A Simple Algebraic Attack on 3-Round LowMC

Fukang Liu[1,3], Takanori Isobe[2,3], Willi Meier[4]

[1] East China Normal University, Shanghai, China
liufukangs@163.com
[2] National Institute of Information and Communications Technology, Tokyo, Japan
[3] University of Hyogo, Hyogo, Japan
takanori.isobe@ai.u-hyogo.ac.jp
[4] FHNW, Windisch, Switzerland
willimeier48@gmail.com

Abstract. With the proposal of Picnic3, it has become interesting to investigate the security of LowMC with a full S-box layer. To significantly improve the efficiency of the Picnic signature, the designers of Picnic3 recommended to use the 4-round LowMC as the underlying block cipher, which has been shown to be insecure with 2 chosen plaintexts by Liu-Isobe-Meier [6]. However, the attack scenario is very different and constrained in Picnic as the attacker is only allowed to know one single plaintext-ciphertext pair under the same key for LowMC. Recently, Banik et al. proposed guess-and-determine attacks [7] on reduced LowMC in the Picnic setting. A major finding in their attacks is that the 3-bit S-box of LowMC can be linearized by guessing a quadratic equation. Notably, the attack on 2-round LowMC with a full S-box layer can be achieved with time complexity 2^{2m} where m is the number of S-boxes in each round. As $k = 3m$, their attacks can not reach 3 rounds where k is the length of the key in bits. Although Banik et al. have improved the attacks with the meet-in-the-middle strategies [7], its memory complexity is rather high, which is $m \times 2^m$ bits of memory. In this note, we aim at low-memory key-recovery attacks as it is more fair to compare it with a pure exhaustive search. Specifically, we will describe improved algebraic attacks on 2-round LowMC by expressing the 3-bit S-box as 14 linearly independent quadratic boolean equations, which is inspired by the unsuccessful algebraic attacks on AES. As a result, the algebraic attacks on 2-round LowMC with key sizes of 129/192/255 bits can be improved by a factor of $2^4/2^{6.3}/2^{7.6}$, respectively. It seems that our attacks imply the attacks on 3-round LowMC. However, by taking the cost of gaussian elimination into account, the derived attacks on 3-round LowMC with key sizes of 192 and 255 bits are only about $2^{2.3}$ and $2^{3.7}$ times faster than the brute force. Our techniques are further applied to the instances with a partial S-box layer.

Keywords: LowMC, linearization, key recovery, algebraic attack, XL

1 Preliminaries

In this section, we will describe the notations, the specification of LowMC and costs of the exhaustive key search.

1.1 Notation

As there are several instances for both LowMC and LowMC-M, we use the following notations to describe the parameters of LowMC [2].

1. n represents the block size.
2. k represents the size of the master key.
3. m represents the number of S-boxes in each round.
4. R represents the total number of rounds.

1.2 Description of LowMC

LowMC [2] is family of SPN block ciphers proposed by Albrecht et al. in Eurocrypt 2015. Different from conventional block ciphers, the instantiation of LowMC is not fixed and each user can independently choose parameters to instantiate LowMC.

LowMC follows a common encryption procedure as most block ciphers. Specifically, it starts with a key whitening (**WK**) and then iterates a round function by R times. The round function at the $(i+1)$-th $(0 \leq i \leq R-1)$ round can be described below:

1. SBoxLayer (**SB**): A 3-bit S-box $S(x_0, x_1, x_2) = (x_0 \oplus x_1 x_2, x_0 \oplus x_1 \oplus x_0 x_2, x_0 \oplus x_1 \oplus x_2 \oplus x_0 x_1)$ will be applied on the first $3m$ bits of the state in parallel, while an identical mapping is applied on the remaining $n - 3m$ bits.
2. LinearLayer (**L**): A regular matrix $L_i \in \mathbb{F}_2^{n \times n}$ is randomly generated and multiply the n-bit state with L_i.
3. ConstantAddition (**AC**): An n-bit constant $C_i \in F_2^n$ is randomly generated and is XORed to the n-bit state.
4. KeyAddition (**AK**): A full-rank $n \times k$ binary matrix M_{i+1} is randomly generated. The n-bit round key K_{i+1} is obtained by multiplying the k-bit master key with M_{i+1}. Then, the n-bit state is XORed with K_{i+1}.

