Improved Attacks on LowMC with Algebraic Techniques

Yimeng Sun[1,3] Jiamin Cui[1,3] Meiqin Wang[1,2,3]

[1]School of Cyber Science and Technology, Shandong University, Qingdao, China

[2]Quan Cheng Shandong Laboratory, Jinan, China

[3]Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China

March, 2024

Table of Contents

- Proposed at Eurocrypt 2015
- Flexible parameters (partial or complete Sbox layers...)
- The underlying block cipher of the digital signature algorithm PICNIC

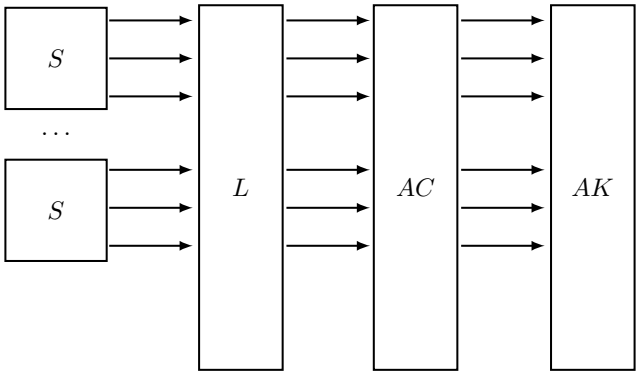


Figure 1: The round function of LowMC

n: block size k: key size m: # S-boxes per round

The Picnic Setting

Security

The Security of PICNIC is directly related to the difficulty of recovering the secret key K from a single plaintext/ciphertext (p, c).

Previous Works

- LowMC → LowMCv2
  - Higher-order differential attack (ICISC 2015)
  - Interpolation attack (Asiacrypt 2015)
- LowMCv2 → LowMCv3
  - Difference enumeration attack (ToSC 2018)
- LowMCv3
  - 2 chosen plaintext-ciphertext pairs
    - Difference enumeration + algebraic method (CRYPTO 2021)
    - Algebraic MITM method (Asiacrypt 2022)
  - 1 known plaintext-ciphertext pair (PICNIC setting)
    - Guess-and-determine (GnD) attack (ToSC 2020, Asiacrypt 2021)
    - Polynomial method (EUROCRYPT 2021)
    - Polynomial method + GnD (Tosc 2022, ePrint 2022)

Problem

Consider a system of m Boolean equations in u variables denoted by $E(x)$ as:

$$E(x) : E_0(x) = E_1(x) = \cdots = E_{m-1}(x) = 0, \tag{1}$$

where the algebraic degree of each E_i is bounded by $\deg(E_i) \leq d$ and $x = (x_0, x_1, \cdots, x_{u-1}) \in \mathbb{F}_2^u$.

- $x \to (y, z) = (y_0, y_1, \cdots, y_{u-u_1-1}, z_0, z_1, \cdots, z_{u_1-1})$.

Problem

How do we solve this equation system?

The first method: the crossbred algorithm for $d = 2$

$E_i(x)$ can be rewritten as follows and each element in M and B is linear and quadratic in y.

$$M \cdot (z_0 z_1, z_0 z_2, \cdots, z_{u_1-2} z_{u_1-1}, z_0, z_1, \cdots, z_{u_1-1})^T = B, \tag{2}$$

$$\xrightarrow{\text{Gauss elimination}} \left(\begin{array}{c|c} E & \star \\ \hline 0 & M'' \end{array}\right) \cdot \begin{pmatrix} z_0 z_1 \\ z_0 z_2 \\ \cdots \\ z_{u_1-2} z_{u_1-1} \\ z_0 \\ z_1 \\ \cdots \\ z_{u_1-1} \end{pmatrix} = \left(\begin{array}{c} * \\ \hline B'' \end{array}\right)$$

$$\rightarrow M'' \cdot (z_0, z_1, \cdots, z_{u_1-1})^T = B''. \tag{3}$$

The second method: Dinur's algorithm for $d \geq 2$

The core idea:

- Randomly take $\ell = u_1 + 1$ different equations from $E(x)$ and construct the new equation system $\tilde{E}(x)$.
- A solution of E must be a solution of $\tilde{E}$. But a solution of $\tilde{E}$ is not always a solution of E. We choose to enumerate the solutions of $\tilde{E}$ instead of E and then verify them by E.
  - $\tilde{A}(x) = \prod_{i=0}^{\ell-1}(\tilde{E}_i(x) \oplus 1)$
    - $\tilde{A}(\hat{x}) = 1$ if and only if $\hat{x}$ is a solution of $\tilde{E}(x)$.
    - Enumerate all possible x such $\tilde{A}(x) = 1$.

