

Improved Cryptanalysis of Polar Bear

Mahdi M. Hasanzadeh Elham Shakour Shahram Khazaei

Zaeim Electronic Industries Company, P.O. BOX 14155-1434, Tehran, Iran
{Hasanzadeh, shakour, Khazaei}@zaeim.com

Abstract. In this paper we propose a Guess-and-Determine based initial state recovery attack on Polar Bear, one of the ECRYPT stream cipher project candidates. The computational complexity and success probability of our attack are $O(2^{31})$ and $2^{-26.4}$ respectively. Our attack can be considered as one with computational complexity of $O(2^{57.4})$ which is much better than the attack recently proposed by J. Mattsson with computational complexity of $O(2^{79})$.

Keywords. Stream Cipher, Guess-and-Determine Attack, Polar Bear, ECRYPT, Security Evaluation.

1 Introduction

Stream ciphers are widely used for fast encryption of sensitive data. Lots of old stream ciphers that have been formerly used can no longer be considered secure, because of their vulnerability to newly developed cryptanalysis techniques. In particular, the NESSIE project [4] did not select any of the proposed stream ciphers for its portfolio, as it was felt that none of the submissions was sufficiently strong. In order to create a portfolio of secure stream ciphers, the ECRYPT project [1] made a call for designs of new stream ciphers which led to submission of 35 proposals to the project by April 2005.

Polar Bear [2] is one of the ECRYPT stream cipher project candidates. The cipher was designed for software applications and dealing with keys of up to 128 bits length. John Mattsson recently found a weakness on the cipher which lead to an initial state recovery attack on it with computational complexity of $O(2^{79})$ according to his note [3]. The details of this attack have not been published yet. In this paper we improve Mattsson's results and propose an attack with computational complexity of $O(2^{57.4})$. Our Analysis is a Guess-and-Determine based initial state recovery attack whose computational complexity and success probability are $O(2^{31})$ and $2^{-26.4}$ respectively which can also be considered as one with computational complexity of $O(2^{57.4})$.

The paper is organized as follows. In Section 2 a brief description of the key-stream generator of Polar Bear is given. The details of our attack are presented in Section 3 and, finally, the paper is concluded in Section 4.

2 Outline of Polar Bear

Polar Bear [2] works with 16-bit words and uses a 7-word LFSR R^0 and a 9-word LFSR R^1 . These are viewed as acting over $GF(2^{16})$. Besides these registers, the internal state of the cipher also depends on a word quantity, S , and a dynamic permutation of bytes, D_8 . The cipher deals with

keys of up to 128 bits length. The IV can be any number of bytes up to a maximum of 31. The initial states of R^0 and R^1 are determined thorough a certain key-IV set up, D_8 is initialized to the table T_8 , the Rijndael S-box, and S is set to zero. The cipher produces two words at each cycle of operation. At each cycle, firstly, the two LFSRs are irregularly clocked according to S . Then, two words from each of R^0 and R^1 are selected and nonlinearly filtered using the permutation D_8 to produce two output words. Afterwards, some selected entries in D_8 are swapped. Finally, S and one word of R^0 are modified in preparation for the next cycle.

Let \parallel denotes concatenation of 16-bit words as well as 8-bit bytes. Moreover, let \oplus and $+_{16}$ respectively denote bitwise XOR and addition modulo 2^{16} of 16-bit words. A complete description of Polar Bear can be given by the following pseudo-code.

