Cryptanalysis of Full LowMC and LowMC-M with Algebraic Techniques

Fukang Liu, Takanori Isobe, Willi Meier

University of Hyogo, Japan
NICT, Japan,
PRESTO, Japan
FHNW, Switzerland

liufukangs@gmail.com

August 10, 2021

The LowMC Primitive

- LowMC was proposed at Eurocrypt 2015.
- LowMC is designed to be MPC/FHE-friendly.
- Users have flexible choices (affine layers, linear key schedule functions, number of S-boxes per round) for concrete instances of LowMC.

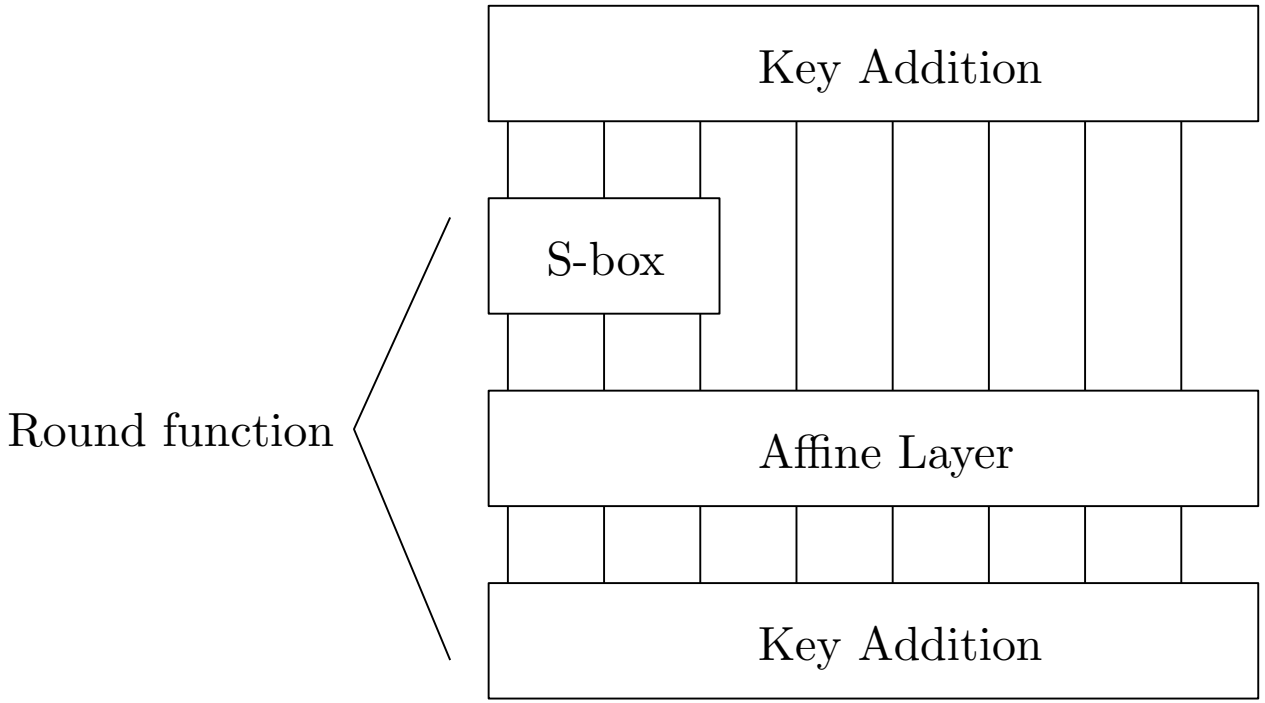


Figure: The round function of LowMC

The Most Important Application: the Picnic Signature

- The Picnic signature scheme was proposed at ACM CCS 2017.
- The security of Picnic relies on the security of LowMC (The difficulty to recover the key from a single plaintext-ciphertext pair).
- LowMC with 10 S-boxes per round is used in Picnic2.
- LowMC with a full S-box layer is used in Picnic3.
- Picnic3 is an alternate 3rd-round candidate in NIST PQC.

Cryptanalytical Results of LowMC

- The higher-order differential attack (ICISC 2015)
- The optimized interpolation attack (Asiacrypt 2015)
- The difference enumeration attack (ToSC 2018)
- The guess-and-determine attack with a plaintext (ToSC 2020)
- The algebraic attack (Eurocrypt 2021) [parallel work]

Difference Enumeration Attacks with MITM

1. Choose an input difference Δ_0 such that there is no active S-box in the first t_0 rounds.
2. Enumerate the differences forwards for t_1 rounds staring from the t_0-th round and store all the reachable state differences
3. Enumerate the differences backwards for $r - t_0 - t_1$ rounds to match the stored reachable state differences.