Difference enumeration attack

- **difference enumeration phase**: enumerate all the possible differential trails to recover the compact differential trails.
- **key-recovery phase**: retrieve the full key from the recovered differential trails.

Difference enumeration phase

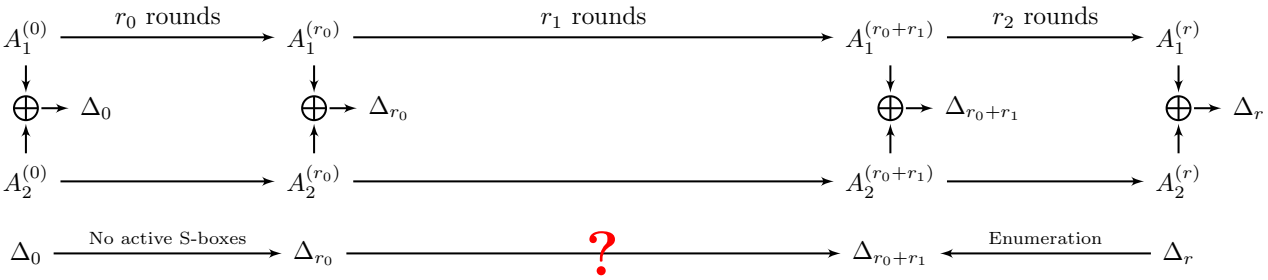


Figure 2: Difference enumeration phase

Difference enumeration phase (Asiacrypt 2022)

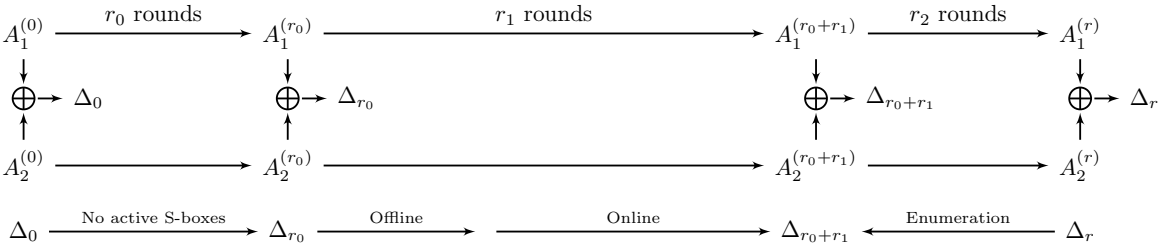


Figure 3: Difference enumeration phase (Asiacrypt 2022)

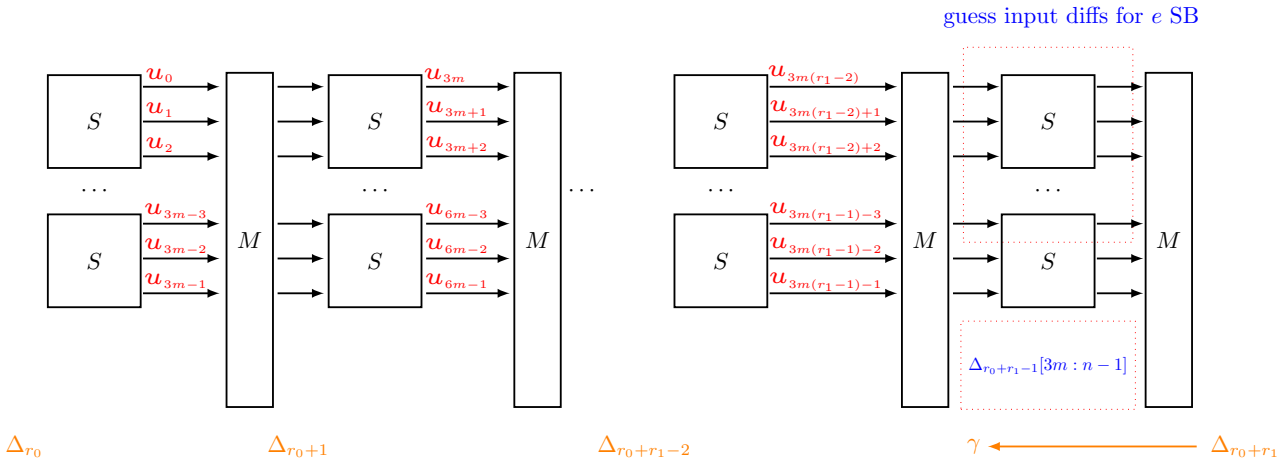


Figure 4: Introduce variables to represent the output differences of the S-boxes

Difference enumeration phase (Asiacrypt 2022)

