Meet-in-the-Middle Attacks on Reduced-Round Midori64

Li Lin, Wenling Wu

Trusted Computing and Information Assurance Laboratory,
Institute of Software,
Chinese Academy of Sciences

FSE 2017
March 8, 2017

Outline

Outline

1. Preliminaries
   - Description of Midori64
   - Definitions and Propositions
   - Reviews of Former Works
   - Basic Attack Model
   - Key Relations to Improve the Complexity
2. Meet-in-the-Middle Attack on 10-Round Midori64
3. Meet-in-the-Middle Attack on 11-Round Midori64
4. Meet-in-the-Middle Attack on 12-Round Midori64
5. Conclusions

Description of Midori64

- In the past few years, lightweight cryptography has become a popular research discipline with a number of ciphers and hash functions proposed;
- The goals of these ciphers range from minimizing the hardware area (e.g., PRESENT, LED, LBlock) to low latency (e.g., PRINCE);
- However, the optimization goal of low energy for block cipher design has not attached much attention.

Description of Midori64

- Midori is a lightweight block cipher designed by Banik et al. at ASIACRYPT 2015;
- The goal of Midori is optimized with respect to the energy consumed by the circuit per bit in encryption or decryption operation;
- Two versions of Midori:
  - Midori64: 64-bit block length, 128-bit key length;
  - Midori128: 128-bit block length, 128-bit key length;

Description of Midori64

- Midori is a lightweight block cipher designed by Banik et al. at ASIACRYPT 2015;
- The goal of Midori is optimized with respect to the energy consumed by the circuit per bit in encryption or decryption operation;
- Two versions of Midori:
  - Midori64: 64-bit block length, 128-bit key length;
  - Midori128: 128-bit block length, 128-bit key length;
- 64-bit data expression (the size of s_i is 4-bit):

$$S = \begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix}$$

Description of Midori64

- A Midori64 round applies the following four operations:
  - **SubCell:** $s_i \leftarrow S(s_i)$;
  - **ShuffleCell:** Each nibble of the state is permuted as follows:

$$\begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix} \leftarrow \begin{pmatrix} s_0 & s_{14} & s_9 & s_7 \\ s_{10} & s_4 & s_3 & s_{13} \\ s_5 & s_{11} & s_{12} & s_2 \\ s_{15} & s_1 & s_6 & s_8 \end{pmatrix}$$

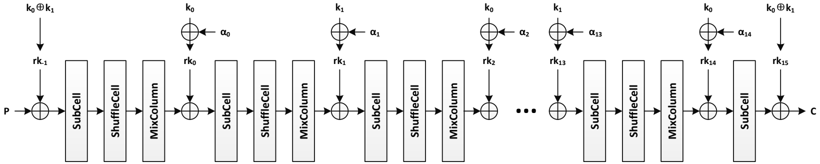
  - **MixColumn:**

$$\begin{pmatrix} s_i \\ s_{i+1} \\ s_{i+2} \\ s_{i+3} \end{pmatrix} \leftarrow \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} s_i \\ s_{i+1} \\ s_{i+2} \\ s_{i+3} \end{pmatrix}$$

  - **KeyAdd:** $S \leftarrow S \oplus rk_i$

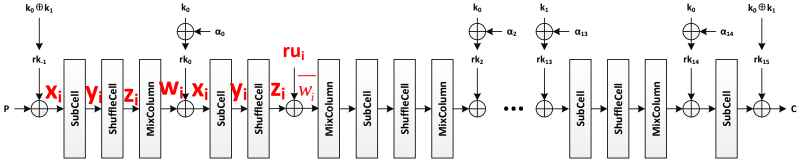
Description of Midori64

- **Key-schedule:** A 128-bit secret key K is denoted as two 64-bit keys k_0 and k_1, the whiten key and the last sub-key are $k_0 \oplus k_1$, and the sub-key for round i is $rk_i = k_{(i \ mod \ 2)} \oplus \alpha_i$, where $0 \leq i \leq R-2$ and α_i is a constant.
- The total round number of Midori64 is 16



Description of Midori64

- **Key-schedule:** A 128-bit secret key K is denoted as two 64-bit keys k_0 and k_1, the whiten key and the last sub-key are $k_0 \oplus k_1$, and the sub-key for round i is $rk_i = k_{(i \ mod \ 2)} \oplus \alpha_i$, where $0 \leq i \leq R-2$ and α_i is a constant.
- The total round number of Midori64 is 16



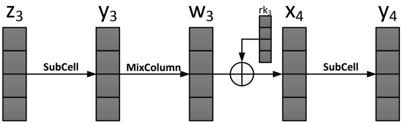
Definitions and Propositions

Definition 1 (2-δ-set)

Let a 2-δ-set be a set of $2^{2\times4}$ states that are all different in two state nibbles (active nibbles) and all equal in the other state nibbles (inactive nibbles).

Definition 2 (Super-box)

For each value of one column of rk_3, a Midori Super-box maps one column of z_3 to one column of y_4. It consists of one SubCell, one MixColumn, one KeyAdd and one SubCell.



Definitions and Propositions

Proposition 1 (Differential Property of S-box)

Given Δ_i and Δ_0 two non-zero differences, the equation of S-box

$$S(x) \oplus S(x \oplus \Delta_i) = \Delta_0, \tag{1}$$

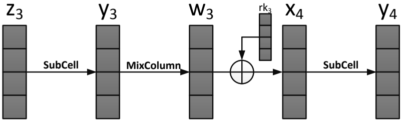
has one solution in average.

Proposition 2 (Differential Property of Super-box)

Given Δ_i and Δ_0 two non-zero differences in $F_{2^{16}}$, the equation of Super-box

$$Super - S(x) \oplus Super - S(x \oplus \Delta_i) = \Delta_0, \tag{2}$$

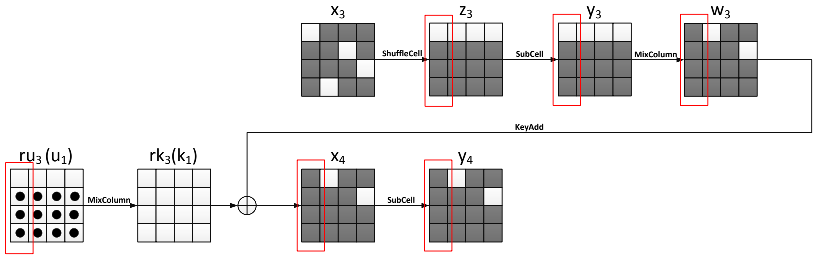
has one solution in average for each key value.



Definitions and Propositions

Proposition 3 (Partial Differential Property of Super-box)

As shown in the figure below, if the first column of z_3 is active only in the last 3 nibbles, $z_3[1,2,3]||y_4[0,1,2,3]$ has one solution in average for each $\Delta z_3[1,2,3]||\Delta y_4[0,1,2,3]||ru_3[1,2,3]$.