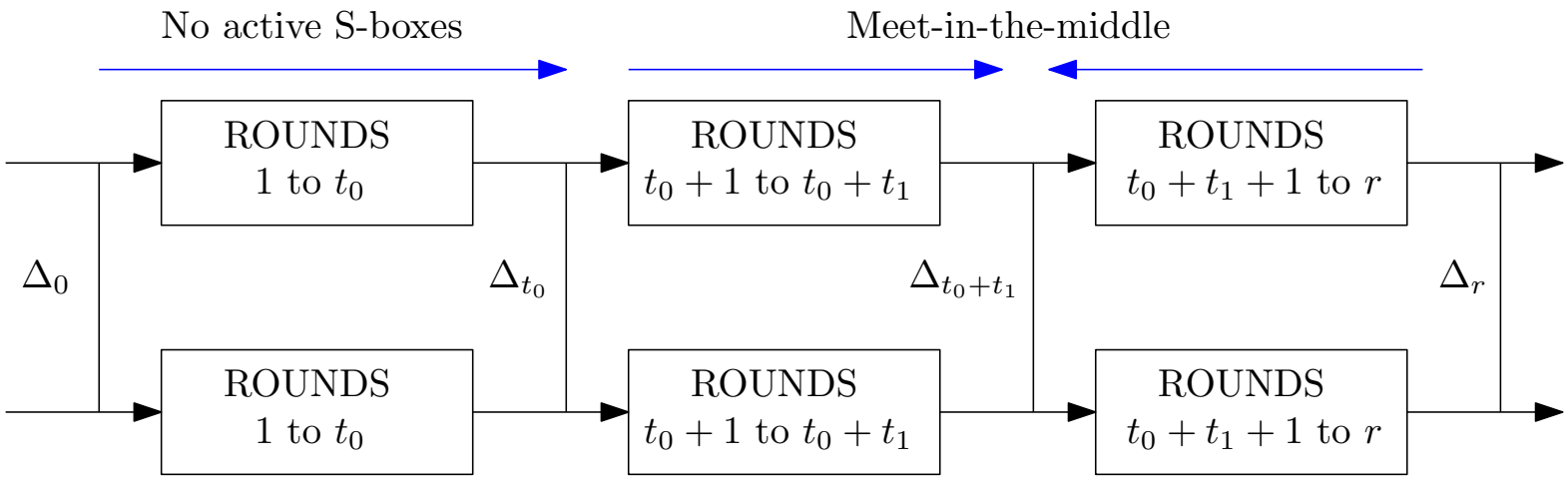


Figure: The framework of the difference enumeration techniques

Difference Enumeration Attacks with MITM

Observation [Rechberger et al., 2018]

For the used 3-bit S-box, the average number of output differences for a uniformly randomly chosen input difference is $3.62 \approx 2^{1.86}$

Constraints in [Rechberger et al., 2018]

There should be on average one valid differential trail left after the matching phase. For attacks on parameters (n, k, m, R) with $k = n$ using two plaintexts, the constraints are $2^{1.86m(r-t_0)} \leq 2^n$ and $t_0 = \lfloor n/m \rfloor$, where n is the state size, k is the key size, m is the number of S-boxes per round and R is the total number of rounds.

The Extended Framework

1. Compute the deterministic differential trail for the first r_0 rounds.
2. Use many plaintexts to find a pair of plaintexts such that there is no active S-box in the last r_3 rounds.
3. Enumerate the differences backwards for r_2 rounds.
4. Compute the difference transitions for the middle r_1 rounds via solving equations.

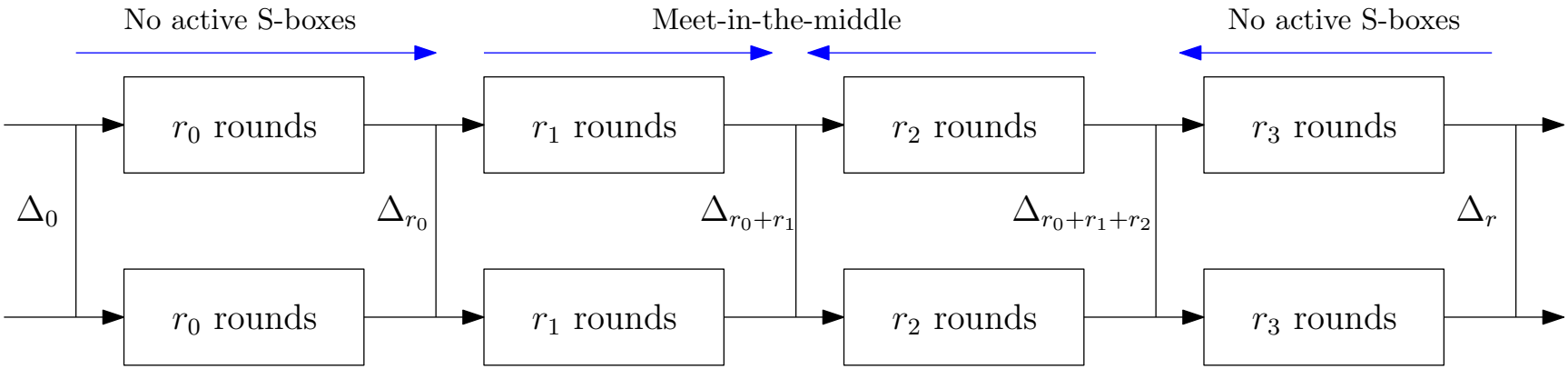


Figure: The extended framework of the difference enumeration techniques

Utilizing Properties of the 3-bit S-box

The specification of the 3-bit S-box:

$$z_0 = x_0 \oplus x_1 x_2,\ \ z_1 = x_0 \oplus x_1 \oplus x_0 x_2,\ \ z_2 = x_0 \oplus x_1 \oplus x_2 \oplus x_0 x_1.$$

Observation 1

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.

Example

When $(\Delta x_0, \Delta x_1, \Delta x_2) = (0, 0, 1)$ and $(\Delta z_0, \Delta z_1, \Delta z_2) = (0, 0, 1)$, it can be derived that $x_0 = 0$ and $x_1 = 0$. Therefore, the expressions of (z_0, z_1, z_2) become $z_0 = 0$, $z_1 = 0$ and $z_2 = x_2$.

Utilizing Properties of the 3-bit S-box

The specification of the 3-bit S-box:

$$z_0 = x_0 \oplus x_1 x_2,\ \ z_1 = x_0 \oplus x_1 \oplus x_0 x_2,\ \ z_2 = x_0 \oplus x_1 \oplus x_2 \oplus x_0 x_1.$$

Observation 2

For each non-zero input difference $(\Delta x_0, \Delta x_1, \Delta x_2)$, its valid output differences form an affine space of dimension 2. A similar property also applies to the inverse of the S-box.

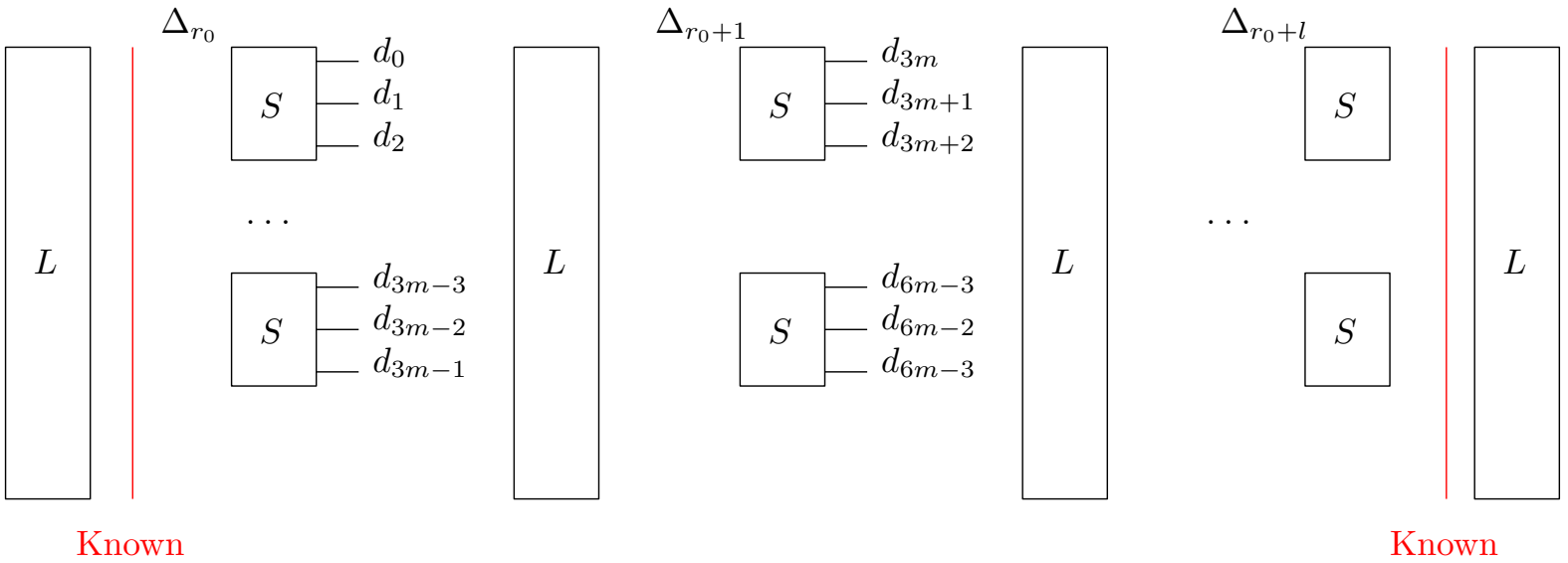
Example

When the input difference is $(0, 1, 1)$, the corresponding valid output differences satisfy $\Delta z_1 \oplus \Delta z_2 = 1$. When the output difference is $(0, 1, 1)$, the corresponding valid input differences satisfy $\Delta x_1 \oplus \Delta x_2 = 1$.