$$M \cdot (u_0, u_1, \cdots, u_{3m\ell-1})^T \oplus \alpha = \gamma \tag{4}$$

$$\xrightarrow{\text{Gauss elimination}} \left(\begin{array}{c|c} \star & E \\ \hline M_1'' & 0 \end{array}\right) \cdot \begin{pmatrix} u_0 \\ u_1 \\ \cdots \\ u_{3t-1} \\ u_{3t} \\ \cdots \\ u_{3m\ell-1} \end{pmatrix} \oplus \left(\begin{array}{c} \star \\ \hline \alpha'' \end{array}\right) = \left(\begin{array}{c} \star \\ \hline \gamma'' \end{array}\right)$$

$$\longrightarrow M_1'' \cdot (u_0, u_1, \cdots, u_{3t-1})^T \oplus \alpha'' = \gamma''. \tag{5}$$

- The offline phase
  - Insert the tuple $(u_0, u_1, \cdots, u_{3t-1}, \gamma'')$ into the table D_u.
- The online phase
  - Each $\gamma \xleftarrow{D_u} (u_0, u_1, \cdots, u_{3t-1})$.
  - Solve for $(u_{3t}, u_{3t+1}, \cdots, u_{3m\ell-1})$ by enumerating all the free variables.

Problem

Problem

When the number of attacked rounds increases, the time complexity to enumerate all the free variables grows since we still assume them all independent variables.
Is it possible to improve further?

New difference enumeration phase

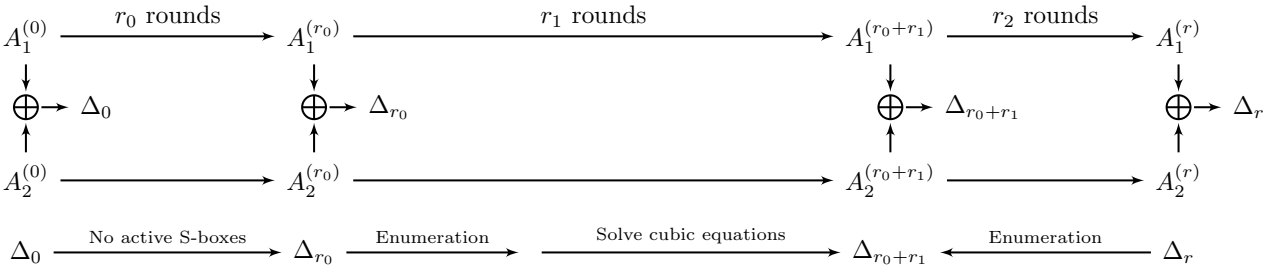


Figure 5: The new difference enumeration attack framework

$$M_1' \cdot (u_0, u_1, \cdots, u_{3t-1})^T \oplus M_2' \cdot (u_{3t}, u_{3t+1}, \cdots, u_{3m\ell-1})^T \oplus \alpha' = \gamma'.$$

The last $\omega = n - 3m + 3e - rank(M_2')$ rows are all zeros. Let

$$\begin{aligned} \omega' &= n - 3m + 3e - (3m\ell - 3t) \\ &= n - 3mr_1 + 3e + 3t. \end{aligned}$$

We can immediately obtain ω' linear equations involving $3t$ variables as:

$$\gamma''' = M_1''' \cdot (u_0, u_1, \cdots, u_{3t-1})^T \oplus \alpha''', \tag{6}$$

- The new offline phase
  - Insert the tuple $(u_0, u_1, \cdots, u_{3t-1}, \gamma''')$ into the table D_u'.

