Bounding Differential Characteristics (Details)

m	number of S-boxes per layer
l	bit-length of the identity part of layer
n	$= 3m + l$

Let $V(i)$ be the number of n-bit vectors encoding a difference activating i S-boxes.

- We choose i out of the m S-boxes,
- for each active S-box there are 7 possible non-zero input differences and
- the bits of the identity part can be chosen freely. So

$$V(i) = \binom{m}{i} \cdot 7^i \cdot 2^l.$$

Bounding Differential Characteristics (Details)

- Let (α_0, α_1) be input/output difference pair for one round.
- Let a_0 be the number of S-boxes activated by α_0.
- An active S-box maps its non-zero input difference to four possible output differences each with probability $\frac{1}{4}$.
- A random binary $n \times n$ matrix maps a given non-zero n-bit vector with probability $\frac{1}{2^n-1}$ to another given non-zero output vector.

The probability that the one-round characteristic (α_0, α_1) has a probability larger than 0 is

$$\frac{4^{a_0}}{2^n - 1}.$$

Bounding Differential Characteristics (Details)

- Let $A = (\alpha_0, \alpha_1, \ldots, \alpha_r)$ be a characteristic over r rounds.
- Let $(a_0, a_1, \ldots, a_{r-1})$ be the numbers of S-boxes activated by α_0, α_1,..., and α_{r-1}.

Calculate the probability that A has a probability larger than 0 in a random instantiation of LowMC as

$$\frac{4^{a_0}}{2^n - 1} \cdot \frac{4^{a_1}}{2^n - 1} \cdots \frac{4^{a_{r-1}}}{2^n - 1} = \frac{4^{a_0 + a_1 + \cdots + a_{r-1}}}{(2^n - 1)^r}.$$

Bounding Differential Characteristics (Details)

Summing over all possible r-round characteristics that activate at most d S-boxes, calculate an upper bound for the probability that there exists an r-round characteristic with d or fewer active S-boxes as

$$\sum_{\substack{0 \leq a_0, a_1, \ldots, a_{r-1} \leq m \\ a_0 + a_1 + \cdots + a_{r-1} \leq d}} V(a_0) \cdot V(a_1) \cdots V(a_{r-1}) \cdot (2^n - 1) \cdot \frac{4^{a_0 + a_1 + \cdots + a_{r-1}}}{(2^n - 1)^r}$$

where the factor $(2^n - 1)$ is the number of choices for the last difference α_r that can take any non-zero value.

Bounding Differential Characteristics (Details)

- Each active S-box reduces the probability of a characteristic by a factor of 2^{-2}.
- From this, calculate the number of rounds after which no good differentials are present except for a negligible probability.
- We consider as good differential characteristics those with a probability higher than 2^{-d}, where d is the allowed data complexity in the respective parameter set.
- We call a negligible probability a probability lower than 2^{-100}.
- Note that this probability only comes into play once when fixing an instantiation of LowMC.

Question: What is the minimal number of rounds needed to reach a given algebraic degree?