Enumerating Differences Via Solving Equations

For the middle $r_1 = l + 1$ rounds,

1. Introduce $3m(l-1) + 2m = m(3l-1)$ intermediate variables to represent the output difference of each S-box in the middle l rounds.
2. Construct $m + (n - 3m) = n - 2m$ equations in these variables according the known state differences.
3. Solve variables and check the difference transitions in these l rounds.



Enumerating Differences Via Solving Equations

To make r_1 the largest, we have

$$m(3l-1) \leq n - 2m \rightarrow n \geq m(3l+1).$$

To make the attack faster than the brute-force attack, we have

$$m(1.86r_2 + 3r_1 - 2) < n + k, \quad 1.86mr_2 < k.$$

To make $r_1 + r_2$ the largest, the time complexity to enumerate differences is

$$max(2^{1.86mr_2}, 2^{m(1.86r_2+3r_1-2)-n}).$$

Enumerating Differences Via Solving Equations

Advantages of the new strategy to enumerate differences:

1. The memory complexity is negligible since there is no need to store all possible reachable state differences.
2. It allows many possible differential trails to exist after the difference enumeration, while only one valid differential trail is allowed to exist in [Rechberger et al., 2018].

A natural problem

How to find the correct differential trail among all the possible differential trails after difference enumeration?

Our solution to the problem

Efficiently check whether a possible differential trail implies a correct key.

Efficient Key-recovery Techniques with 2 Plaintexts

First, introduce intermediate variables to represent the input of the S-box for the last r_3 rounds. There are in total $3mr_3$ variables.

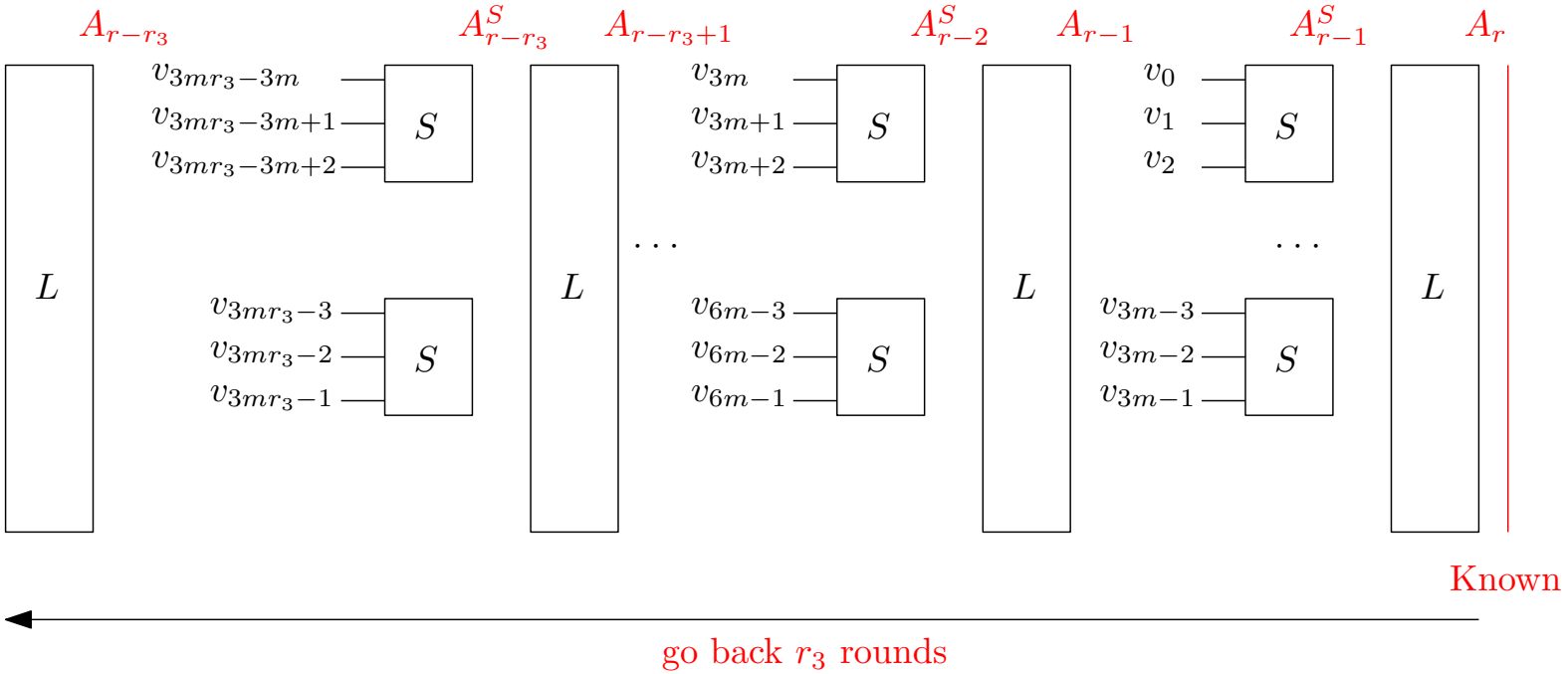


Figure: Linearizing the last r_3 rounds

Efficient Key-recovery Techniques with 2 Plaintexts

According to Observation 1, once an S-box is active, the S-box is freely linearized and there are two linear conditions on the 3 output bits.

