Algebraic Meet-in-the-Middle Attack on LowMC

Fukang Liu[1], Santanu Sarkar[4], Gaoli Wang[5], Willi Meier[6], Takanori Isobe[1,2,3]

[1]University of Hyogo, Japan
[2]NICT, Japan,
[3]PRESTO, Japan
[4]Indian Institute of Technology Madras, India
[5]East China Normal University, China
[6]FHNW, Switzerland

liufukangs@gmail.com

Asiacrypt 2022, December 6

The LowMC Primitive

- Proposed at Eurocrypt 2015
- Designed to be MPC/FHE/ZK-friendly
- Flexible parameters (affine layers, KSF, #S-boxes per round)

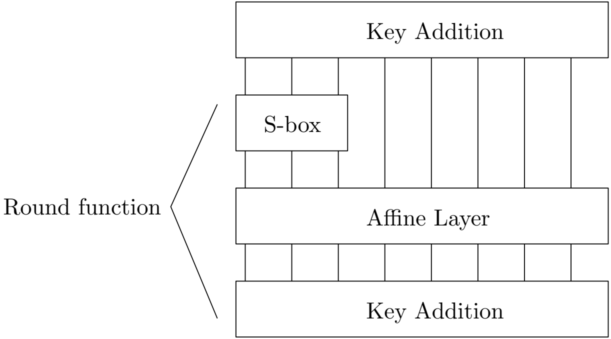


Figure: The round function of LowMC

Previous Results of LowMC

- > 3 chosen plaintext-ciphertext pairs
  - Higher-order differential attack (ICISC 2015)
  - Interpolation attack (Asiacrypt 2015)
- $= 3$ chosen plaintext-ciphertext pairs
  - Difference enumeration attack (ToSC 2018)
- $= 2$ chosen plaintext-ciphertext pairs (Security proof of Picnic)
  - Difference enumeration + algebraic method (CRYPTO 2021)
- $= 1$ known plaintext-ciphertext pair (Security of Picnic)
  - Guess-and-determine (GnD) attack (ToSC 2020, Asiacrypt 2021)
  - Polynomial method (EUROCRYPT 2021)
  - Polynomial method + GnD (ToSC 2022)

Previous Results of LowMC

- > 3 chosen plaintext-ciphertext pairs
  - Higher-order differential attack (ICISC 2015)
  - Interpolation attack (Asiacrypt 2015)
- $= 3$ chosen plaintext-ciphertext pairs
  - Difference enumeration attack (ToSC 2018)
- $= 2$ chosen plaintext-ciphertext pairs (Security proof of Picnic)
  - Difference enumeration + algebraic method (CRYPTO 2021)
  - Algebraic MITM method (Asiacrypt 2022)
- $= 1$ known plaintext-ciphertext pair (Security of Picnic)
  - Guess-and-determine (GnD) attack (ToSC 2020, Asiacrypt 2021)
  - Polynomial method (EUROCRYPT 2021)
  - Polynomial method + GnD (ToSC 2022)

On Difference Enumeration Attack (ToSC 2018)

The general idea:

- Step 1: Compute input and output differences Δ_0 and Δ_r.
- Step 2: Enumerate and store all $\Delta_0 \to \Delta_1 \to \Delta_2 \to \cdots \to \Delta_i$.
- Step 3: Enumerate all $\Delta_r \to \Delta_{r-1} \to \cdots \to \Delta_i$ and match Δ_i.
- Step 4: Compute the key from $\Delta_0 \to \Delta_1 \to \Delta_2 \to \cdots \to \Delta_r$.