1. Using the initialization process, determine the values of $(R_6^0, R_5^0, R_4^0, R_3^0, R_2^0, R_1^0, R_0^0)$ and $(R_8^1, R_7^1, R_6^1, R_5^1, R_4^1, R_3^1, R_2^1, R_1^1, R_0^1)$.
2. $S \leftarrow 0, D_8 \leftarrow T_8$.
3. For $t = 1$ to $N/2$ do (N is the required number of output words):
 - 3.1. $b_0 \leftarrow 2 + (\lfloor S / 2^{14} \rfloor \bmod 2), b_1 \leftarrow 2 + (\lfloor S / 2^{15} \rfloor \bmod 2)$.
 - 3.2. Clock R^0 and R^1 LFSRs b_0 and b_1 times, respectively.
 - 3.3. $\alpha_0^0 \parallel \alpha_1^0 \parallel \alpha_2^0 \parallel \alpha_3^0 \leftarrow R_6^0 \parallel R_5^0$.
 - 3.4. $\beta_0^0 \parallel \beta_1^0 \parallel \beta_2^0 \parallel \beta_3^0 \leftarrow D_8(\alpha_0^0) \parallel D_8(\alpha_1^0) \parallel D_8(\alpha_2^0) \parallel D_8(\alpha_3^0)$.
 - 3.5. $(D_8(\alpha_0^0), D_8(\alpha_1^0), D_8(\alpha_2^0), D_8(\alpha_3^0)) \leftarrow (\beta_2^0, \beta_0^0, \beta_3^0, \beta_1^0)$.
 - 3.6. $\alpha_0^1 \parallel \alpha_1^1 \parallel \alpha_2^1 \parallel \alpha_3^1 \leftarrow R_8^1 \parallel R_7^1$.
 - 3.7. $\beta_0^1 \parallel \beta_1^1 \parallel \beta_2^1 \parallel \beta_3^1 \leftarrow D_8(\alpha_0^1) \parallel D_8(\alpha_1^1) \parallel D_8(\alpha_2^1) \parallel D_8(\alpha_3^1)$.
 - 3.8. $(D_8(\alpha_0^1), D_8(\alpha_1^1), D_8(\alpha_2^1), D_8(\alpha_3^1)) \leftarrow (\beta_2^1, \beta_0^1, \beta_3^1, \beta_1^1)$.
 - 3.9. $\gamma_0^1 \parallel \gamma_1^1 \leftarrow \beta_0^1 \parallel \beta_1^1 \parallel \beta_2^1 \parallel \beta_3^1$.
 - 3.10. $\gamma_0^0 \parallel \gamma_1^0 \leftarrow \beta_0^0 \parallel \beta_1^0 \parallel \beta_2^0 \parallel \beta_3^0$.
 - 3.11. $S \leftarrow S +_{16} \gamma_0^1$.
 - 3.12. $R_5^0 \leftarrow R_5^0 +_{16} \gamma_1^1$.
 - 3.13. $Z_t^0 \leftarrow \gamma_0^1 \oplus \gamma_0^0, Z_t^1 \leftarrow \gamma_1^1 \oplus \gamma_1^0$.

The sequence $\{Z_0^1, Z_1^1, Z_0^2, Z_1^2, \dots, Z_0^{N/2}, Z_1^{N/2}\}$ is the output sequence of the cipher. The feedback polynomials of the registers are primitive over $\text{GF}(2^{16})$ and given by $\mu^0 x^7 + \theta^0 x^6 - 1 = 0$ and $\mu^1 x^9 + \theta^1 x^4 - 1 = 0$ in accordance with the recursive equations $R_{n+7}^0 = \theta^0 \cdot R_{n+1}^0 + \mu^0 \cdot R_n^0$ and $R_{n+9}^1 = \theta^1 \cdot R_{n+5}^1 + \mu^1 \cdot R_n^1$ for the output sequences of R^0 and R^1 LFSRs, respectively. Here ‘+’ and ‘ \cdot ’ respectively denote addition and multiplication operations of the finite field $\text{GF}(2^{16})$. Refer to [2] for more details on the cipher and definition of the finite field $\text{GF}(2^{16})$. In the rest of this paper we drop the multiplication operation symbol for simplicity.

3 Description of the Attack

In this section we present our attack on Polar Bear which is an improvement of one recently proposed by John Mattsson [3]. Both attacks use known plain-text scenario and recover the initial states of registers.