The new online phase

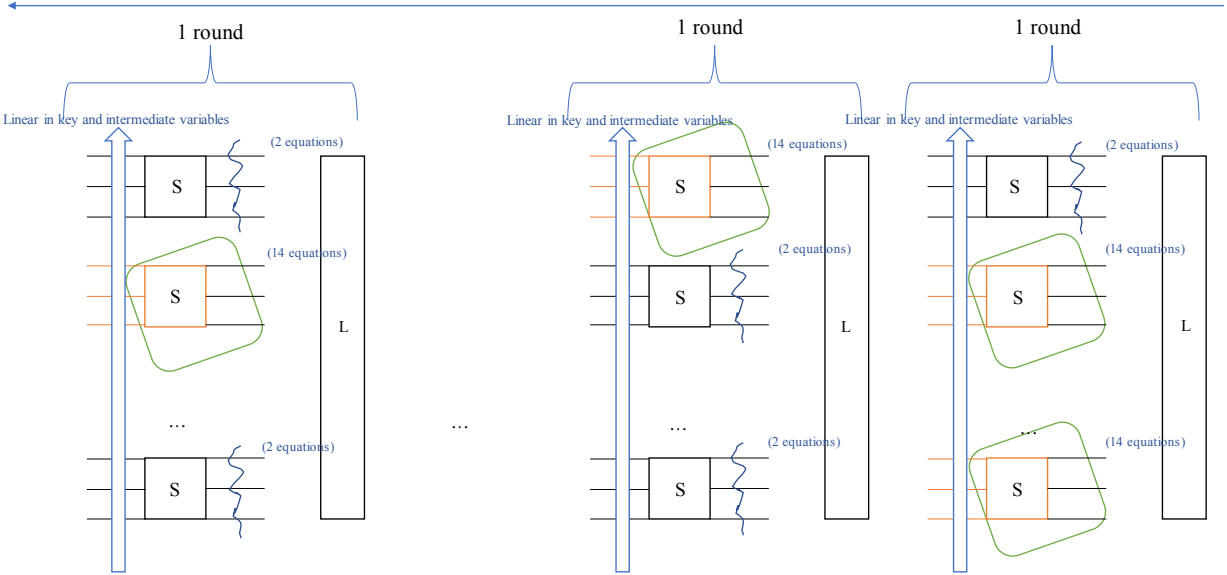
Observation1

Denote the input difference and output difference of the 3-bit LowMC S-box by $(\Delta x_0, \Delta x_1, \Delta x_2)$ and $(\Delta z_0, \Delta z_1, \Delta z_2)$, respectively. The following 2 cubic equations are sufficient to describe its DDT:

$$(1 \oplus \Delta x_0)(1 \oplus \Delta x_1)(1 \oplus \Delta x_2) = (1 \oplus \Delta z_0)(1 \oplus \Delta z_1)(1 \oplus \Delta z_2),$$
$$(1 \oplus \Delta x_0)(1 \oplus \Delta x_1)(1 \oplus \Delta x_2) = \Delta x_0 \Delta z_0 \oplus \Delta x_1 \Delta z_1 \oplus \Delta x_2 \Delta z_2 \oplus 1.$$

- The new online phase.
  - Each $\gamma \xleftarrow{D_u} (u_0, u_1, \cdots, u_{3t-1})$.
  - Obtain $2g$ equations of degree 3 based on the Observation1 and perform Dinur's algorithm on them and find the solution.

Key-recovery phase



- active S-box: 2 linear equations
- inactive S-box: 14 quadratic equations

a: # active S-boxes b: # inactive S-boxes $k + 3b$: # variables

Key-recovery phase (Asiacrypt 2022)

- $2a \geq k + 3b$: Gaussian elimination.
- $2a < k + 3b$: Gaussian elimination + Linearization technique.

Problem

Lower success probability.

New key-recovery phase

- $2a \geq k + 3b$: Gaussian elimination.
- $2a < k + 3b$
  - For the first λ inactive S-boxes, we can construct 14λ quadratic equations in these free variables, $1 \leq \lambda \leq b$.
  - Gaussian elimination + the crossbred algorithm.
- In order to facilitate the complexity calculation, we have constrained the number of a to the lower bound, so that the success probability of our attack is about 0.9.

Results

n	k	m	D	R	r_0	r_1	r_2	r	T	M	Pro.	$R-r$
128	128	1	2	182	42	68	67	177	125.38	122.76	0.56	5
					42	67	68	177	123.21	117.18	0.90	5
					42	69	68	179	126.91	120.90	0.90	3
128	128	10	2	20	4	7	6	17	125.2	98.58	0.56	3
					4	7	6	17	117.59	106.02	0.90	3
192	192	1	2	273	64	101	102	267	189.72	182.28	0.51	6
					64	101	102	267	186.12	178.56	0.95	6
					64	104	102	270	191.68	184.14	0.95	3
192	192	10	2	30	6	9	10	25	189.72	124.62	0.51	5
					6	10	9	25	165.38	159.96	0.95	5
					6	11	9	26	183.93	178.56	0.95	4
256	256	1	2	363	85	136	136	357	253.34	247.38	0.54	6
					85	135	137	357	250.46	241.8	0.97	6
					85	138	137	360	255.44	249.24	0.97	3
256	256	10	2	38	8	13	13	34	253.82	187.86	0.54	4
					8	13	13	34	231.08	217.62	0.97	4
					8	14	13	35	249.63	236.22	0.97	3