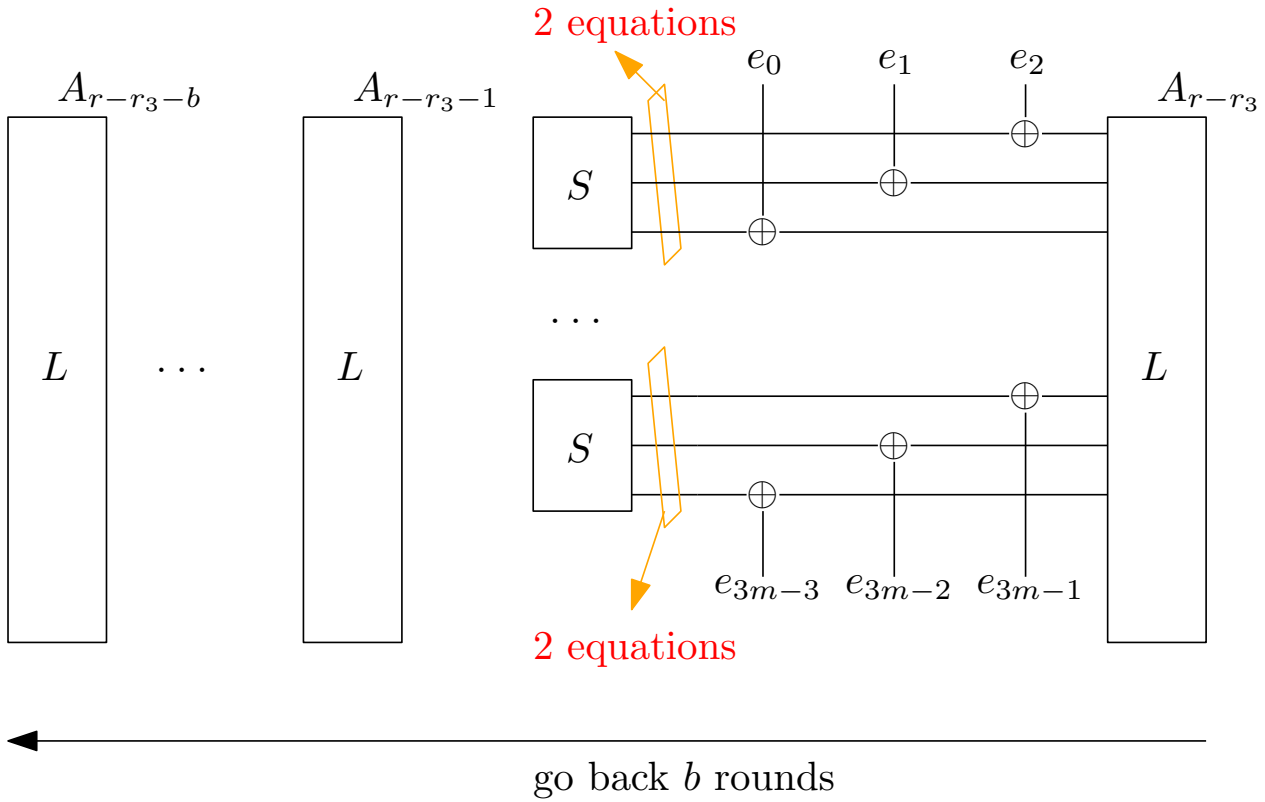


Figure: Extract linear equations from the active S-boxes

Efficient Key-recovery Techniques with 2 Plaintexts

Suppose all S-boxes in the middle $r_1 + r_2$ rounds are active, we can extract at most $2m(r_1 + r_2)$ linear equations in terms of the k-bit key and the $3mr_3$ intermediate variables.