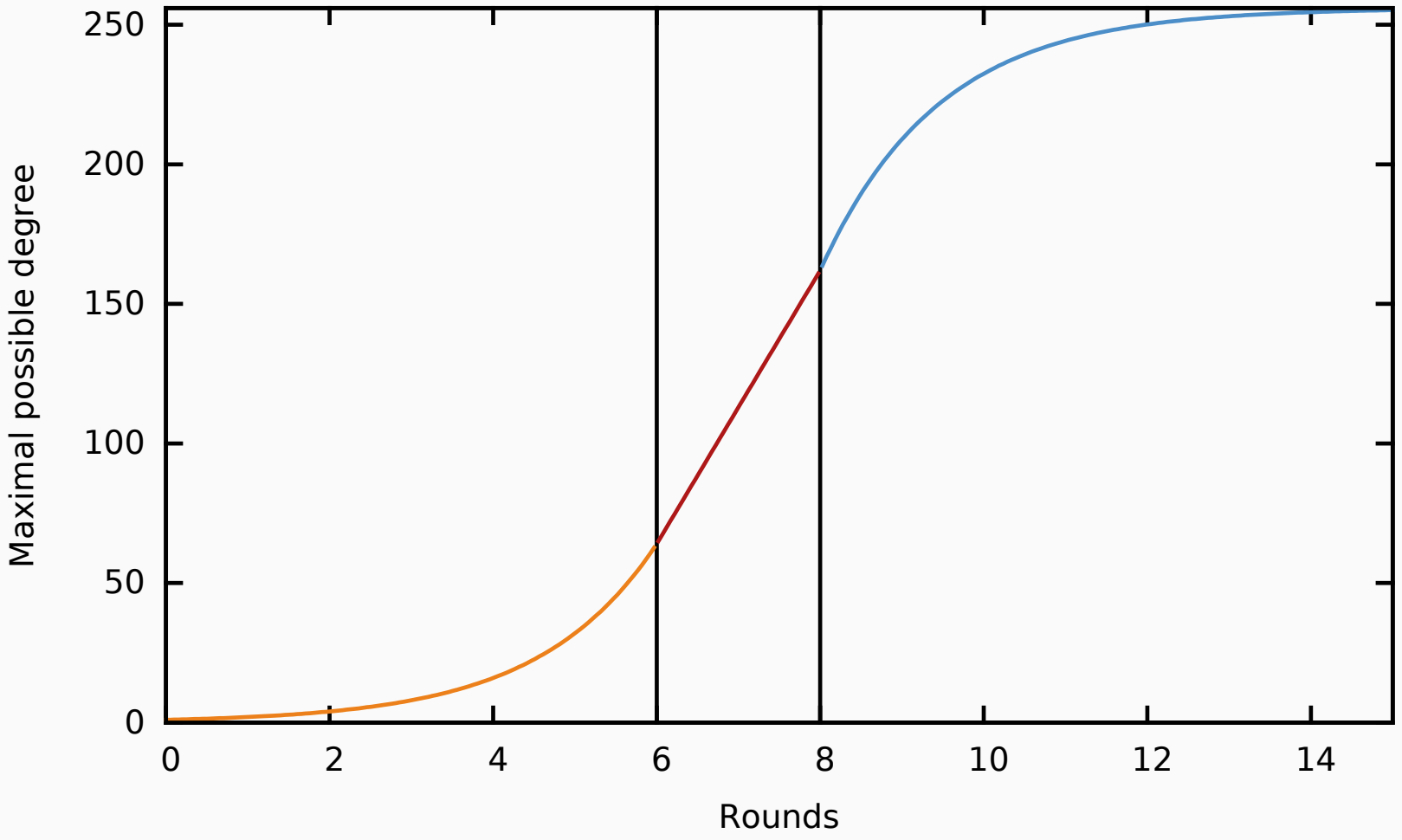
Lemma

If algebraic degree is d_r after r rounds, max. degree in round $r+1$ is

$$\min\left(2d_r, m + d_r, \frac{n}{2} + \frac{d_r}{2}\right).$$

- The first bound is trivial.
- The second bound is new.
- Third bound was proven by Boura, Canteaut, and De Cannière [BCC11]

Degree Growth



Formula to calculate number of rounds

Round formula

$$r \geq \max(r_{\text{stat}}, r_{\text{deg}}, r_{\text{cmbnd}}) + r_{\text{outer}}$$

- r_{stat}: bound for differential and linear distinguishers
- r_{deg}: bound for sufficient degree
- r_{cmbnd}: bound for combined distinguishers
- r_{outer}: bound for rounds that can be peeled off

For r_{outer}, we use the ad-hoc formular

$$r_{\text{outer}} = r_{\text{stat}}.$$

We thank Dmitry Khovratovich for pointing out that combined attacks can be more effective than others.