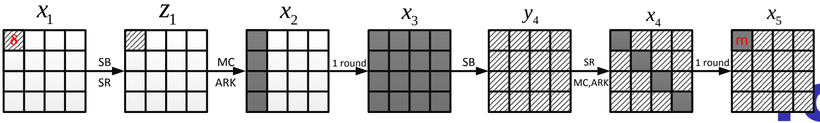


Reviews of Former Works

- **Demirci and Selçuk distinguisher.** The distinguishers of Demirci and Selçuk attacks are based on the proposition below:

Proposition 4

Consider the encryption of a δ-set through four full AES rounds. For each of the 16 bytes of the state, the ordered sequence of 256 values of that byte in the corresponding ciphertexts is fully determined by just 25 byte-parameters. Consequently, for any fixed byte position, there are at most 2^{200} possible sequences when we consider all the possible choices of keys and δ-sets.

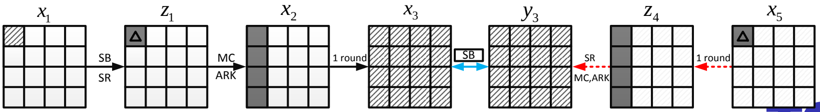


Reviews of Former Works

- **Dunkelman et al. distinguisher.** Dunkelman et al. proposed multiset, key-bridging technique and differential enumeration technique to improve Demirci and Selçuk's attack.
- **Derbez et al. distinguisher.** Derbez et al. proposed efficient tabulation to improve differential enumeration technique.

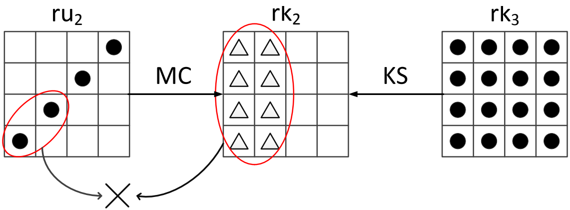
Proposition 5 (Differential Enumeration Technique with Efficient Tabulation)

If a message of δ-set belongs to a pair conforming to the 4-round truncated differential trail below, the values of multiset are only determined by 10 byte-parameters of intermediate states $\Delta z_1[0]||x_2[0,1,2,3]||\Delta x_5[0]||z_4[0,1,2,3]$ *presented as gray cells.*



Reviews of Former Works

- **Li et al. distinguisher.** Li et al. introduced the key-dependent sieve technique. The precomputation procedure allows to deduce $ru_2[3,6,9,12]$ and rk_3, independently. Meanwhile, by the key-schedule of AES-192, $rk_3 \Rightarrow$ Column 0 and Column 1 of rk_2. This means that the value of the equivalent sub-key $ru_2[3,6]$ can be deduced from rk_3. The probability of this is 2^{-16}.



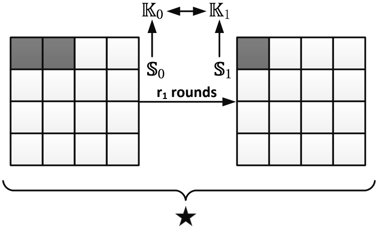
Basic Attack Model

- Precomputation phase:

- Online phase:

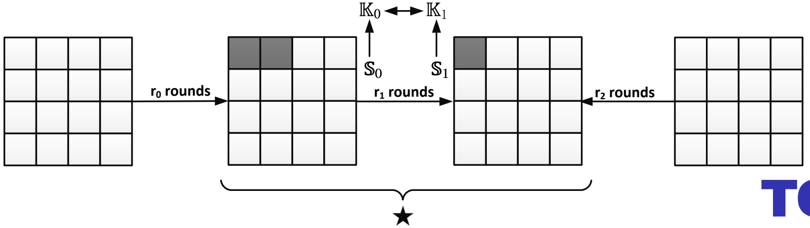
Basic Attack Model

- Precomputation phase:
  - In the precomputation phase, we build a lookup table T containing all the possible sequences constructed from a 2-δ-set such that one message verifies Proposition 5.
- Online phase:



Basic Attack Model

- Precomputation phase:
  - In the precomputation phase, we build a lookup table T containing all the possible sequences constructed from a 2-δ-set such that one message verifies Proposition 5.
- Online phase:
  - Using a large number of plaintexts and ciphertexts, and expecting that for each key candidate, there is one pair of plaintexts satisfying the truncated differential trail, use this pair of plaintexts to build a 2-δ-set;
  - Finally, we partially decrypt the associated 2-δ-set through the last r_2 rounds and check whether it belongs to T.

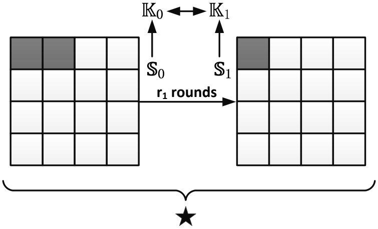


Key Relations to Improve the Complexity

- By the key-schedule, a lot of key relations can be found.
- These key relations cannot improve the complexity directly.

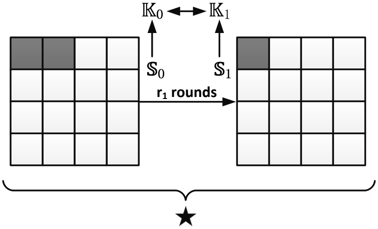
Key Relations to Improve the Complexity

- By the key-schedule, a lot of key relations can be found.
- These key relations cannot improve the complexity directly.
- $\mathbb{S}_0 \Rightarrow \mathbb{K}_0$, $\mathbb{S}_1 \Rightarrow \mathbb{K}_1$ and $\mathbb{K}_0 = \mathbb{K}_1$, $\mathbb{S}_0 \cup \mathbb{S}_1$ need to be guessed.
- $T_0 \xleftarrow{\mathbb{K}_0} \mathbb{S}_0$, $\mathbb{S}_1 \Rightarrow \mathbb{K}_1 \Rightarrow \mathbb{K}_0 \xrightarrow{T_0} \mathbb{S}_0$



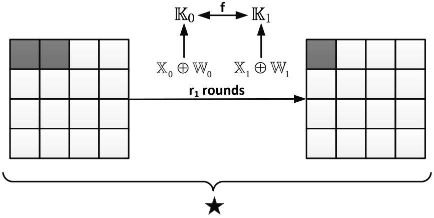
Key Relations to Improve the Complexity

- By the key-schedule, a lot of key relations can be found.
- These key relations cannot improve the complexity directly.
- $\mathbb{S}_0 \Rightarrow \mathbb{K}_0$, $\mathbb{S}_1 \Rightarrow \mathbb{K}_1$ and $\mathbb{K}_0 = \mathbb{K}_1$, $\mathbb{S}_0 \cup \mathbb{S}_1$ need to be guessed.
- $T_0 \xleftarrow{\mathbb{K}_0} \mathbb{S}_0$, $\mathbb{S}_1 \Rightarrow \mathbb{K}_1 \Rightarrow \mathbb{K}_0 \xrightarrow{T_0} \mathbb{S}_0$
- $\mathbb{S}_0 \xrightarrow{T_0} \mathbb{S}_1 \xrightarrow{T_1} \mathbb{S}_2$