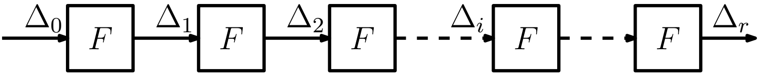


Figure: An r-round differential trail

Difference Enumeration Attack on LowMC (ToSC 2018)

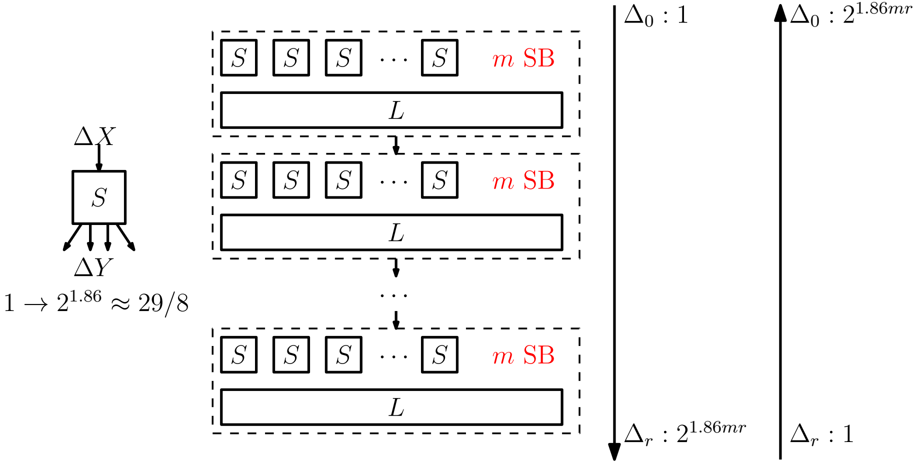


Figure: On the number of possible differences

Difference Enumeration Attack on LowMC (ToSC 2018)

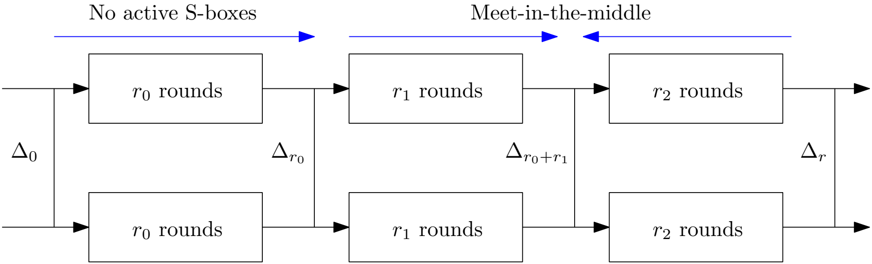


Figure: The original attack framework

Difference Enumeration Attack on LowMC (ToSC 2018)

- Drawbacks:
  - × Too strict constraint $1.86m(r_1 + r_2) \leq n$.
  - × Too inefficient key retrieval.
  - × Too much memory, i.e. $O(2^{1.86mr_1})$.

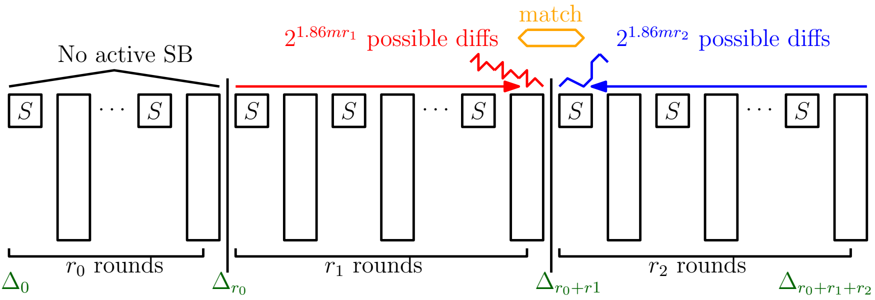


Figure: The original attack framework

Difference Enumeration + Algebraic (CRYPTO 2021)

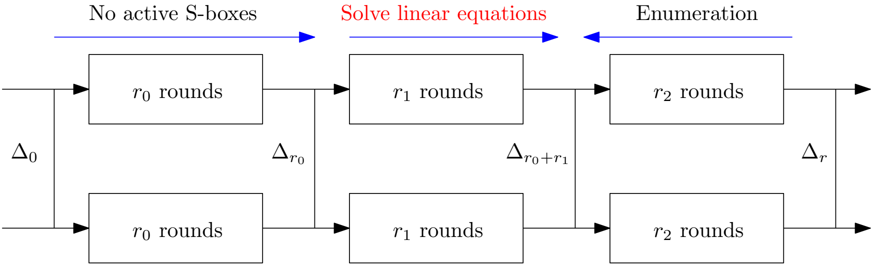


Figure: The new framework proposed at CRYPTO 2021

- Drawbacks:
  - × The linear equation system cannot be under-determined.
  - × The key recovery still relies on guess-and-determine.