The whitening key is denoted by K_0 and it is also calculated by multiplying the master key with a random $n \times k$ binary matrix M_0.

1.3 Costs of the Exhaustive Key Search

In Picnic3, the LowMC with a full S-box layer is adopted, where $k = n = 3m$. To fairly compare our attacks with the exhaustive key search, we introduce an equivalent representation of LowMC in order to speed up the exhaustive key search. Specifically, we compute the inverse of M_0, which will be denoted by M_0^{-1}. Then, M_i $(0 \leq i \leq R)$ will be multiplied with M_0^{-1} and the obtained

matrices are denoted by E_i, i.e. $E_i = M_i \cdot M_0^{-1}$ $(0 \leq i \leq R)$. In this way, E_0 is an identity matrix. Therefore, in the exhaustive key search, the cost to compute the whitening key is eliminated. Then, to further accelerate the matrix multiplication in the linear layer and the round key addition, for each row of E_i $(i > 0)$ and L_j $(j \geq 0)$, we only record the positions whose value is "1". As E_i and L_j are all randomly generated, it is expected that in each row, the number of such positions is $3m/2$. Therefore, the cost of the matrix multiplication is reduced from $3m \times 3m$ to $4.5m^2$ bit operations. As a result, for the exhaustive key search attack on r-round LowMC, the costs can be evaluated as $r \times 2 \times 4.5m^2 = 9rm^2$ bit operations as the matrix multiplication is the most expensive part.

2 Quadratic Boolean Equations inside the S-box

Denote the 3-bit input and output of the S-box by (x_0, x_1, x_2) and (x_3, x_4, x_5), respectively. Based on the definition of the S-box, the following relations hold:

$$x_3 = x_0 \oplus x_1 x_2, \tag{1}$$
$$x_4 = x_0 \oplus x_1 \oplus x_0 x_2, \tag{2}$$
$$x_5 = x_0 \oplus x_1 \oplus x_2 \oplus x_0 x_1. \tag{3}$$

Therefore, the inverse of the S-box can be written as follows:

$$x_0 = x_3 \oplus x_4 \oplus x_4 x_5, \tag{4}$$
$$x_1 = x_4 \oplus x_3 x_5, \tag{5}$$
$$x_2 = x_3 \oplus x_4 \oplus x_5 \oplus x_3 x_4. \tag{6}$$

2.1 More Quadratic Boolean Equations

From the definition of the S-box, we can obtain 8 additional quadratic boolean equations as shown below:

$$x_3 x_1 = x_0 x_1 \oplus x_1 x_2, \tag{7}$$
$$x_3 x_2 = x_0 x_2 \oplus x_1 x_2, \tag{8}$$
$$x_4 x_0 = x_0 \oplus x_0 x_1 \oplus x_0 x_2, \tag{9}$$
$$x_4 x_2 = x_1 x_2, \tag{10}$$
$$x_5 x_0 = x_0 \oplus x_0 x_2, \tag{11}$$
$$x_5 x_1 = x_1 \oplus x_1 x_2, \tag{12}$$
$$x_3 x_0 \oplus x_0 = x_4 x_1 \oplus x_0 x_1 \oplus x_1, \tag{13}$$
$$x_4 x_1 \oplus x_0 x_1 \oplus x_1 = x_5 x_2 \oplus x_0 x_2 \oplus x_1 x_2 \oplus x_2. \tag{14}$$