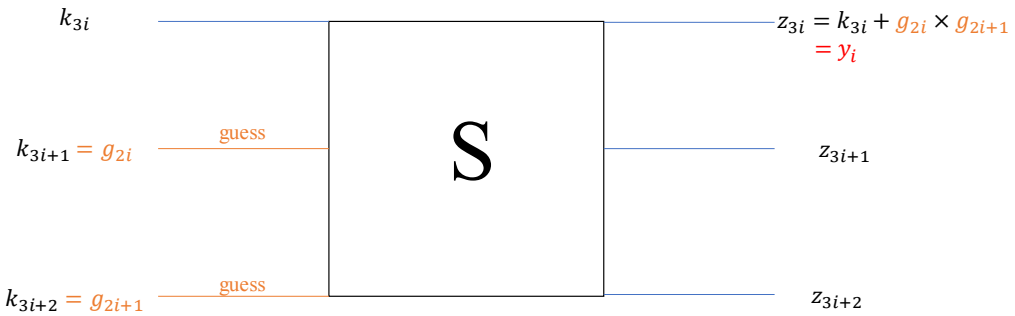


Figure 6: $S(k_{3i}, k_{3i+1}, k_{3i+2}) = (z_{3i}, z_{3i+1}, z_{3i+2})$

If we guess two input bits as $k_{3i+1} = g_{2i}$ and $k_{3i+2} = g_{2i+1}$, the 3-bit key variable takes value from the set

$$B^{g_{2i}g_{2i+1}} = \{(y_i \oplus g_{2i}g_{2i+1}, g_{2i}, g_{2i+1}) \mid y_i \in \{0,1\}\},$$

where $g_{2i}, g_{2i+1} \in \{0,1\}$.

- For example, consider the case when $g_{2i} = 0$ and $g_{2i+1} = 0$, then we have $B^{00} = \{(0,0,0),(1,0,0)\}$.

Construct the equation system

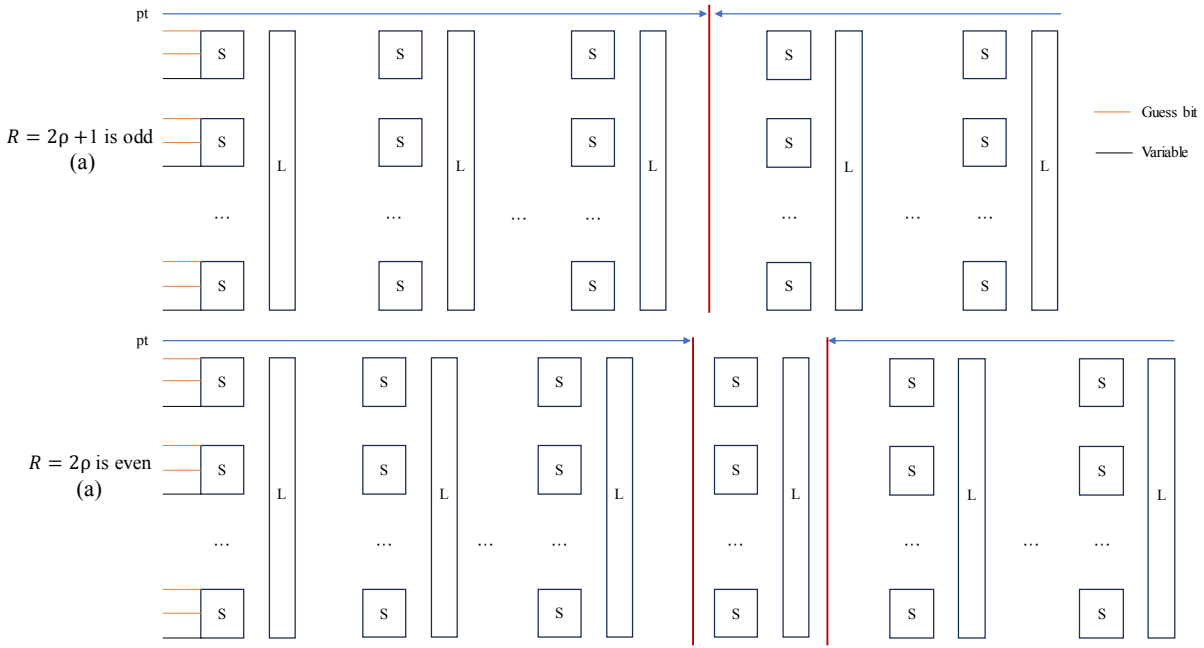


Figure 7: The attack setup for even and odd rounds for LowMC instances. The first round is linearized.