Parameter Sets

Parameters

S-boxes	blocksize	data	r_{stat}	r_{bmrg}	r_{deg}	total rounds
49	256	2^{64}	5	6	6	11
63	256	2^{128}	5	6	7	12

- But LowMC is not limited to this parameter set
- Dependent on optimization metric, size parameters and security parameters other parameter sets can be calculated
- As little as 9 rounds possible for data security of 128 bits

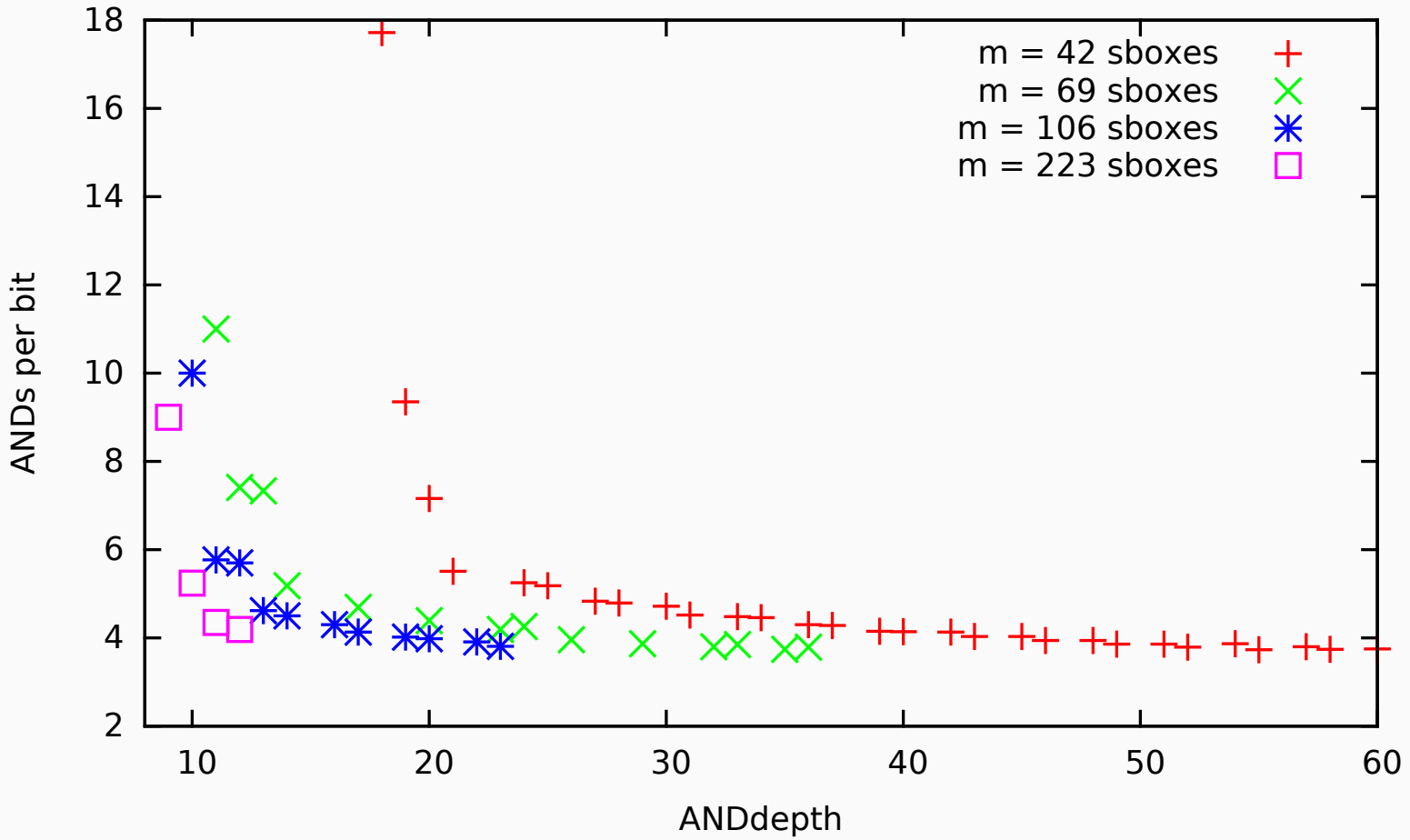
Improved Interpolation Attacks

Instance	Rounds	Instances[1]	Data	Time	Memory
LowMC-80	9/11	1/1	2^{35}	2^{38}	2^{35}
	10/11	1/1	2^{39}	2^{57}	2^{39}
	11/11	2^{-38}	2^{39}	2^{57}	2^{39}
LowMC-128	11/12	1	2^{70}	2^{86}	2^{70}
	12/12	2^{-122}	2^{70}	2^{86}	2^{70}
	12/12	1	2^{73}	2^{118}	2^{80}

Itai Dinur, Yunwen Liu, Willi Meier, and Qingju Wang
Optimized Interpolation Attacks on LowMC
Cryptology ePrint Archive, Report 2015/418. 2015

[1] Over randomness of matrix creation.