Therefore, combining with the definitions of the S-box and its inverse, there will be 14 quadratic boolean equations. By treating the quadratic term as independent variables, we find that these 14 quadratic boolean equations are

indeed linearly independent by applying the gaussian elimination. We also feel interested whether it is possible to find more independent quadratic boolean equations describing the 3-bit S-box. As a result, we searched all possible quadratic equations over GF(2) in terms of 6 variables, whose time complexity is $8 \times 2^{21} = 2^{24}$. For all the obtained valid quadratic boolean equations, after the gaussian elimination is applied, it is found that there are only 14 independent quadratic equations. Consequently, 14 is the maximum number of the linearly independent quadratic boolean equations describing the 3-bit S-box of LowMC. It also proves that the above 14 quadratic equations are indeed one combination of the 14 linearly independent equations. Indeed, such an idea to write as many linearly independent quadratic equations as possible to describe an S-box has been used in the unsuccessful attacks on AES [4]. In addition, the similar idea has also been used in the XL algorithm [3] to generate more equations to solve an overdefined system of quadratic multivariate polynomial equations.

Remark. In the algebraic attack on AES [4], one input-output point, i.e. the input and output are both 0, is discarded in order to represent the 8-bit S-box of AES as a simple quadratic polynomial equation $XY = 1$, where X and Y represent the input and output of the S-box, respectively. Consequently, we are also motivated to consider whether it is possible to obtain more equations by discarding an input-output point of the 3-bit S-box of LowMC. By discarding any nonzero input-output point, i.e. the input of the discarded point satisfies $(x_0, x_1, x_2) \neq (0, 0, 0)$, we find that the number of linearly independent quadratic boolean equations is increased to 15 from 14. However, as only 1 more equation can be used and the cost of the success probability will be decreased if using 15 equations, we find that such a small increase indeed cannot improve the attacks by only using 14 probability-1 quadratic boolean equations. We are not interested to investigate the case when 2 and more points are discarded as the success probability decreases significantly.

3 Improved Algebraic Attacks on Reduced LowMC

It has been proved by Banik et al. that guessing arbitrary output bit[5] for the 3-bit S-box will make the 3 output bits linear in the 3 input bits and vice versa [7]. Indeed, even without such an observation on the S-box of LowMC, we can achieve the same time complexity with the linearization technique.

Specifically, considering the definition of the S-box, we find that if x_0 is guessed, then (x_4, x_5) can be written as linear expressions in terms of (x_1, x_2). However, the expression of x_3 is quadratic in (x_1, x_2). Therefore, we introduce a new variable to represent x_3 and treat this variable as independent of the variables representing the key bits. In this way, by guessing 1 input bit of each S-box in the first round and introducing m extra variables, the first round is linearized. Similarly, in the second round, by guessing 1 output bit of each S-box, we can obtain $3m$ linear equations in terms of $3m+m-m = 3m$ variables as

[5] The results are more general and we refer the interested readers to [7].

m key bits are already known. Therefore, the time complexity to attack 2-round LowMC is 2^{2m} times of solving $3m$ linear equations in terms of $3m$ variables. With the best time complexity of gaussian elimination using the M4RI library [1], the attack on 2-round LowMC can be evaluated as $2^{2m} \times (3m)^{2.8}$ bit operations.

3.1 Improved Attacks on 2-Round LowMC

To improve the attacks, we try to make full use of the guessed information and the 14 quadratic boolean equations describing the 3-bit S-box. Specifically, when guessing 1 input bit of the S-box, assuming that the guessed bit is x_0, we cannot write x_3 as linear expressions in terms of (x_1, x_2). However, we can derive 3 linearly independent quadratic equations rather than 1 equation in terms of (x_1, x_2, x_3), as specified below:

$$x_3 = x_0 \oplus x_1 x_2, \tag{15}$$

$$x_3 x_1 = x_0 x_1 \oplus x_1 x_2, \tag{16}$$

$$x_3 x_2 = x_0 x_2 \oplus x_1 x_2. \tag{17}$$

Similarly, if 1 output bit of the S-box is guessed, supposing the guessed bit is x_4, we cannot write x_1 as linear expressions in terms of (x_3, x_5). However, we can still derive 3 linearly independent quadratic boolean equations in terms of (x_3, x_4, x_5), as specified below:

$$x_1 = x_4 \oplus x_3 x_5, \tag{18}$$

$$x_1 x_3 = x_3 x_4 \oplus x_3 x_5, \tag{19}$$

$$x_1 x_5 = x_4 x_5 \oplus x_3 x_5. \tag{20}$$

In our improved attacks, for all the S-boxes in the first round, 1 input bit is guessed and m new variables are introduced, thus implying that the first round is fully linearized. For the second round, we only try to linearize x S-boxes by guessing the corresponding x output bits. Then, we reduce the number of variables from $3m$ to $3m - 3x$ by choosing $3m - 3x$ variables as free variables, which can be achieved by applying the gaussian elimination to the $3x$ linear equations derived from the x guessed output bits. In this way, we need to compute the solutions of the $3m - 3x$ variables. To make this phase efficient, we adopt the linearization technique once again, i.e. the $3m - 3x$ variables can form at most $\frac{(3m-3x)(3m-3x-1)}{2}$ quadratic terms and they are treated as new variables. It should be emphasized that applying the gaussian elimination to the $3x$ linear equations is still time-consuming. However, in our attacks, this cost is negligible compared with the cost to compute the solutions of the $3m - 3x$ variables with the linearization technique. The reason is that $\frac{(3m-3x)(3m-3x-1)}{2} + (3m - 3x)$ is much larger than $3m$.

From our previous description to collect quadratic equations, it can be found that the number of quadratic equations is

$$3m + 3x + 14(m - x).$$

Therefore, the following constraint should hold to efficiently compute the solution of the $3m - 3x$ variables:

$$\begin{aligned} 3m + 3x + 14(m - x) &\geq \frac{(3m - 3x)(3m - 3x - 1)}{2} + (3m - 3x) \\ &\Rightarrow 28m \geq (3m - 3x)(3m - 3x - 1) + 16x \end{aligned}$$

When m takes 43, 64, and 85, the maximal value of x is 35, 54 and 74, respectively. Therefore, the costs of our attacks on 2-round LowMC with key sizes of 129, 192 and 255 bits are $2^{43+35} \times 300^{2.8} \approx 2^{101.24}$, $2^{64+54} \times 465^{2.8} \approx 2^{142.92}$ and $2^{85+74} \times 561^{2.8} \approx 2^{184.76}$ bit operations, respectively, while the corresponding costs are $2^{105.6}$, $2^{149.28}$ and $2^{192.4}$ bit operations in Banik et al.'s attacks. Consequently, the attack on 2-round LowMC with key sizes of 129, 192 and 255 bits are improved by a factor of about 2^4, $2^{6.3}$ and $2^{7.6}$, respectively.

3.2 Attacks on 3-Round LowMC

For the attacks on 3-round LowMC, we fully linearize the first round by guessing m input bits of all the S-boxes and introduce m new variables. Then, we fully linearize the last round by guessing m output bits of the S-boxes and again introduce m new variables. For the S-boxes in the second round, only x S-boxes are linearized by guessing x bits. Then, the number of unknowns is still $3m - 3x$ as $2m$ linear equations in terms of the key bits are already guessed. However, the number of quadratic equations will be increased, which is

$$6m + 3x + 14(m - x).$$

As a result, to efficiently compute the solutions of the $3m - 3x$ variables, the following constraint should hold:

$$\begin{aligned} 6m + 3x + 14(m - x) &\geq \frac{(3m - 3x)(3m - 3x - 1)}{2} + (3m - 3x) \\ &\Rightarrow 34m \geq (3m - 3x)(3m - 3x - 1) + 16x \end{aligned}$$

When m takes 43, 64, and 85, the maximal value of x is 33, 52 and 71, respectively. Therefore, the costs of our attacks on 3-round LowMC with key sizes of 129, 192 and 255 bits are $2^{43+43+33} \times 465^{2.8} \approx 2^{143.92}$, $2^{64+64+52} \times 666^{2.8} \approx 2^{206.32}$ and $2^{85+85+71} \times 903^{2.8} \approx 2^{268.72}$ bit operations.