Mattsson's attack recovers the initial states of the registers under the assumption that in the first six cycles both registers are clocked two steps and all the values of α_j^i 's, totally 48 values, are different. Under these conditions D_8 is known and is equal to T_8 for those entries used in the first six cycles.

Let $(R_{n+8}^1, R_{n+7}^1, R_{n+6}^1, R_{n+5}^1, R_{n+4}^1, R_{n+3}^1, R_{n+2}^1, R_{n+1}^1, R_n^1)$ be the state of LFSR R^1 after n steps. Mattsson guesses the 64 bits $R_9^1, R_{10}^1, R_{11}^1$ and R_{13}^1 to recover the unknown initial states of the registers in a Guess-and-Determine manner. According to Mattsson's notes [3], the time complexity of his attack is $O(2^{79})$.

Our attack recovers the initial states of the registers under the assumption that in the first eight cycles, R^0 is clocked two steps in all cycles, the sequence of number of steps for R^1 is $\{2, 3, 3, 3, 3, 2, 3, 2\}$, and all the values of α_j^i 's, totally 64 values, are different. Under these conditions D_8 is known and is equal to T_8 for those entries used in the first eight cycles. Note since S is initialized to zero the two registers are clocked twice in the first cycle. Therefore, the probability of validity of the assumed sequences for the number of steps for the registers in the first eight cycles is equal to 2^{-14} . The probability that all the 64 values of α_j^i 's in the first eight cycles are different is equal to $256 \times 255 \times \dots \times 193 / 256^{64} \approx 2^{-12.4}$. Our attack is a Guess-and-Determine based attack which first guesses the values of R_{18}^1 and R_{19}^1 and then recovers the initial states of the registers with a little effort. The total number of possible values for R_{18}^1 and R_{19}^1 is equal to 2^{31} (see the remark at the end of this section). Therefore, the computational complexity and success probability of our attack are $O(2^{31})$ and $2^{-26.4}$ respectively. One can interpret the attack as one with computational complexity of $O(2^{57.4})$.

Let $(R_{n+8}^1, R_{n+7}^1, R_{n+6}^1, R_{n+5}^1, R_{n+4}^1, R_{n+3}^1, R_{n+2}^1, R_{n+1}^1, R_n^1)$ be the state of the LFSR R^1 after n steps. We denote the state of the register R^0 after n steps by $(R_{n+6}^0, R_{n+5}^0, R_{n+4}^0, R_{n+3}^0, R_{n+2}^0, R_{n+1}^0, R_n^0)$ where R_{n+j}^0 ($0 \leq j \leq 6$) may have a hat and is replaced by \hat{R}_{n+j}^0 . We use a hat for R_{n+j}^0 if it is a shifted value of the cell number five of the register R^0 and its value has been nonlinearly updated through the step 3.12 of the pseudo-code. For example, since the registers are clocked twice at the first cycle, the state of the register R^0 will be $(R_8^0, \hat{R}_7^0, R_6^0, R_5^0, R_4^0, R_3^0, R_2^0)$ after the first cycle. After the second cycle, the state of R^0 will be $(R_{10}^0, \hat{R}_9^0, R_8^0, \hat{R}_7^0, R_6^0, R_5^0, R_4^0)$ or $(R_{11}^0, \hat{R}_{10}^0, R_9^0, R_8^0, \hat{R}_7^0, R_6^0, R_5^0)$ if the register R^0 is respectively clocked two or three steps at the second cycle. And so on.

The 8 by 8 S-box T_8 acts on 8-bit bytes. For our convenience we define a 16 by 16 S-box T which acts on 16-bit words by applying T_8 on the two bytes of its input word. To be more precise, if w_1 and w_0 are two arbitrary 8-bit bytes, we have $T(w_0 || w_1) = T_8(w_0) || T_8(w_1)$. Using this definition together with the introduced notations for the instantaneous internal state of R^0 and R^1 , and taking

into account the assumed clocking way of the registers and the difference assumption of α_j^i 's at the first eight cycles of cipher operation, one can easily trace the relations between different parts of the cipher and derive the relations between the internal state variables as well as the relations of output sequence of the cipher. We have derived and summarized these relations in the Table 1. We have not mentioned the relation for swapping the D_8 entries and updating of S .