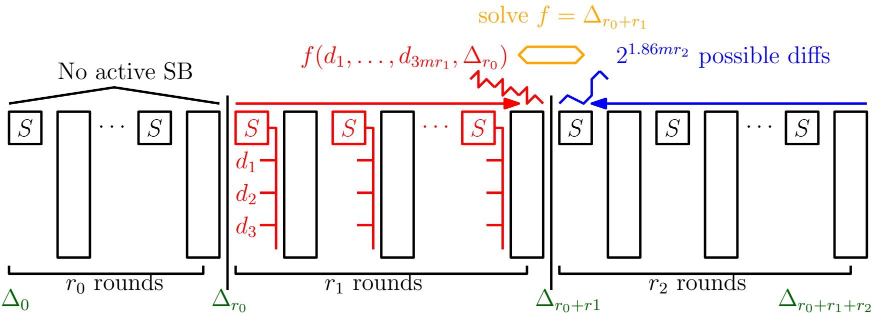


Figure: The new framework proposed at CRYPTO 2021

Further Improvements

- Problems left:
  - How to further reduce the memory complexity of the original difference enumeration attack?
  - How to further extend r_1?
    - $1.86mr_1 < k$ (difference enumeration).
    - $3mr_1 \leq n$ (difference enumeration + algebraic).
    - Can we use additional memory to extend r_0?
    - What to store in advance?
  - T_k is still exponential in k.
    - How to further optimize T_k to allow larger $r_1 + r_2$.

The Algebraic MITM Attack Framework

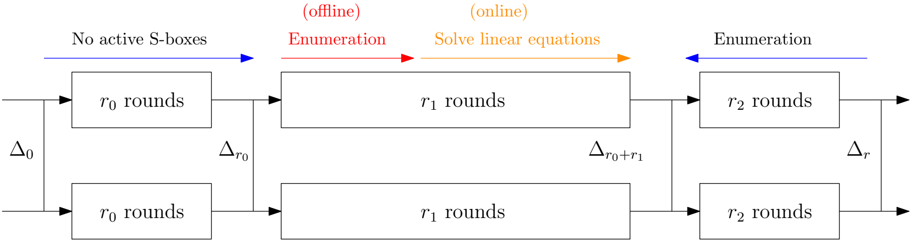
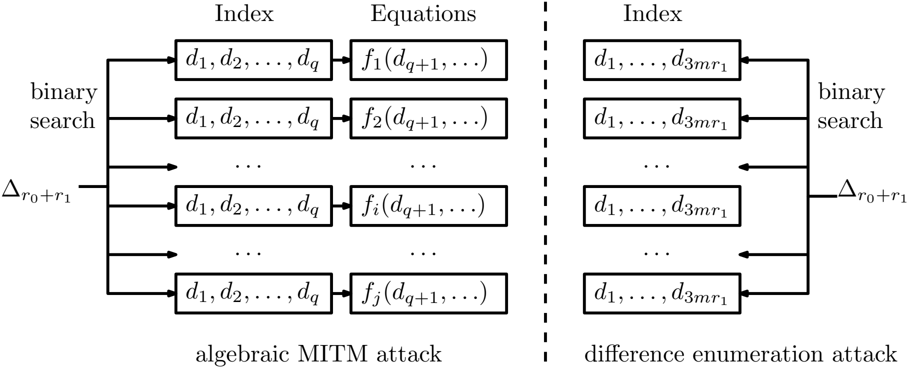


Figure: The algebraic MITM attack framework

The Algebraic MITM Method to Extend r_1



■ Imagination: Given any $\Delta_{r_0+r_1}$, we can directly determine some d_i and compute the remaining d_i by solving linear equations $f = \Delta_{r_0+r_1}$.