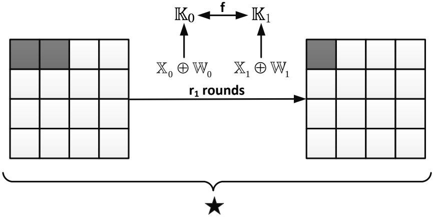
Key Relations to Improve the Complexity

- By the key-schedule, a lot of key relations can be found.
- These key relations cannot improve the complexity directly.
- $\mathbb{S}_0 \Rightarrow \mathbb{K}_0$, $\mathbb{S}_1 \Rightarrow \mathbb{K}_1$ and $\mathbb{K}_0 = \mathbb{K}_1$, $\mathbb{S}_0 \cup \mathbb{S}_1$ need to be guessed.
- $T_0 \xleftarrow{\mathbb{K}_0} \mathbb{S}_0$, $\mathbb{S}_1 \Rightarrow \mathbb{K}_1 \Rightarrow \mathbb{K}_0 \xrightarrow{T_0} \mathbb{S}_0$
- $\mathbb{S}_0 \xrightarrow{T_0} \mathbb{S}_1 \xrightarrow{T_1} \mathbb{S}_2$
- $\mathbb{K}_0 = f(\mathbb{K}_1)$, $\mathbb{W}_0 \oplus \mathbb{K}_0 = \mathbb{X}_0$ and $\mathbb{W}_1 \oplus \mathbb{K}_1 = \mathbb{X}_1$.
  $T_3 \xleftarrow{\chi=\mathbb{X}_0 \oplus f(\mathbb{W}_1)}$ States, $\chi' = \mathbb{W}_0 \oplus f(\mathbb{X}_1) \xrightarrow{T_3}$ States.



Key Relations to Improve the Complexity

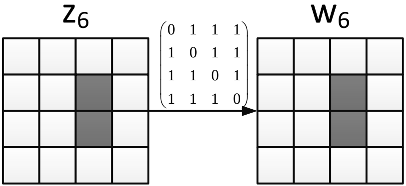
- By the key-schedule, a lot of key relations can be found.
- These key relations cannot improve the complexity directly.
- $\mathbb{S}_0 \Rightarrow \mathbb{K}_0$, $\mathbb{S}_1 \Rightarrow \mathbb{K}_1$ and $\mathbb{K}_0 = \mathbb{K}_1$, $\mathbb{S}_0 \cup \mathbb{S}_1$ need to be guessed.
- $T_0 \xleftarrow{\mathbb{K}_0} \mathbb{S}_0$, $\mathbb{S}_1 \Rightarrow \mathbb{K}_1 \Rightarrow \mathbb{K}_0 \xrightarrow{T_0} \mathbb{S}_0$
- $\mathbb{S}_0 \xrightarrow{T_0} \mathbb{S}_1 \xrightarrow{T_1} \mathbb{S}_2$
- $\mathbb{K}_0 = f(\mathbb{K}_1)$, $\mathbb{W}_0 \oplus \mathbb{K}_0 = \mathbb{X}_0$ and $\mathbb{W}_1 \oplus \mathbb{K}_1 = \mathbb{X}_1$.
  $T_3 \xleftarrow{\chi = \mathbb{X}_0 \oplus f(\mathbb{W}_1)}$ States, $\chi' = \mathbb{W}_0 \oplus f(\mathbb{X}_1) \xrightarrow{T_3}$ States.
- **state-bridge technique**.



Outline

6-Round Distinguisher on Midori64

- By the MixColumn Operation:

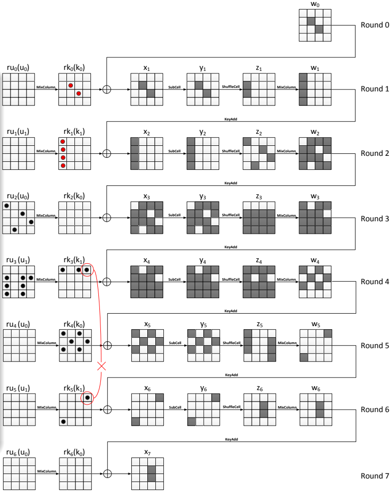


- Since $w_6[9] = z_6[8] \oplus z_6[10] \oplus z_6[11]$, $w_6[10] = z_6[8] \oplus z_6[9] \oplus z_6[11]$, then $w_6[9] \oplus w_6[10] = z_6[9] \oplus z_6[10]$. Let $e_{in} = z_6[9] \oplus z_6[10]$, $e_{out} = x_7[9] \oplus x_7[10]$, so $e_{out} = e_{in} \oplus rk_6[9] \oplus rk_6[10]$.

6-Round Distinguisher on Midori64

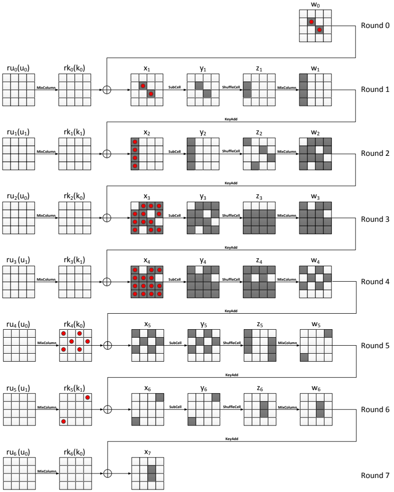
Proposition 6 (6-Round Distinguisher)

Let $\{w_0^0, \cdots, w_0^{255}\}$ be a 2-δ-set where $w_0[5]$ and $w_0[10]$ are the active nibbles. Consider the encryption of the first 33 values $(w_0^0, \cdots, w_0^{32})$ of the 2-δ-set through 6-round Midori64, in the case of that a message of the 2-δ-set belongs to a pair which conforms to the truncated differential trail outlined in the left, then the corresponding 128-bit ordered sequence $(e_{out}^1 \oplus e_{out}^0, \cdots, e_{out}^{32} \oplus e_{out}^0)$ only takes about 2^{104} values.



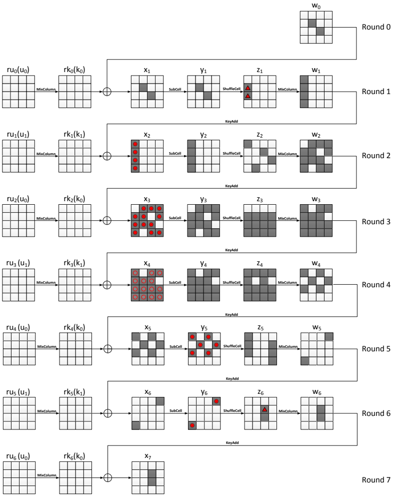
6-Round Distinguisher on Midori64

- The output sequence is determined by the 42 nibble-parameters below:



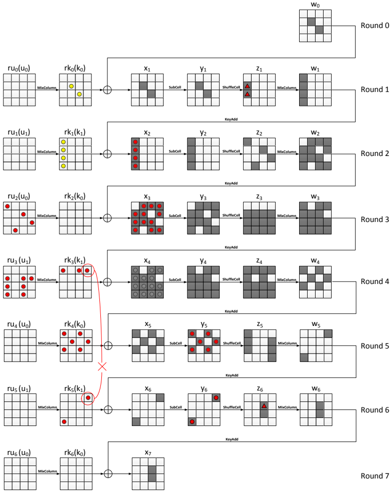
6-Round Distinguisher on Midori64

- The 42 nibble-parameters is determined by the 27 nibble-parameters:



6-Round Distinguisher on Midori64

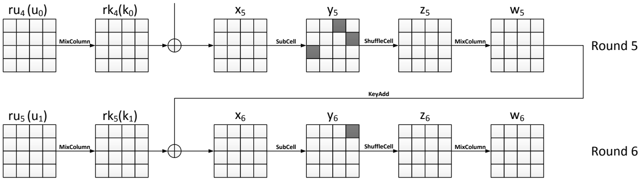
- The output sequence can take about 2^{104} values:



10-Round Attack on Midori64 (Precomputation Phase)

- Table T_1:

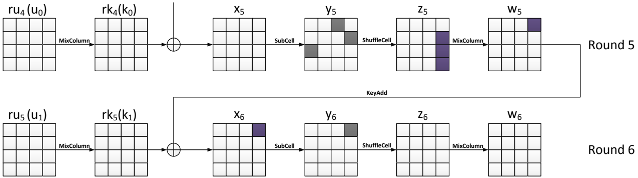
$y_6[12], y_5[2,8,13] \longrightarrow x_6[12], w_5[12] \longrightarrow rk_5[12]$



10-Round Attack on Midori64 (Precomputation Phase)

- Table T_1:

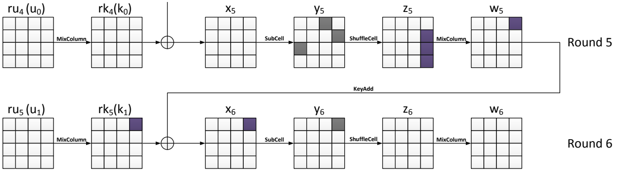
$$y_6[12], y_5[2,8,13] \longrightarrow x_6[12], w_5[12] \longrightarrow rk_5[12]$$



10-Round Attack on Midori64 (Precomputation Phase)

- Table T_1:

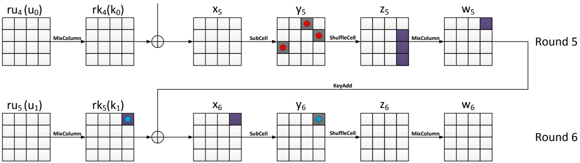
$$y_6[12], y_5[2, 8, 13] \longrightarrow x_6[12], w_5[12] \longrightarrow rk_5[12]$$



10-Round Attack on Midori64 (Precomputation Phase)

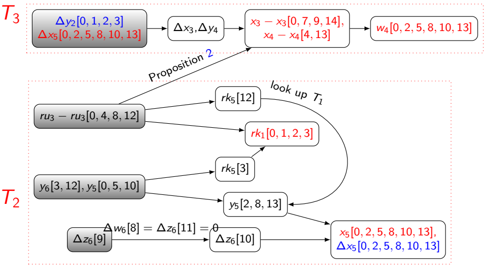
- Table T_1:

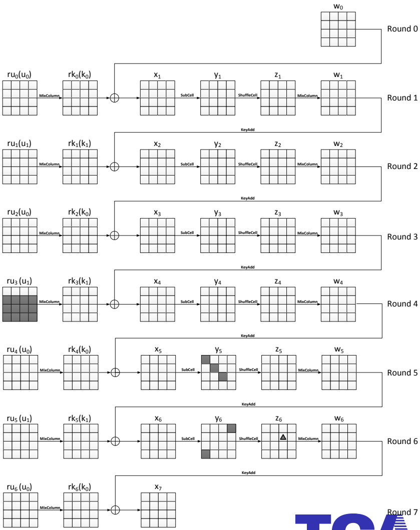
$y_6[12], y_5[2,8,13] \longrightarrow x_6[12], w_5[12] \longrightarrow rk_5[12]$



10-Round Attack on Midori64 (Precomputation Phase)

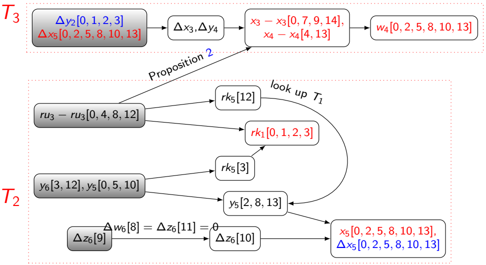
- Table T_2 and T_3:

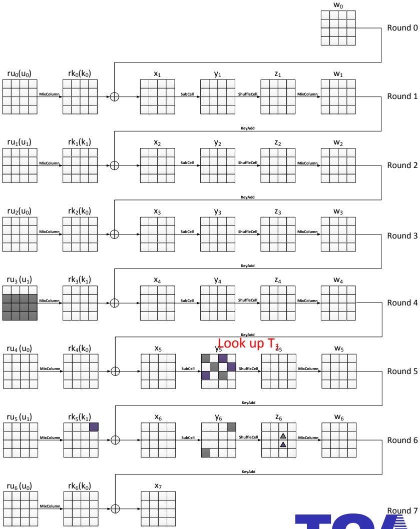




10-Round Attack on Midori64 (Precomputation Phase)

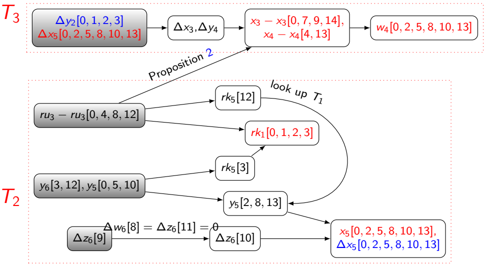
- Table T_2 and T_3:

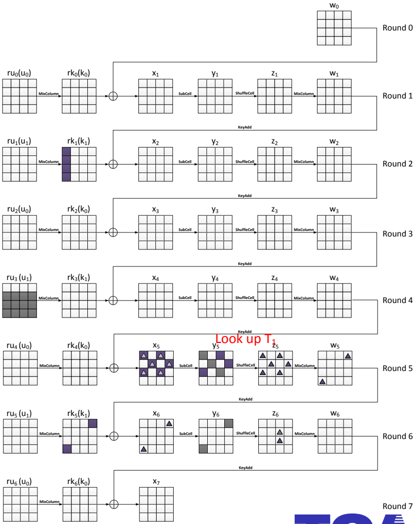




10-Round Attack on Midori64 (Precomputation Phase)

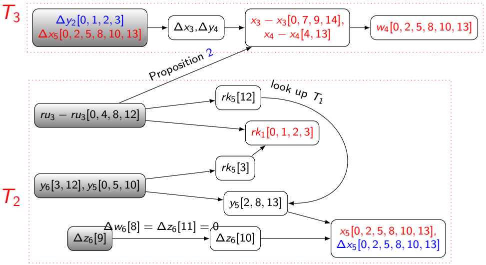
- Table T_2 and T_3:

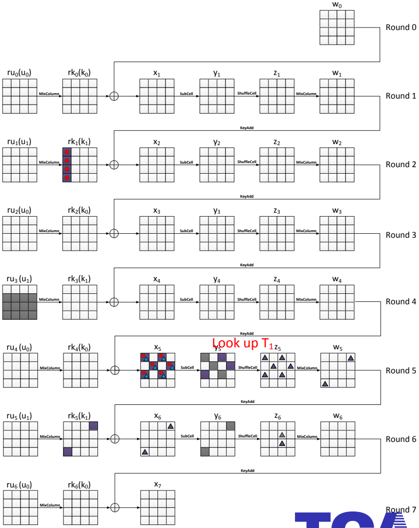




10-Round Attack on Midori64 (Precomputation Phase)