Table 1. Internal and output relations of the first eight cycles of cipher operation under our assumptions.

| Cycle | R^0 Relations | R^1 Relations | Output Relations | R^0 Nonlinear Update |
|-------|--|---|--|--|
| 1 | (1) $R_7^0 = \theta^0 R_1^0 + \mu^0 R_0^0$ (2) $R_8^0 = \theta^0 R_2^0 + \mu^0 R_1^0$ | (3) $R_9^1 = \theta^1 R_5^1 + \mu^1 R_0^1$ (4) $R_{10}^1 = \theta^1 R_6^1 + \mu^1 R_1^1$ | (5) $T(R_7^0) \oplus T(R_9^1) = Z_1^1$ (6) $T(R_8^0) \oplus T(R_{10}^1) = Z_0^1$ | (7) $\hat{R}_7^0 = R_7^0 +_{16} T(R_9^1)$ |
| 2 | (1) $R_9^0 = \theta^0 R_3^0 + \mu^0 R_2^0$ (2) $R_{10}^0 = \theta^0 R_4^0 + \mu^0 R_3^0$ | (3) $R_{11}^1 = \theta^1 R_7^1 + \mu^1 R_2^1$ (4) $R_{12}^1 = \theta^1 R_8^1 + \mu^1 R_3^1$ (5) $R_{13}^1 = \theta^1 R_9^1 + \mu^1 R_4^1$ | (6) $T(R_9^0) \oplus T(R_{12}^1) = Z_1^2$ (7) $T(R_{10}^0) \oplus T(R_{13}^1) = Z_0^2$ | (8) $\hat{R}_9^0 = R_9^0 +_{16} T(R_{12}^1)$ |
| 3 | (1) $R_{11}^0 = \theta^0 R_5^0 + \mu^0 R_4^0$ (2) $R_{12}^0 = \theta^0 R_6^0 + \mu^0 R_5^0$ | (3) $R_{14}^1 = \theta^1 R_{10}^1 + \mu^1 R_5^1$ (4) $R_{15}^1 = \theta^1 R_{11}^1 + \mu^1 R_6^1$ (5) $R_{16}^1 = \theta^1 R_{12}^1 + \mu^1 R_7^1$ | (6) $T(R_{11}^0) \oplus T(R_{15}^1) = Z_1^3$ (7) $T(R_{12}^0) \oplus T(R_{16}^1) = Z_0^3$ | (8) $\hat{R}_{11}^0 = R_{11}^0 +_{16} T(R_{15}^1)$ |
| 4 | (1) $R_{13}^0 = \theta^0 \hat{R}_7^0 + \mu^0 R_6^0$ (2) $R_{14}^0 = \theta^0 R_8^0 + \mu^0 \hat{R}_7^0$ | (3) $R_{17}^1 = \theta^1 R_{13}^1 + \mu^1 R_8^1$ (4) $R_{18}^1 = \theta^1 R_{14}^1 + \mu^1 R_9^1$ (5) $R_{19}^1 = \theta^1 R_{15}^1 + \mu^1 R_{10}^1$ | (6) $T(R_{13}^0) \oplus T(R_{18}^1) = Z_1^4$ (7) $T(R_{14}^0) \oplus T(R_{19}^1) = Z_0^4$ | (8) $\hat{R}_{13}^0 = R_{13}^0 +_{16} T(R_{18}^1)$ |