The Algebraic MITM Method to Extend r_1

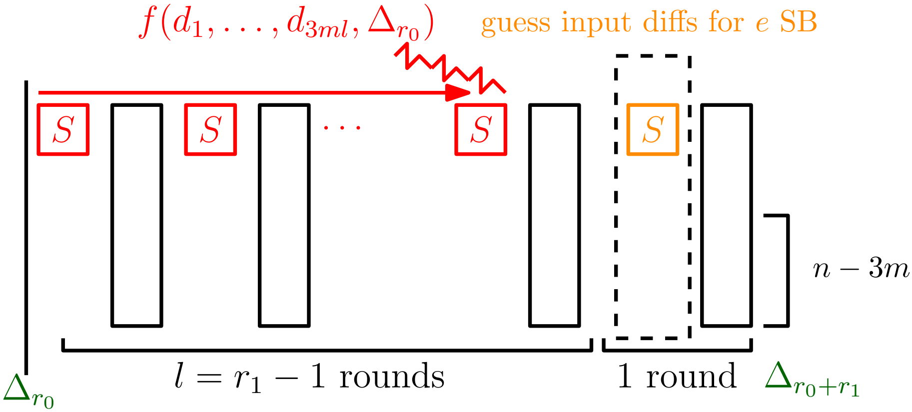


Figure: Illustration of the new idea

- An underdetermined linear equation system, i.e. $3ml < 3e + n - 3m$.

The Algebraic MITM Method to Extend r_1

Pre-processing procedure:

1. Construct the coefficient matrix M for f. M is of size $3ml \times 3ml$ where $r_1 = l + 1$.

$$M \cdot (d_1, d_2, \ldots, d_{3ml})^T \oplus \alpha = \gamma.$$

2. Let $M = M_0 || M_1$ where M_0 represents the first q columns of M. Apply Guassian elimination to M_1 until it becomes the reduced row echelon form and denote the transformation matrix by Q_1.
3. We have

$$Q_1 \cdot M_0(d_1, \ldots, d_q)^T \oplus Q_1 \cdot M_1(d_{q+1}, \ldots, d_{3ml})^T + Q_1(\alpha) = Q_1(\gamma).$$

4. We know that the last

$$\omega = n - 3m + 3e - rank(Q_1 \cdot M_1)$$

rows in $Q_0 = Q_1 \cdot M_1$ are all zero.

The Algebraic MITM Method to Extend r_1

Pre-processing procedure:

5. We obtain ω linear equations only in terms of $(d_1, \dots, d_q)$ and denote them by
$$P_0'(d_1, \dots, d_q)^T \oplus \epsilon' = \beta'$$
where P_0' is deduced from $Q_1 \cdot M_0$ and
$$\begin{aligned} \epsilon &= Q_1(\alpha), \\ \epsilon' &= \epsilon[n-3m+3e-\omega+1 : n-3m+3e], \\ \beta &= Q_1(\gamma), \\ \beta' &= \beta[n-3m+3e-\omega+1 : n-3m+3e]. \end{aligned}$$

5. Record P_0', ϵ', Q_1.

The Algebraic MITM Method to Extend r_1

The offline phase:

1. Let

$$q = 3t.$$

Enumerate all the $2^{1.86t}$ possible values of $(d_1, \ldots, d_{3t})$ and compute the corresponding β' with

$$P_0'(d_1, \ldots, d_q)^T \oplus \epsilon' = \beta'.$$

Store $(\beta', d_1, \ldots, d_{3t})$ in a table D_u.

2. Analysis: β' is an ω-bit value and hence each β' in D_u corresponds to on average $2^{1.86t}/2^{\omega}$ values of $(d_1, \ldots, d_{3t})$.