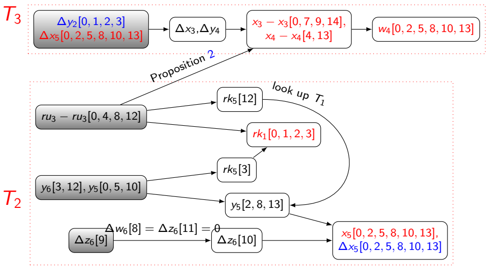
- Table T_2 and T_3:

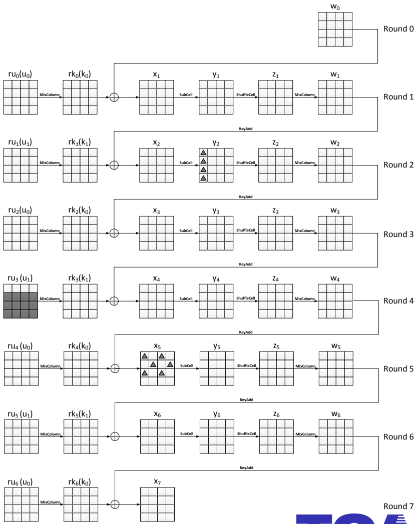




10-Round Attack on Midori64 (Precomputation Phase)

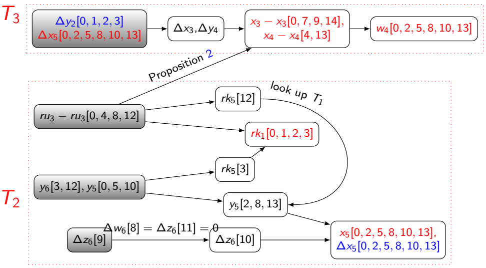
- Table T_2 and T_3:

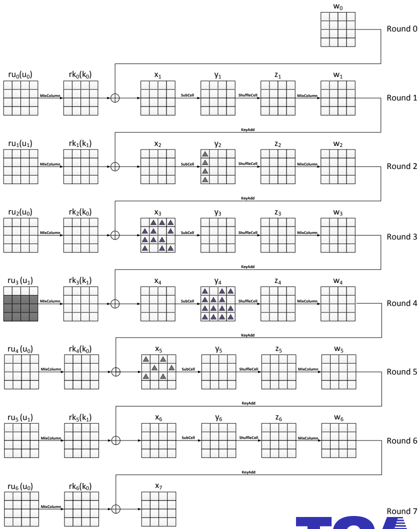




10-Round Attack on Midori64 (Precomputation Phase)

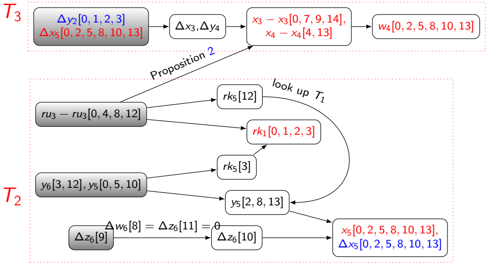
- Table T_2 and T_3:

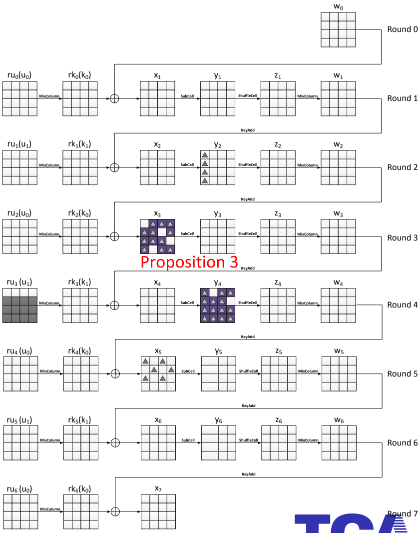




10-Round Attack on Midori64 (Precomputation Phase)

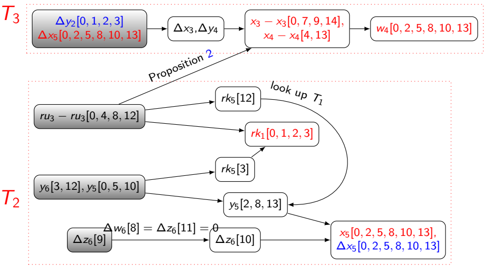
- Table T_2 and T_3:

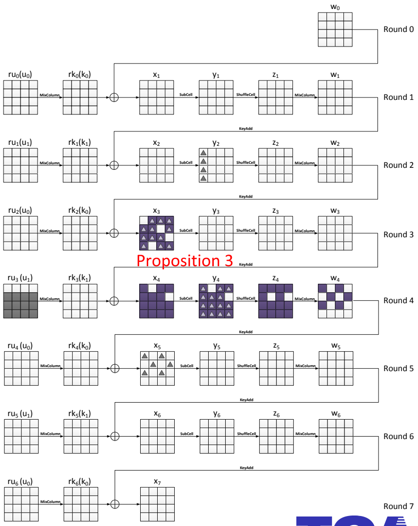




10-Round Attack on Midori64 (Precomputation Phase)

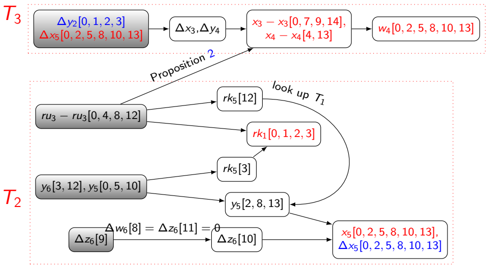
- Table T_2 and T_3:

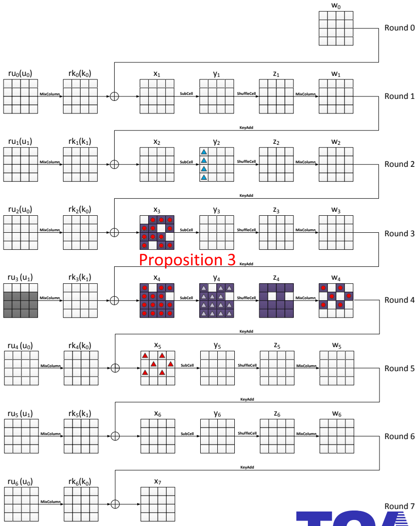




10-Round Attack on Midori64 (Precomputation Phase)

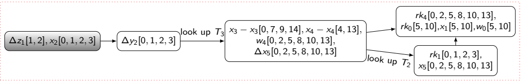
- Table T_2 and T_3:



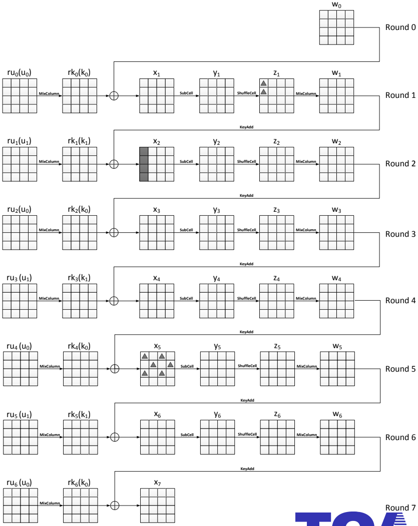


10-Round Attack on Midori64 (Precomputation Phase)