For 3-round LowMC, the number of bit operations can be estimated as $27m^2$. Therefore, our attacks on 3-round LowMC with key sizes of 192 and 255 bits are only about $2^{2.3}$ and $2^{3.7}$ times faster than the brute force, respectively. It should be emphasized that the designers of Rasta [5] also adopted the same way to compute the time complexity of algebraic attacks, i.e. comparing the number of bit operations.

4 Attacks on LowMC with a Partial S-box Layer

The above strategies can be trivially applied to the case where a partial S-box layer is used. Taking the ongoing LowMC competition into account, we only focus on two cases $m = 1$ and $m = 10$ with $n = k \in \{128, 192, 256\}$.

Suppose our aim is to attack r rounds of LowMC. For the first $r-1$ rounds, assume that there are t S-boxes where one input bit will be guessed and there are v S-boxes that we do not guess. In this way, there will be $3t + 14v + 14m$ quadratic boolean equations and $n-3m+t$ linear equations in terms of $t+3v+n$ variables. In addition, we have

$$t + v = m(r - 1)$$

as there are in total $m(r-1)$ S-boxes in the first $r-1$ rounds.

First, we apply the gaussian elimination to the $n - 3m + t$ linear equations and get $(t+3v+n)-(n-3m+t) = 3v+3m$ free variables. Then, to efficiently compute the solution of these $3v + 3m$ variables, it is required that

$$3t + 14v + 14m \geq (3v + 3m) + \frac{(3v + 3m) \times (3v + 3m - 1)}{2}$$

Consider the challenge, i.e. the attack on $r = \lfloor \frac{n}{m} \rfloor$ rounds of LowMC.

Attack on $(n, k, m, r) = (128, 128, 1, 128)$**.** When $n = 128$ and $m = 1$, we have $r = 128$ and hence $t + v = 127$. In this case, we have

$$3 \times (127 - v) + 14v + 14 \geq (3v + 3) + \frac{(3v + 3) \times (3v + 2)}{2}$$
$$v \leq 9$$

Therefore, the time complexity to attack 128 rounds of LowMC with $m = 1$ and $n = k = 128$ is $2^{127-9} \times 465^{2.8} \div 128^3 \approx 2^{121.9}$.

Attack on $(n, k, m, r) = (192, 192, 1, 192)$**.** When $n = 192$ and $m = 1$, we have $r = 192$ and hence $t + v = 191$. In this case, we have

$$3 \times (191 - v) + 14v + 14 \geq (3v + 3) + \frac{(3v + 3) \times (3v + 2)}{2}$$
$$\rightarrow v \leq 11$$

Therefore, the time complexity to attack 192 rounds of LowMC with $m = 1$ and $n = k = 192$ is $2^{191-11} \times 666^{2.8} \div 192^3 \approx 2^{183.5}$.

Attack on $(n, k, m, r) = (256, 256, 1, 256)$**.** When $n = 256$ and $m = 1$, we have $r = 256$ and hence $t + v = 255$. In this case, we have

$$3 \times (255 - v) + 14v + 14 \geq (3v + 3) + \frac{(3v + 3) \times (3v + 2)}{2}$$
$$\rightarrow v \leq 13$$

Therefore, the time complexity to attack 256 rounds of LowMC with $m = 1$ and $n = k = 256$ is $2^{255-13} \times 903^{2.8} \div 256^3 \approx 2^{245.5}$.

Attack on $(n, k, m, r) = (128, 128, 10, 12)$**.** When $n = 128$ and $m = 10$, we have $r = 12$ and hence $t + v = 110$. In this case, we have

$$3 \times (110 - v) + 14v + 140 \geq (3v + 30) + \frac{(3v + 30) \times (3v + 29)}{2}$$
$$\rightarrow v = 0$$

Therefore, the time complexity to attack 12 rounds of LowMC with $m = 10$ and $n = k = 128$ is $2^{110} \times 465^{2.8} \div (12 \times 128^2) \approx 2^{117.3}$.