Denote the key variables by $K = (k_0, k_1, \cdots, k_{n-1})$, $n = k = 3m$.

Associated Function

- After guessing $2m$ bits $G = (g_0, g_1, \cdots, g_{2m-1}) \in \{0,1\}^{2m}$, we obtain n equations over the partial space $K \in B^G = B^{g_0 g_1} \times \cdots \times B^{g_{2m-2} g_{2m-1}}$, denoted by

$$E_G(K) \colon E_{G,0}(K) = E_1(K) = \cdots = E_{G,n-1}(K) = 0.$$

- For any given $G = (g_0, g_1, \cdots, g_{2m-1})$,

$$E_{G,i}(K) = \boldsymbol{E}_{G,i}(Y), 0 \leq i \leq n-1.$$

- We call $\boldsymbol{E}_{G,i}$ the **associated function** of $E_{G,i}$ and Y the **associated vector** of K.

The details of the algorithm

- $K \to (K_1|K_2)$, $G \to (G_1|G_2)$.
- Randomly select l equations from the n equations and denote the new equation system as:

$$\tilde{E}_G(K_1, K_2)\colon R_{G,0}(K_1, K_2) = \cdots = R_{G,l-1}(K_1, K_2) = 0.$$

Note

A solution of E_G must be a solution of $\tilde{E}_G$, but a solution of $\tilde{E}_G$ is not always a solution of E_G. We choose to enumerate solutions of $\tilde{E}_G$ and then verify them by E_G.

The details of the algorithm

- $\tilde{E}_G(K_1^*, K_2^*) \longleftrightarrow \tilde{A}_G(K_1^*, K_2^*) = 1 \longleftrightarrow \tilde{F}_G(K_1^*) = 1$
  - $\tilde{A}_G(K_1, K_2) = \prod_{i=0}^{l-1}(R_{G,i}(K_1, K_2) \oplus 1)$
  - $\tilde{F}_G(K_1) = \bigoplus_{K_2 \in B^{G_2}} \tilde{A}_G(K_1, K_2)$
- Let $\tilde{\boldsymbol{F}}_{G,i}[0](Y_1) = \bigoplus_{Y_2 \in \{0,1\}^{m_1}, y_{m-m_1+i}=0} \tilde{\boldsymbol{A}}(Y_1, Y_2)$

Proposition

Assume (K_1^*, K_2^*) to be an isolated solution of $\tilde{E}_G$ and (Y_1^*, Y_2^*) to be the associated vector of (K_1^*, K_2^*).
If $Y_2^* = (y_{m-m_1}^*, y_{m-m_1+1}^*, \cdots, y_{m-1}^*)$, then we have $\tilde{\mathbf{F}}_G(Y_1^*) = 1$ and $\tilde{\mathbf{F}}_{G,i}[0](Y_1^*) = y_{m-m_1+i}^* + 1$ for $i \in [0, m_1 - 1]$.

Results

r	n	k	m	N	n_1	l	T	M	Gray Code	Exh.Search
5	129	129	43	4	9 3	7 2	134.43 133.87	86.46 45.00	136	146
5	192	192	64	4	/ 9 9	/ 7 4	192 196.39 196.78	173 125.38 65.00	199	210
5	255	255	85	4	/ 15 15	/ 11 6	251 256.74 258.92	228 165.52 84.59	262	274
6	192	192	64	4	6 6	5 3	197.96 197.52	128.4 65.59	199	211
6	255	255	85	4	12 9	9 4	259.62 260.54	167.28 86.00	262	275

Summary

- For difference enumeration attacks, we significantly push the security margin of difference instances of LowMC with partial nonlinear layers to 3/4 rounds.
- For the picnic setting, we drastically reduce the memory complexity for 5-/6-round LowMC instances with full nonlinear layers.
- The security evaluation of different instances of LowMC under extremely low data complexity has not yet come to an end.

Thanks for Your Attention!

sunyimeng@mail.sdu.edu.cn for any question!