| 5 | (1) $R_{15}^0 = \theta^0 \hat{R}_9^0 + \mu^0 R_8^0$ (2) $R_{16}^0 = \theta^0 R_{10}^0 + \mu^0 \hat{R}_9^0$ | (3) $R_{20}^1 = \theta^1 R_{16}^1 + \mu^1 R_{11}^1$ (4) $R_{21}^1 = \theta^1 R_{17}^1 + \mu^1 R_{12}^1$ (5) $R_{22}^1 = \theta^1 R_{18}^1 + \mu^1 R_{13}^1$ | (6) $T(R_{15}^0) \oplus T(R_{21}^1) = Z_1^5$ (7) $T(R_{16}^0) \oplus T(R_{22}^1) = Z_0^5$ | (8) $\hat{R}_{15}^0 = R_{15}^0 +_{16} T(R_{21}^1)$ |
| 6 | (1) $R_{17}^0 = \theta^0 \hat{R}_{11}^0 + \mu^0 R_{10}^0$ (2) $R_{18}^0 = \theta^0 R_{12}^0 + \mu^0 \hat{R}_{11}^0$ | (3) $R_{23}^1 = \theta^1 R_{19}^1 + \mu^1 R_{14}^1$ (4) $R_{24}^1 = \theta^1 R_{20}^1 + \mu^1 R_{15}^1$ | (5) $T(R_{17}^0) \oplus T(R_{23}^1) = Z_1^6$ (6) $T(R_{18}^0) \oplus T(R_{24}^1) = Z_0^6$ | (7) $\hat{R}_{17}^0 = R_{17}^0 +_{16} T(R_{23}^1)$ |
| 7 | (1) $R_{19}^0 = \theta^0 \hat{R}_{13}^0 + \mu^0 R_{12}^0$ (2) $R_{20}^0 = \theta^0 R_{14}^0 + \mu^0 \hat{R}_{13}^0$ | (3) $R_{25}^1 = \theta^1 R_{21}^1 + \mu^1 R_{16}^1$ (4) $R_{26}^1 = \theta^1 R_{22}^1 + \mu^1 R_{17}^1$ (5) $R_{27}^1 = \theta^1 R_{23}^1 + \mu^1 R_{18}^1$ | (6) $T(R_{19}^0) \oplus T(R_{26}^1) = Z_1^7$ (7) $T(R_{20}^0) \oplus T(R_{27}^1) = Z_0^7$ | (8) $\hat{R}_{19}^0 = R_{19}^0 +_{16} T(R_{26}^1)$ |
| 8 | (1) $R_{21}^0 = \theta^0 \hat{R}_{15}^0 + \mu^0 R_{14}^0$ (2) $R_{22}^0 = \theta^0 R_{16}^0 + \mu^0 \hat{R}_{15}^0$ | (3) $R_{28}^1 = \theta^1 R_{24}^1 + \mu^1 R_{19}^1$ (4) $R_{29}^1 = \theta^1 R_{25}^1 + \mu^1 R_{20}^1$ | (5) $T(R_{21}^0) \oplus T(R_{28}^1) = Z_1^8$ (6) $T(R_{22}^0) \oplus T(R_{29}^1) = Z_0^8$ | (7) $\hat{R}_{21}^0 = R_{21}^0 +_{16} T(R_{28}^1)$ |