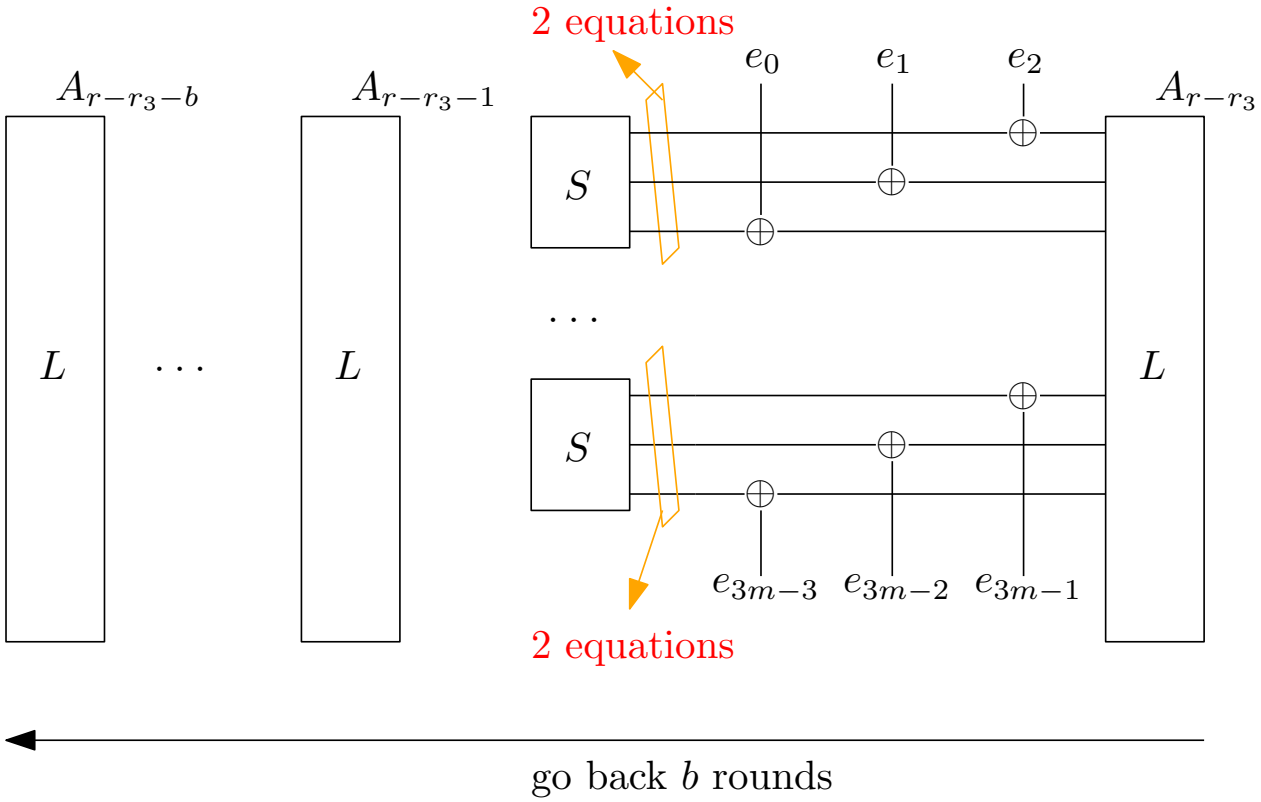


Figure: Extract linear equations from the active S-boxes

Efficient Key-recovery Techniques with 2 Plaintexts

If $2m(r_1 + r_2) > k + 3mr_3$, solving the equation system suggests the key and the key can be easily verified via the plaintext-ciphertext pair. Indeed, we only need to use b rounds, i.e. $2mb \geq k + 3mr_3$ and $b \leq r_1 + r_2$.