- Table T_4:

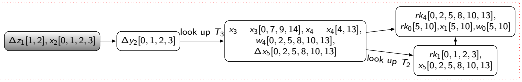


get the 42 nibble-parameters of Proposition 6

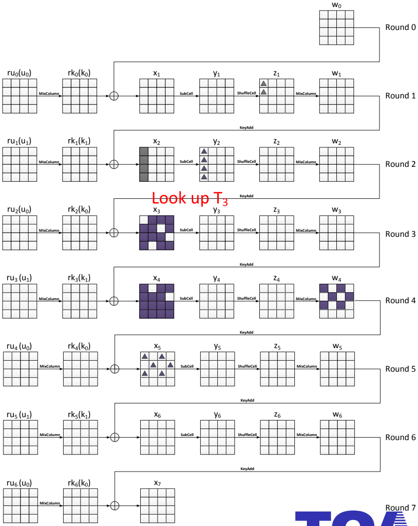


10-Round Attack on Midori64 (Precomputation Phase)

- Table T_4:

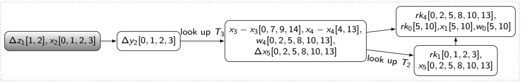


get the 42 nibble-parameters of Proposition 6

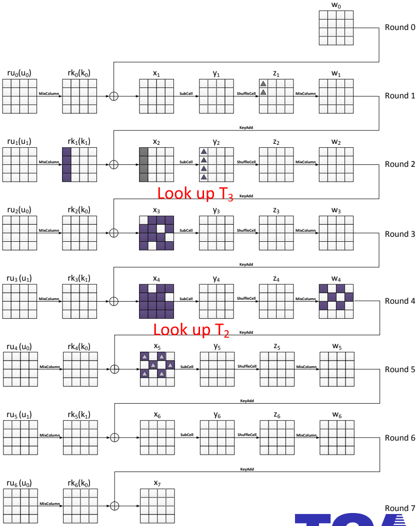


10-Round Attack on Midori64 (Precomputation Phase)

- Table T_4:

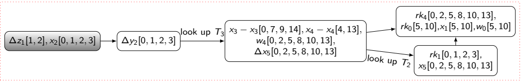


get the 42 nibble-parameters of Proposition 6

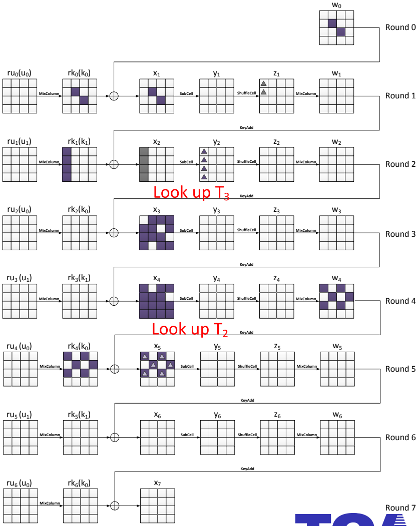


10-Round Attack on Midori64 (Precomputation Phase)

- Table T_4:

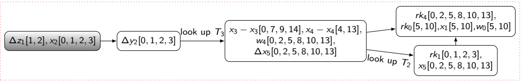


get the 42 nibble-parameters of Proposition 6

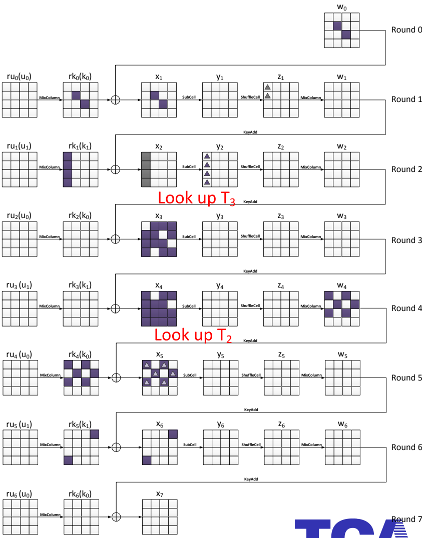


10-Round Attack on Midori64 (Precomputation Phase)

- Table T_4:

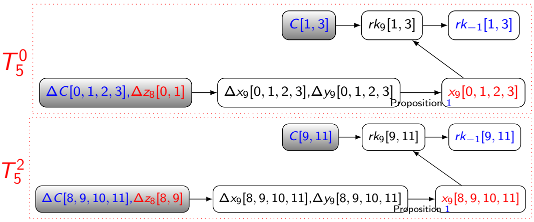


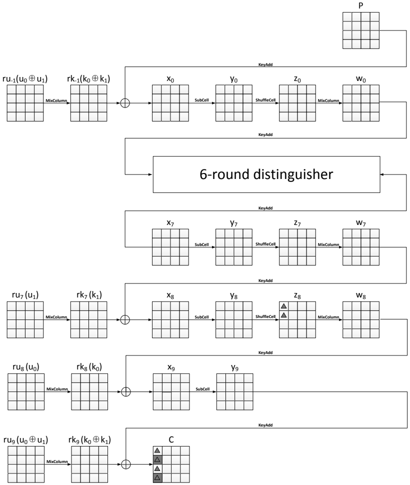
get the 42 nibble-parameters of Proposition 6



10-Round Attack on Midori64 (Precomputation Phase)

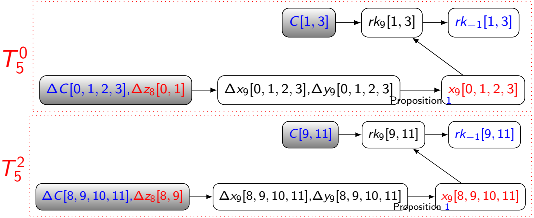
- Table T_5^0, T_5^2:

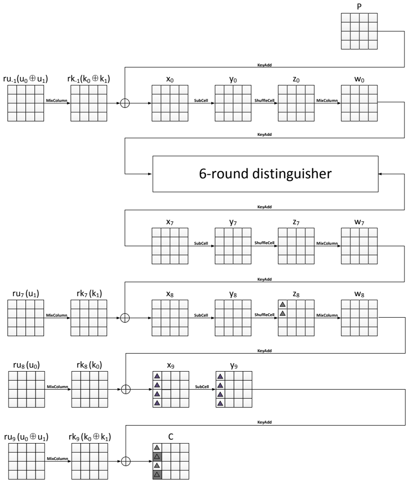




10-Round Attack on Midori64 (Precomputation Phase)

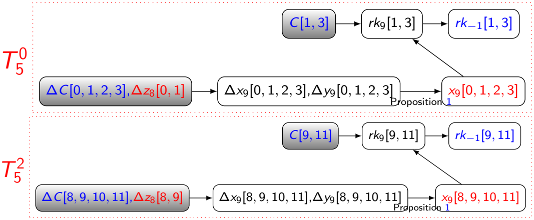
- Table T_5^0, T_5^2:

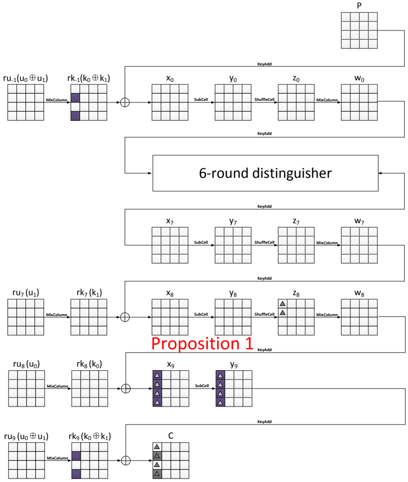




10-Round Attack on Midori64 (Precomputation Phase)

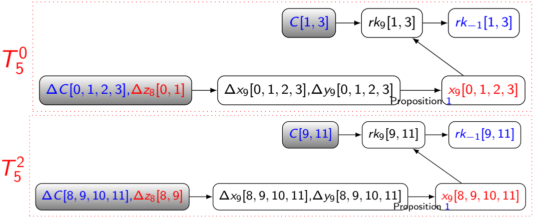
- Table T_5^0, T_5^2:

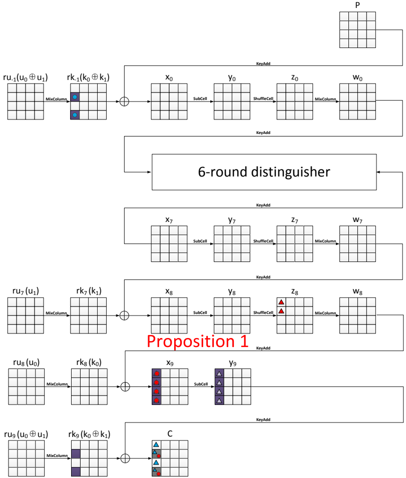




10-Round Attack on Midori64 (Precomputation Phase)

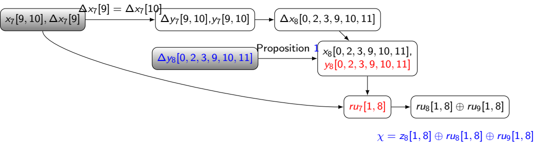
- Table T_5^0, T_5^2:

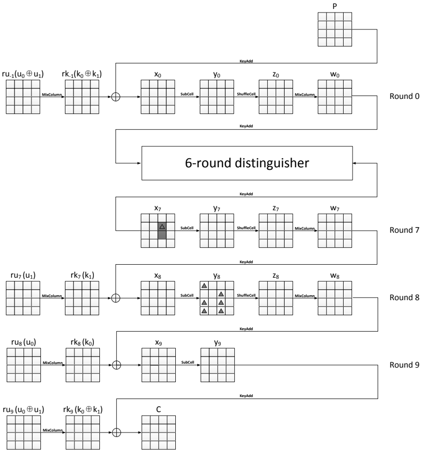




10-Round Attack on Midori64 (Precomputation Phase)

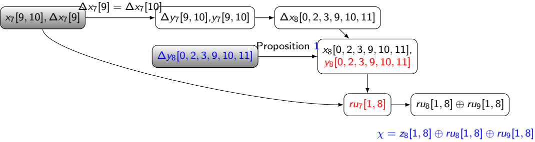
- Table T_6 (State-Bridge Techinique):

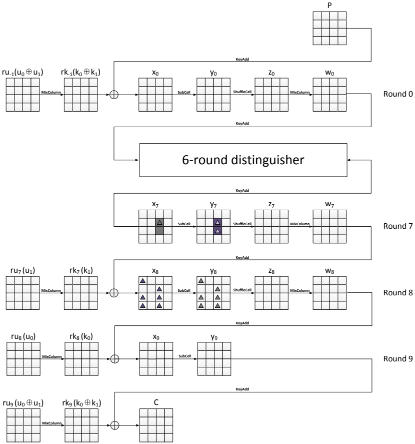




10-Round Attack on Midori64 (Precomputation Phase)

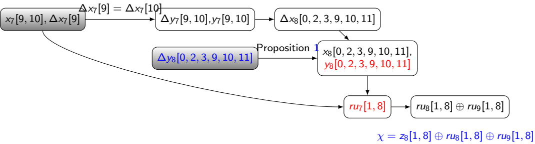
- Table T_6 (State-Bridge Techinique):

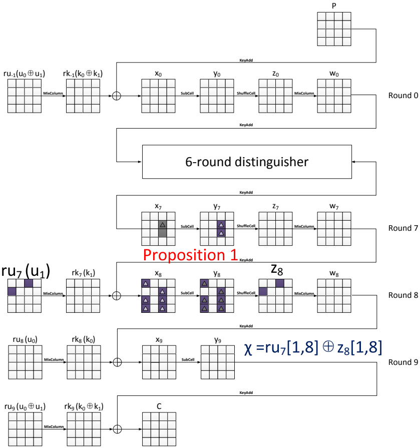




10-Round Attack on Midori64 (Precomputation Phase)

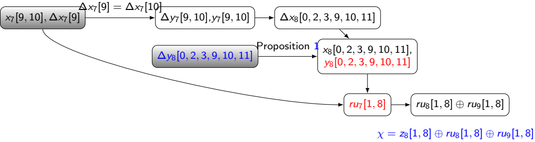
- Table T_6 (State-Bridge Techinique):

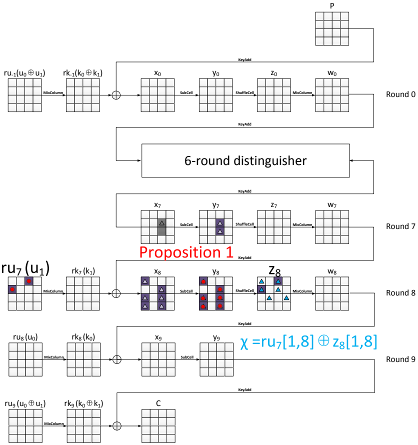




10-Round Attack on Midori64 (Precomputation Phase)

- Table T_6 (State-Bridge Techinique):



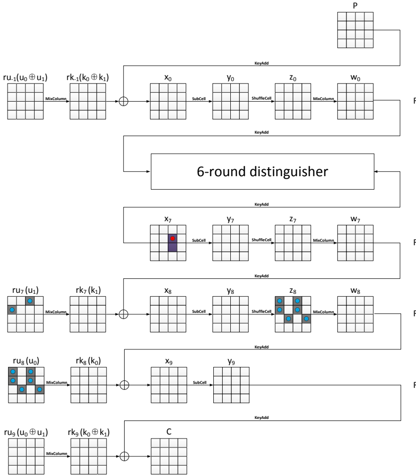


10-Round Attack on Midori64 (Precomputation Phase)

- Table T_7:

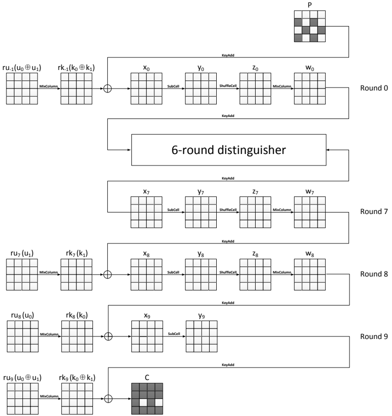
$\overline{w_8}[0, 1, 6, 8, 9, 14]$ → e_{out}