Hence,

$$\text{Time} : 2^{1.86t} < 2^k. \quad \text{Memory} : O(2^{1.86t}).$$

The Algebraic MITM Method to Extend r_1

The online phase:

1. Given any challenge γ that depends on $\Delta_{r_0+r_1}$ and some guessed output differences in e S-boxes, compute

$$\begin{aligned} \beta &= Q_1(\gamma), \\ \beta' &= \beta[n-3m+3e-\omega+1 : n-3m+3e]. \end{aligned}$$

2. Retrieve $(d_1, \ldots, d_{3t})$ from D_u according to β'.
3. Solve the first $rank(Q_1 \cdot M_1) = n - 3m + 3e - \omega$ rows of the following equation system:

$$Q_1 \cdot M_0(d_1, \ldots, d_{3t})^T \oplus Q_1 \cdot M_1(d_{3t+1}, \ldots, d_{3ml})^T + Q_1(\alpha) = Q_1(\gamma),$$

where there are only $3ml - 3t$ variables. Check the solution and obtain a trail $\Delta_0 \to \cdots \to \Delta_{r_0+r_1+r_2}$.

The Algebraic MITM Method to Extend r_1

The online phase (analysis):

Each β' corresponds to about $2^{1.86t-\omega}$ different $(d_1, \ldots, d_{3t})$ in D_u:

$$\begin{aligned} 1\ \beta' &\rightarrow 2^{1.86t-\omega}\ (d_1, \ldots, d_{3t}) \\ &\rightarrow 2^{1.86t-\omega} \times 2^{3ml-3t-(n-3m+3e-\omega)}\ (d_1, \ldots, d_{3ml}) \\ &= 2^{3mr_1-n-3e-1.14t}\ (d_1, \ldots, d_{3ml}) \end{aligned}$$

Hence,

$$\mathrm{Time}: \quad 2^{max(1.86(mr_2+e), 1.86(mr_2+e)+3mr_1-n-3e-1.14t)} < 2^k,$$

$$\begin{aligned} \Rightarrow \quad & 1.86(mr_2+e) < k, \\ & 1.86(mr_2+e) + 3mr_1 - n - 3e - 1.14t < k. \end{aligned}$$

The Algebraic MITM Method to Extend r_1

The concrete examples:

■ Case 1: $m = 1$, $n = k = 128$

r_1	43	44	...	61	62	63	64	65	66	67	68
t	1	3	...	47	50	53	55	58	61	63	66
$\mathtt{M}_0$	78.1	79.9	...	111.6	113.4	115.3	117.1	119.0	120.9	122.7	124.6
$\mathtt{M}_1$	1.8	5.5	...	87.4	93	98.5	102.3	107.8	113.4	117.1	122.7

$$\begin{aligned} \mathtt{M}_0 &= 1.86m(r_1 - 1) \text{ [difference enumeration]}, \\ \mathtt{M}_1 &= 1.86t \text{ [algebraic MITM]}. \end{aligned}$$

✓ Improve the memory complexity of the difference enumeration attack.

The Algebraic MITM Method to Extend r_1

The concrete examples:

- Case 2: $m = 10$, $n = k = 128$
  - $r_1 = 7$: $2^{111.6}$ (difference enumeration)
  - $r_1 = 7$: 2^{93} (algebraic MITM) [$t = 50, e = 8$]

✓ Improve the memory complexity of the difference enumeration attack.

The Algebraic MITM Method to Extend r_1

The concrete examples:

■ Case 3: $m = 1$, $n = 1024, k = 128$

- $r_1 = 68$: $2^{126.4}$ (difference enumeration)
- $r_1 = 342$: $O(1)$ (algebraic)
- $r_1 = 367$: $2^{126.48}$ (algebraic MITM) [$t = 68, e = 0$]

✓ Extend r_1 using additional memory

Improving the Key Recovery

Consequences caused by the algebraic MITM method:

$\Rightarrow$ r_1 is larger

$\Rightarrow$ $2^{1.86m(r_1+r_2)-n}$ is larger, i.e. more candidate trails left

$\Rightarrow$ Retrieving keys from more candidate trails is required

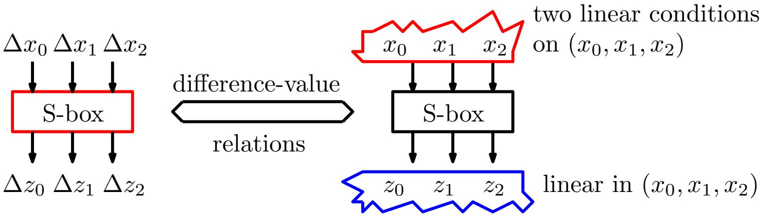
$\Rightarrow$ Optimizing the time to retrieve keys from a differential trail is required

Improving the Key Recovery

- Some related properties of the LowMC S-box.

Property 1 (CRYPTO 2021)

For each valid non-zero difference transition $(\Delta x_0, \Delta x_1, \Delta x_2) \to (\Delta z_0, \Delta z_1, \Delta z_2)$, the inputs conforming to such a difference transition will form an affine space of dimension 1. In addition, (z_0, z_1, z_2) becomes linear in (x_0, x_1, x_2), i.e. the S-box is freely linearized for a valid non-zero difference transition. A similar property also applies to the inverse of the S-box.



Improving the Key Recovery

- Some related properties of the LowMC S-box.

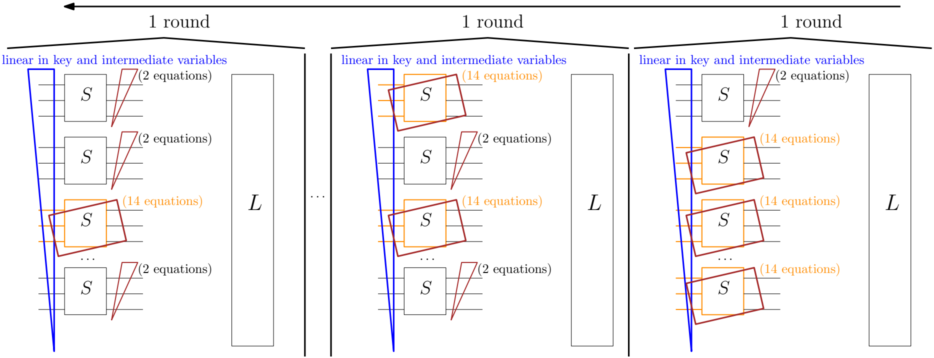
Property 2 (ToSC 2022)

For the input (x_0, x_1, x_2) and output (z_0, z_1, z_2) of the S-box, there are 14 linearly independent quadratic equations:

$$
\begin{aligned}
& z_0 = x_0 \oplus x_1x_2,\; z_1 = x_0 \oplus x_1 \oplus x_0x_2,\; z_2 = x_0 \oplus x_1 \oplus x_2 \oplus x_0x_1,\\
& x_0 = z_0 \oplus z_1 \oplus z_1z_2,\; x_1 = z_1 \oplus z_0z_2,\; x_2 = z_0 \oplus z_1 \oplus z_2 \oplus z_0z_1,\\
& z_0x_1 = x_0x_1 \oplus x_1x_2,\; z_0x_2 = x_0x_2 \oplus x_1x_2,\; z_1x_0 = x_0 \oplus x_0x_1 \oplus x_0x_2,\\
& z_1x_2 = x_1x_2,\; z_2x_0 = x_0 \oplus x_0x_2,\; z_2x_1 = x_1 \oplus x_1x_2,\\
& z_0x_0 \oplus x_0 = z_1x_1 \oplus x_0x_1 \oplus x_1,\\
& z_1x_1 \oplus x_0x_1 \oplus x_1 = z_2x_2 \oplus x_0x_2 \oplus x_1x_2 \oplus x_2.
\end{aligned}
$$

Improving the Key Recovery

Improve the key retrieval by solving an overdefined quadratic equation system (no more guess-and-determine + solving a linear equation system)