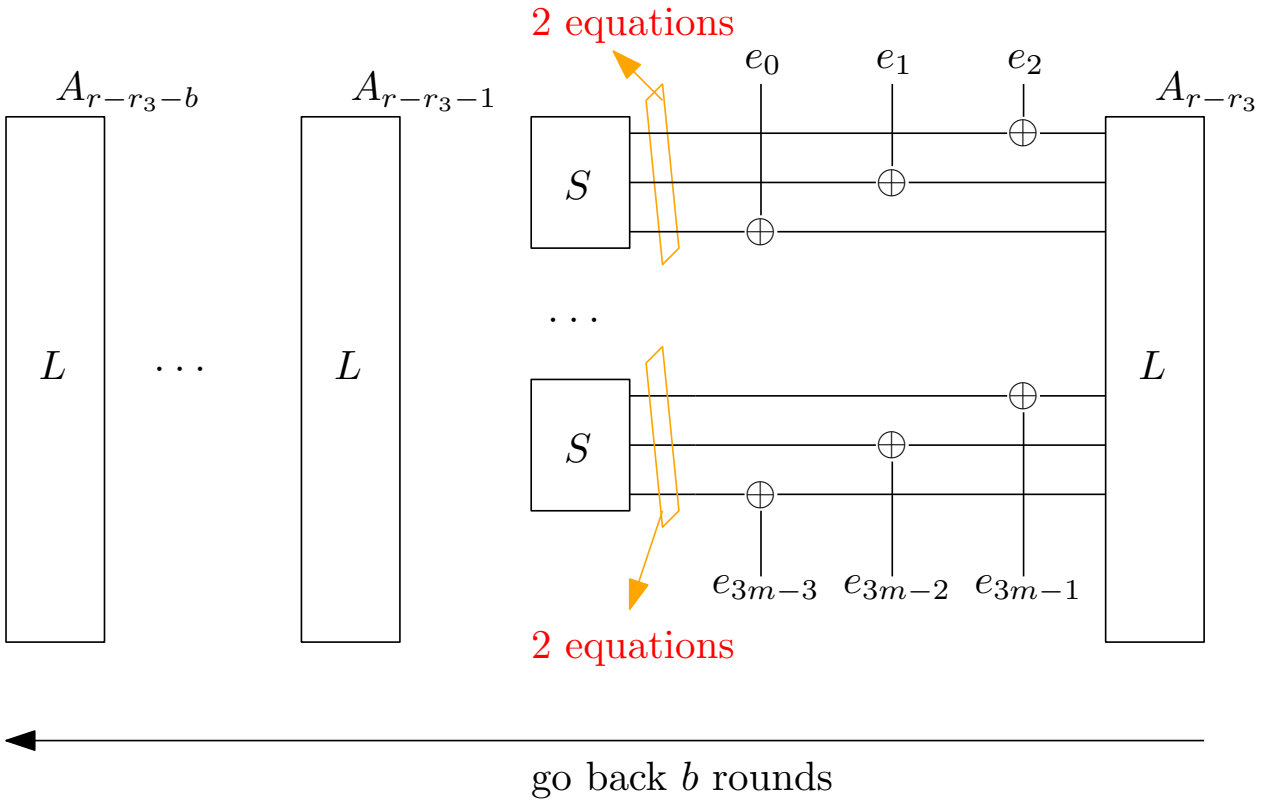


Figure: Extract linear equations from the active S-boxes

Efficient Key-recovery Techniques with 2 Plaintexts

What will happen if there are inactive S-boxes? We just guess two output bits to linearize them. The expectation of the time complexity to retrieve the key is:

$$T \approx N \times 2^{0.46m(b-1)} \times (1 + 2^{2h-1.54m}),$$

where

$$\begin{aligned} h &= \lceil \frac{(k + 3mr_3) - 2m(b-1)}{2} \rceil, \\ N &= 2^{1.86m(r_1+r_2)-n}. \end{aligned}$$

Application to LowMC

With the above algebraic techniques to compute differential trails and retrieve the key from a differential trail, we could break 7 instances of LowMC-M and one instance of LowMC with $n >> k$.

Table: The results for LowMC with a partial S-box layer

n	k	m	D	R	r_0	r_1	r_2	r_3	r	Data	Time
128	128	1	1	182	42	43	67	0	152	2	$2^{124.62}$
128	128	10	1	20	4	5	6	0	15	2	$2^{122.8}$
192	192	1	1	273	64	64	101	0	229	2	$2^{187.86}$
192	192	10	1	30	6	7	10	0	23	2	2^{186}
256	256	1	1	363	85	86	137	0	306	2	$2^{254.82}$
256	256	10	1	38	8	9	13	0	30	2	$2^{241.8}$
1024	128	1	1	776	341	342	66	0	749	2	$2^{122.76}$
1024	256	1	1	819	341	342	136	0	819	2	2^{253}

Application to LowMC-M

With the above algebraic techniques to compute differential trails and retrieve the key from a differential trail, we could break 7 instances of LowMC-M and one instance of LowMC with $n >> k$.

Table: The results for LowMC-M

n	k	m	D	R	r_0	r_1	r_2	r_3	r	Data	Time
128	128	1	64	208	122	43	64	21	250	2^{64}	2^{120}
128	128	2	64	104	61	22	32	10	125	2^{61}	2^{120}
128	128	3	64	70	40	15	21	7	83	2^{64}	$2^{118.18}$
128	128	10	64	23	12	5	6	2	25	2^{61}	2^{118}
256	256	1	64	384	253	86	136	21	496	2^{64}	$2^{252.96}$
256	256	3	64	129	83	29	45	7	164	2^{64}	$2^{250.1}$
256	256	20	64	21	12	5	6	1	24	2^{61}	2^{232}