All the relations of Table 1 are invertible in all the input variables. In other words, if we know all the input variables except one for each equation, the unknown variable is uniquely determined. Such kinds of equations are suitable to be solved in a Guess-and-Determine manner. In a Guess-

and-Determine attack, we first guess some variables and then try to recover the reminder variables efficiently. The less the space size of the guessed variables is, the less the computational complexity is required. The validity of a guess is determined using some additional check equations.

It is easy to show that it is not possible to uniquely solve the system of equations of Table 1 by guessing less than two variables. Moreover, guessing the values of R_{18}^1 and R_{19}^1 reveals the initial state of the registers, that is $(R_6^0, R_5^0, R_4^0, R_3^0, R_2^0, R_1^0, R_0^0)$ and $(R_8^1, R_7^1, R_6^1, R_5^1, R_4^1, R_3^1, R_2^1, R_1^1, R_0^1)$ which are our desires. We have summarized the steps which lead to recovering the initial states of the registers in Table 2.

Each step of Table 2 states that which equation from Table 1 must be used to determine one of the variables using previously determined variables. For example, at 20th step the variable R_{10}^0 is determined using equation 1 at cycle 6 of the Table 1 because R_{17}^0 and \hat{R}_{11}^0 have already been determined at the 7th and 19th steps respectively. More precisely we have $R_{10}^0 = (R_{17}^0 - \theta^0 \hat{R}_{11}^0) / \mu^0$ where $-$ and $/$ are the subtraction and division operations of the finite field $\text{GF}(2^{16})$.

The correct initial state can be find by running the cipher some cycles and comparing the resulting output sequence with the given key-stream sequence.

Remark on the total number of possible values for R_{18}^1 : Although R_{18}^1 is an 16-bit word, under the assumed clocking way for the registers, there are only 2^{15} possibilities for it. Indeed, let S_3 and S_4 be the values of S at the end of 3rd and 4th cycles. We have $S_4 = S_3 +_{16} T(R_{18}^1)$. Since we have assumed that R^1 and R^0 have respectively clocked three times and twice at both the 3rd and the 4th cycles, the two most significant bits of both S_3 and S_4 are 10. This proves the two most significant bits of $T(R_{18}^1)$ can be either 00 or 11 which shows the existence of 2^{15} possible choice for R_{18}^1 .

4 Conclusion

In this paper we proposed a Guess-and-Determine based initial state recovery attack whose computational complexity and success probability are $O(2^{31})$ and $2^{-26.4}$ respectively. Our attack can be considered as one with computational complexity of $O(2^{57.4})$ which is much better than one recently proposed by Mattsson with computational complexity of $O(2^{79})$. The weakness, which enables these attacks, can effectively be countered by initializing the dynamic permutation D_8 to an 8 by 8 key-IV dependent S-box provided that it seems random to an attacker.

References

1. eSTREAM, the ECRYPT Stream Cipher Project (2005) <http://www.ecrypt.eu.org/stream/>.
2. Håstad J. and Näslund M., The Stream Cipher Polar Bear. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/021 (2005) <http://www.ecrypt.eu.org/stream/>.
3. Mattsson J., Weakness in Polar Bear. eSTREAM, ECRYPT Stream Cipher Project, Discussion Forum (2005) <http://www.ecrypt.eu.org/stream/phorum/read.php?1,219/>.
4. NESSIE: New European Schemes for Signature, Integrity and Encryption, <http://www.nessie.eu.org/nessie/>.

Table 2. The details of the procedure of recovering the initial state of the registers, by guessing R_{18}^1 and R_{19}^1 .