Improving the Key Recovery

1. Initialize two counters a and b as 0, where a and b denotes the number of active and inactive S-boxes, respectively.
2. Process the S-box one by one and round by round backwards.
   - 2.1 If the S-box is active, $a = a + 1$ and linearize the S-box for free.
   - 2.2 If the S-box is inactive, $b = b + 1$ and introduce 3 intermediate variables to represent its 3 input bits.
3. If

$$2a \geq k + 3b$$

or

$$2a < k + 3b,\ 14b \geq \binom{k + 3b - 2a}{1} + \binom{k + 3b - 2a}{2},$$

solve the equation system with the linearization technique.

4. Check the key and exit.

$2a$: # linear equations $14b$: # quadratic equations $k + 3b$: # variables

Improving the Key Recovery

1. To make the key recovery phase work efficiently, we only consider constrained candidate trails and the success probability of our attack is about 0.5 since we will not perform the key recovery for about half of the total candidate trails.
2. The time complexity to retrieve the key from a given trail is about $O(1)$ when compared with the number of bit operations of the LowMC encryption, i.e. $T_k \approx O(1)$.

Application to LowMC

Table: Summary of the attacks on LowMC, where D, T, M, Pro. and $R-r$ represent the $\log_2$ data/time/memory complexity, success probability and security margin, respectively. Moreover, $-$ represents negligible memory.

n	k	m	D	R	r_0	r_1	r_2	t	e	r	D	T	M	Pro.	$R-r$
128	128	1	1	182	42	43	67	0	0	152	1	124.62	$-$	1	30
					42	68	67	66	0	177	1	125.38	122.76	0.56	5
128	128	10	1	20	4	5	6	0	0	15	1	122.8	$-$	1	5
					4	7	6	53	7	17	1	125.2	98.58	0.56	3
192	192	1	1	273	64	64	101	0	0	229	1	187.86	$-$	1	44
					64	101	102	98	0	267	1	189.72	182.28	0.51	6
192	192	10	1	30	6	7	10	0	0	23	1	186	$-$	1	7
					6	9	10	67	2	25	1	189.72	124.62	0.51	5

Application to LowMC

Table: Summary of the attacks on LowMC, where D, T, M, Pro. and $R - r$ represent the $\log_2$ data/time/memory complexity, success probability and security margin, respectively. Moreover, $-$ represents negligible memory.

n	k	m	D	R	r_0	r_1	r_2	t	e	r	D	T	M	Pro.	$R - r$
256	256	1	1	363	85	86	137	0	0	306	1	254.82	$-$	1	57
					85	136	136	133	0	357	1	253.34	247.38	0.54	9
256	256	10	1	38	8	9	13	0	0	30	1	241.8	$-$	1	8
					8	13	13	101	6	34	1	253.82	187.86	0.54	4
1024	128	1	1	776	341	342	66	0	0	749	1	122.76	$-$	1	27
					341	367	68	68	0	776	1	127.48	126.48	1	0
1024	256	1	1	819	341	342	136	0	0	819	1	253	$-$	1	0
					341	393	136	136	0	870	1	253.96	252.96	1	-51

Application to LowMC-M v2

Update the security margins of LowMC-M v2:

n	k	m	R	previous security margin $\rightarrow$ new security margin
128	128	1	294	$44 \rightarrow 17$
		2	147	$22 \rightarrow 9$
		3	99	$16 \rightarrow 7$
		10	32	$7 \rightarrow 5$
256	256	1	555	$59 \rightarrow 9$
		3	186	$22 \rightarrow 5$
		20	30	$6 \rightarrow 4$

Summary

1. New algebraic attacks on LowMC are found and the feature of partial nonlinear layers is highly related to the improvement (algebraic MITM strategy).
2. The key recovery is significantly improved by solving an overdefined quadratic equation system rather than a linear equation system.
3. Can we further improve the attack? E.g. can we extend r_2? In all attacks, the constraint is $1.86mr_2 < k$.

Thank you