Refining the Attack Framework for the Full S-box Layer

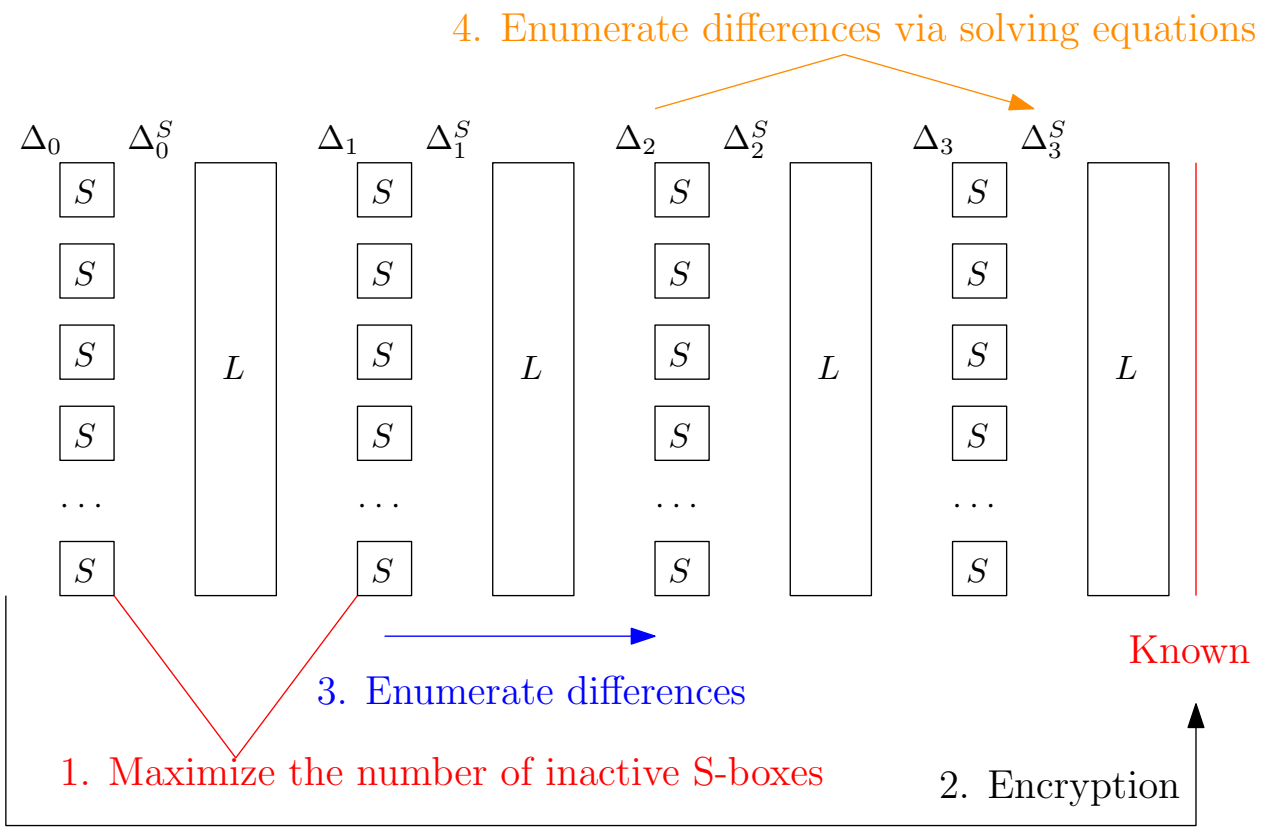


Figure: The attack framework for 4-round LowMC with a full S-box layer

Refining the Attack Framework for the Full S-box Layer

- Find a suitable assignment for Δ_0^S such that the number of inactive S-boxes in the 2nd round can be maximized and there is only one active S-box in the first round. Denote the number of inactive S-boxes in the 2nd round by q.
- Choose a value for Δ_0 such that it can reach Δ_0^S and encrypt two arbitrary plaintexts whose difference equals Δ_0. Collect the corresponding ciphertexts and compute Δ_3^S.
- Enumerate 4^{m-q} possible difference transitions from Δ_1 to Δ_2. For each possible difference transition, move to Step 4.
- For each obtained Δ_2, we enumerate the possible difference transitions from Δ_2 to Δ_3^S via solving a linear equation system and then check the validity of the obtained differential trail by solving key variables.

Attacks on 4-Round LowMC with Full S-box Layers

With a similar method to compute the expectation of the time complexity, all the recommended parameters for LowMC in Picnic3 are insecure if the attacker can choose two plaintexts to encrypt.

Table: The results for 4-round LowMC with a full S-box layer

n	k	m	D	R	Data	Time	Pro.
129	129	43	1	4	2	$2^{122.6}$	0.62
129	129	43	1	4	2	2^{104}	0.24
192	192	64	1	4	2	$2^{187.6}$	0.99
192	192	64	1	4	2	2^{180}	0.82
192	192	64	1	4	2	2^{156}	0.247
255	255	85	1	4	2	$2^{246.6}$	0.986
255	255	85	1	4	2	$2^{236.6}$	0.848
255	255	85	1	4	2	2^{208}	0.2465

Conclusion

1. Devise efficient attacks on LowMC with only 2 chosen plaintexts and negligible memory. Parameters with 2 plaintexts are required to be secure in the Picnic security proof and hence such attacks are meaningful. However, our attack on 4-round LowMC does not threaten the security of Picnic3.
2. The efficiency of the attacks contributes to some special properties of the 3-bit S-box.
3. Break 7 instances of LowMC-M, a backdoor construction proposed in CRYPTO 2020.

References

Christian Rechberger and Hadi Soleimany and Tyge Tiessen (2018)
Cryptanalysis of Low-Data Instances of Full LowMCv2
ToSC Issue(3), 163 – 181.

Thank you