Parameter space for AES-like security



Comparison with most competitive other ciphers

AES-like security

Cipher	Key size	Block size	Data sec.	ANDdepth	ANDs/bit
AES-128	128	128	128	40 (60)	43 (40)
Simon	128	128	128	68	34
Noekeon	128	128	128	32	16
Robin	128	128	128	96	24
Fantomas	128	128	128	48	16.5
LowMC	128	256	128	12	8.85

Comparison with most competitive other ciphers

Lightweight security

Cipher	Key size	Block size	Data sec.	ANDdepth	ANDs/bit
PrintCipher-96	160	96	96	96	96
PrintCipher-48	80	48	48	48	48
Present	80 or 128	64	64	62 (93)	62 (31)
Simon	96	64	64	42	21
Simon	64	32	32	32	16
Prince	128	64	64	24	30
KATAN64	80	64	64	74	36
KATAN32	80	32	32	64	24
DES	56	64	56	261	284
LowMC	80	256	64	11	6.31

Benchmark results

Lightweight Security

Cipher	Present		Simon		LowMC	
Comm. [GB]	7.4		5.0		**2.5**	
	LAN	WAN	LAN	WAN	LAN	WAN
Total [s]	216.88	488.24	272.22	605.41	**45.36**	**155.75**

Long-Term Security

Cipher	AES		Simon		LowMC	
Comm. [GB]	16		13		**3.5**	
	LAN	WAN	LAN	WAN	LAN	WAN
Total [s]	555.91	947.79	447.27	761.90	**64.37**	**215.01**

Benchmark results FHE using HElib by Halevi & Shoup

d	n	ANDdepth	t_{block}	t_{bit}	Cipher	Ref.	Key Sched.
128	128	40	1.5s	0.0119s	AES-128	[GHS12]	excluded
128	128	40	55s	0.2580s	AES-128	[DHS14]	excluded
128	128	40	22m	10.313s	AES-128	[MS13]	excluded
128	128	40	14m	6.562s	AES-128	[MS13]	excluded
128	256	12	0.8s	0.0033s	LowMC	this work	included
64	size	24	3.3s	0.0520s	PRINCE	[DSES14]	excluded
64	256	11	0.64s	0.0025s	LowMC	this work	included

Conclusion

Conclusion

- Proposed flexible block cipher design of extremely low number of ANDs/bit and extremely low ANDdepth
- Provided experimental and theoretical cryptanalysis to ensure soundness of design
- Demonstrate that symmetric design and cryptanalysis can significantly contribute to make applications of MPC and FHE more practical
- Measured speed-up factors between 2 and 25

Open problems

- Can the cost of LowMC in the traditional setting be reduced by using a sparser affine layer without reducing security claims?
- Improve implementations of LowMC in MPC and FHE settings
- What designs can minimize the multiplicative complexity over larger fields than $\mathbb{F}_2$?
- Further refinement of round number formula, explicitly include key size
- Further cryptanalysis needed

Thank you

Questions?

paper http://thomaschneider.de/papers/ARSTZ15.pdf

fhe impl. https://bitbucket.org/malb/lowmc-helib

mpc impl. https://github.com/encryptogroup/aby

References


Christina Boura, Anne Canteaut, and Christophe De Cannière.
Higher-order differential properties of Keccak and Luffa.
In *Fast Software Encryption (FSE)*, volume 6733 of *LNCS*, pages 252–269. Springer, 2011.


Yarkın Doröz, Yin Hu, and Berk Sunar.
Homomorphic AES evaluation using NTRU.
Cryptology ePrint Archive, Report 2014/039, 2014.
`http://eprint.iacr.org/2014/039`.


Yarkın Doröz, Aria Shahverdi, Thomas Eisenbarth, and Berk Sunar.
Toward practical homomorphic evaluation of block ciphers using Prince.
Cryptology ePrint Archive, Report 2014/233, 2014.
`http://eprint.iacr.org/2014/233`, presented at Workshop on Applied Homomorphic Cryptography and Encrypted Computing (WAHC'14).