| Step | Known Words | (Cycle-Relation) | Deduced Word | Step | Known Words | (Cycle-Relation) | Deduced Word |
|------|----------------------------|------------------|------------------|------|----------------------------|------------------|--------------|
| 1 | R_{18}^1 | (4-6) | R_{13}^0 | 29 | \hat{R}_{15}^0, R_{16}^0 | (8-2) | R_{22}^0 |
| 2 | R_{13}^0, R_{18}^1 | (4-8) | \hat{R}_{13}^0 | 30 | R_{22}^0 | (8-6) | R_{29}^1 |
| 3 | R_{19}^1 | (4-7) | R_{14}^0 | 31 | \hat{R}_{15}^0, R_{14}^0 | (8-1) | R_{21}^0 |
| 4 | \hat{R}_{13}^0, R_{14}^0 | (7-2) | R_{20}^0 | 32 | R_{21}^0 | (8-5) | R_{28}^1 |
| 5 | R_{20}^0 | (7-7) | R_{27}^1 | 33 | R_{28}^1, R_{19}^1 | (8-3) | R_{24}^1 |
| 6 | R_{27}^1, R_{18}^1 | (7-5) | R_{23}^1 | 34 | R_{24}^1, R_{15}^1 | (6-4) | R_{20}^1 |
| 7 | R_{23}^1 | (6-5) | R_{17}^0 | 35 | R_{20}^1, R_{29}^1 | (8-4) | R_{25}^1 |
| 8 | R_{23}^1, R_{19}^1 | (6-3) | R_{14}^1 | 36 | R_{25}^1, R_{21}^1 | (7-3) | R_{16}^1 |
| 9 | R_{14}^1, R_{18}^1 | (4-4) | R_9^1 | 37 | R_{16}^1, R_{20}^1 | (5-3) | R_{11}^1 |
| 10 | R_9^1 | (1-5) | R_7^0 | 38 | R_{16}^1 | (3-7) | R_{12}^0 |
| 11 | R_7^0, R_9^1 | (1-7) | \hat{R}_7^0 | 39 | R_{12}^0, R_6^0 | (3-2) | R_5^0 |
| 12 | \hat{R}_7^0, R_{14}^0 | (4-2) | R_8^0 | 40 | R_5^0, R_{11}^0 | (3-1) | R_4^0 |
| 13 | \hat{R}_7^0, R_{13}^0 | (4-1) | R_6^0 | 41 | R_{10}^0, R_4^0 | (2-2) | R_3^0 |
| 14 | R_8^0 | (1-6) | R_{10}^1 | 42 | \hat{R}_{13}^0, R_{12}^0 | (7-1) | R_{19}^0 |
| 15 | R_{10}^1, R_{14}^1 | (3-3) | R_5^1 | 43 | R_{19}^0 | (7-6) | R_{26}^1 |
| 16 | R_5^1, R_9^1 | (1-3) | R_1^0 | 44 | R_{26}^1, R_{22}^1 | (7-4) | R_{17}^1 |
| 17 | R_{10}^1, R_{19}^1 | (4-5) | R_{15}^1 | 45 | R_{17}^1, R_{21}^1 | (5-4) | R_{12}^1 |
| 18 | R_{15}^1 | (3-6) | R_{11}^0 | 46 | R_{12}^1 | (2-6) | R_9^0 |
| 19 | R_{11}^0, R_{15}^1 | (3-8) | \hat{R}_{11}^0 | 47 | R_9^0, R_3^0 | (2-1) | R_2^0 |
| 20 | R_{17}^0, \hat{R}_{11}^0 | (6-1) | R_{10}^0 | 48 | R_1^0, R_7^0 | (1-1) | R_0^0 |
| 21 | R_{10}^0 | (2-7) | R_{13}^1 | 49 | R_{17}^1, R_{13}^1 | (4-3) | R_8^1 |
| 22 | R_{13}^1, R_9^1 | (2-5) | R_4^1 | 50 | R_{16}^1, R_{12}^1 | (3-5) | R_7^1 |
| 23 | R_{13}^1, R_{18}^1 | (5-5) | R_{22}^1 | 51 | R_{15}^1, R_{11}^1 | (3-4) | R_6^1 |
| 24 | R_{22}^1 | (5-7) | R_{16}^0 | 52 | R_{12}^1, R_8^1 | (2-4) | R_3^1 |
| 25 | R_{16}^0, R_{10}^0 | (5-2) | \hat{R}_9^0 | 53 | R_{11}^1, R_7^1 | (2-3) | R_2^1 |
| 26 | \hat{R}_9^0, R_8^0 | (5-1) | R_{15}^0 | 54 | R_{10}^1, R_6^1 | (1-4) | R_1^1 |
| 27 | R_{15}^0 | (5-6) | R_{21}^1 | 55 | R_9^1, R_5^1 | (1-3) | R_0^1 |
| 28 | R_{15}^0, R_{21}^1 | (5-8) | \hat{R}_{15}^0 | | | | |