Attack on $(n, k, m, r) = (192, 192, 10, 19)$**.** When $n = 192$ and $m = 10$, we have $r = 19$ and hence $t + v = 180$. In this case, we have

$$3 \times (180 - v) + 14v + 140 \geq (3v + 30) + \frac{(3v + 30) \times (3v + 29)}{2}$$
$$\rightarrow v \leq 2$$

Therefore, the time complexity to attack 19 rounds of LowMC with $m = 10$ and $n = k = 192$ is $2^{178} \times 666^{2.8} \div (19 \times 192^2) \approx 2^{184.9}$.

Attack on $(n, k, m, r) = (256, 256, 10, 25)$**.** When $n = 256$ and $m = 10$, we have $r = 25$ and hence $t + v = 240$. In this case, we have

$$3 \times (240 - v) + 14v + 140 \geq (3v + 30) + \frac{(3v + 30) \times (3v + 29)}{2}$$
$$\rightarrow v \leq 4$$

Therefore, the time complexity to attack 25 rounds of LowMC with $m = 10$ and $n = k = 256$ is $2^{236} \times 903^{2.8} \div (25 \times 256^2) \approx 2^{242.9}$.

Our results to solve the challenges are summarized in Table 1.

Table 1: Our results to solve the challenges

n	k	m	r	Time	Memory (in bits)
128	128	1	128	$2^{121.9}$	465×465
128	128	10	12	$2^{117.3}$	465×465
192	192	1	192	$2^{183.5}$	666×666
192	192	10	19	$2^{184.9}$	666×666
256	256	1	256	$2^{245.5}$	903×903
256	256	10	25	$2^{242.9}$	903×903
192	192	64	3	$2^{189.7}$	666×666
255	255	85	3	$2^{251.3}$	903×903

5 Conclusion

By expressing the 3-bit S-box of LowMC with 14 linearly independent quadratic boolean equations and deriving additional linearly independent quadratic boolean equations from the guessed quadratic equations, we are able to improve the key-recovery attacks on 2-round LowMC and even achieve faster key-recovery attacks on 3-round LowMC, though the advantage over the brute force is limited. A future research may be to compare our simple attacks with the Gröbner basis attacks or SAT-based attacks. We believe our way to deduce more equations is also very meaningful to Gröbner basis attacks since the time complexity to solve an overdefined system of multivariate nonlinear equations will decrease as the number of equations increases. However, it seems that new techniques are essential in order to break full-round (4-round) LowMC in the Picnic setting.

References

1. Martin Albrecht and Gregory Bard. *The M4RI Library*. The M4RI Team. `http://m4ri.sagemath.org`.
2. Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454. Springer, 2015.
3. Nicolas T. Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 392–407. Springer, 2000.
4. Nicolas T. Courtois and Josef Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*, pages 267–287. Springer, 2002.
5. Christoph Dobraunig, Maria Eichlseder, Lorenzo Grassi, Virginie Lallemand, Gregor Leander, Eik List, Florian Mendel, and Christian Rechberger. Rasta: A cipher with low anddepth and few ands per bit. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 662–692. Springer, 2018.
6. Fukang Liu, Takanori Isobe, and Willi Meier. Cryptanalysis of full LowMC and LowMC-M with algebraic techniques. Cryptology ePrint Archive, Report 2020/1034, 2020. `https://eprint.iacr.org/2020/1034`.
7. Banik Subhadeep, Barooti Khashayar, and Vaudenay Serge. Lowmc cryptanalysis challenge: 2nd round, 2021. `https://raw.githubusercontent.com/`

lowmcchallenge/lowmcchallenge-material/master/docs/lowmc_analysis_2.pdf.