$ru_7[1, 8], ru_8[0, 1, 6, 8, 9, 14]$ → e_{out}



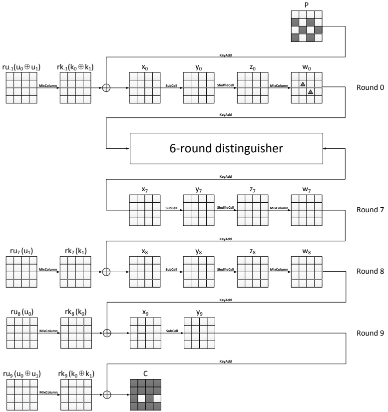
10-Round Attack on Midori64 (Online Phase)

- Define a structure of 2^{24} plaintexts where $P[1, 3, 6, 9, 11, 14]$ takes all the possible values, then we can get 2^{47} pairs. Choose 2^{29} structures to get about 2^{76} pairs. About 2^{68} pairs to verify that $\Delta C[6, 14] = 0$;



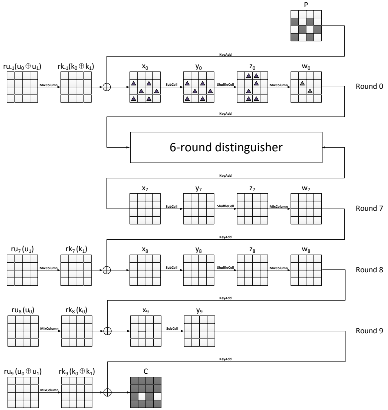
10-Round Attack on Midori64 (Online Phase)

- Define a structure of 2^{24} plaintexts where $P[1, 3, 6, 9, 11, 14]$ takes all the possible values, then we can get 2^{47} pairs. Choose 2^{29} structures to get about 2^{76} pairs. About 2^{68} pairs to verify that $\Delta C[6, 14] = 0$;
- For each of the 2^{68} remaining pairs:
    - Guess $\Delta w_0[5, 10]$, deduce Δy_0. Deduce x_0, then deduce rk_{-1};



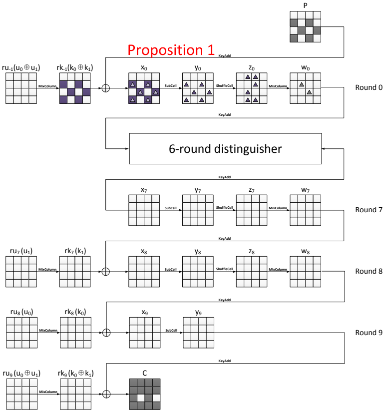
10-Round Attack on Midori64 (Online Phase)

- Define a structure of 2^{24} plaintexts where $P[1,3,6,9,11,14]$ takes all the possible values, then we can get 2^{47} pairs. Choose 2^{29} structures to get about 2^{76} pairs. About 2^{68} pairs to verify that $\Delta C[6,14]=0$;
- For each of the 2^{68} remaining pairs:
  - Guess $\Delta w_0[5,10]$, deduce Δy_0. Deduce x_0, then deduce rk_{-1};



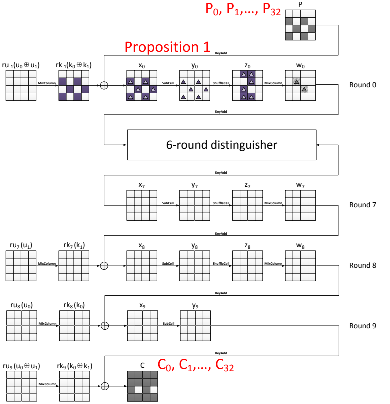
10-Round Attack on Midori64 (Online Phase)

- Define a structure of 2^{24} plaintexts where $P[1, 3, 6, 9, 11, 14]$ takes all the possible values, then we can get 2^{47} pairs. Choose 2^{29} structures to get about 2^{76} pairs. About 2^{68} pairs to verify that $\Delta C[6, 14] = 0$;
- For each of the 2^{68} remaining pairs:
  - Guess $\Delta w_0[5, 10]$, deduce Δy_0. Deduce x_0, then deduce rk_{-1};



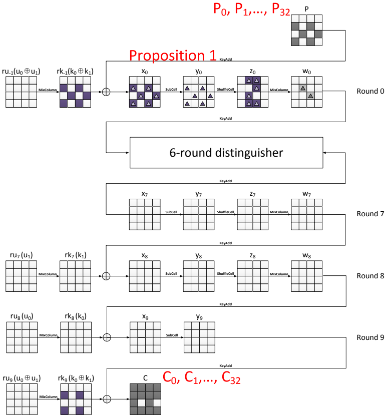
10-Round Attack on Midori64 (Online Phase)

- Define a structure of 2^{24} plaintexts where $P[1,3,6,9,11,14]$ takes all the possible values, then we can get 2^{47} pairs. Choose 2^{29} structures to get about 2^{76} pairs. About 2^{68} pairs to verify that $\Delta C[6,14]=0$;
- For each of the 2^{68} remaining pairs:
  - Guess $\Delta w_0[5,10]$, deduce Δy_0. Deduce x_0, then deduce rk_{-1};
  - Deduce $z_0[4,6,7,8,9,11]$, Change the value of $w_0[5,10]$ to be $(0,1,\cdots,32)$ and compute their corresponding plaintexts $(P^0,P^1,\cdots,P^{32})$, then get the corresponding ciphertexts;



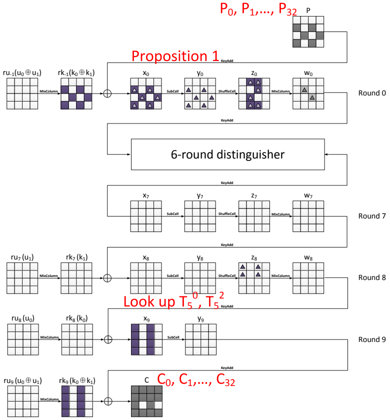
10-Round Attack on Midori64 (Online Phase)

- Define a structure of 2^{24} plaintexts where $P[1,3,6,9,11,14]$ takes all the possible values, then we can get 2^{47} pairs. Choose 2^{29} structures to get about 2^{76} pairs. About 2^{68} pairs to verify that $\Delta C[6,14] = 0$;
- For each of the 2^{68} remaining pairs:
  - Guess $\Delta w_0[5,10]$, deduce Δy_0. Deduce x_0, then deduce rk_{-1};
  - Deduce $z_0[4,6,7,8,9,11]$, Change the value of $w_0[5,10]$ to be $(0,1,\cdots,32)$ and compute their corresponding plaintexts $(P^0, P^1, \cdots, P^{32})$, then get the corresponding ciphertexts;
  - For each of the deduced $rk_{-1}[1,3,6,9,11,14]$, compute $rk_9[1,3]$ and $rk_9[9,11]$. Look up T_5^0 and T_5^2, deduce $x_9[0,1,2,3]$, $\Delta z_8[0,1]$ and $x_9[8,9,10,11]$, $\Delta z_8[8,9]$, deduce $rk_9[0,2]$ and $rk_9[8,10]$;



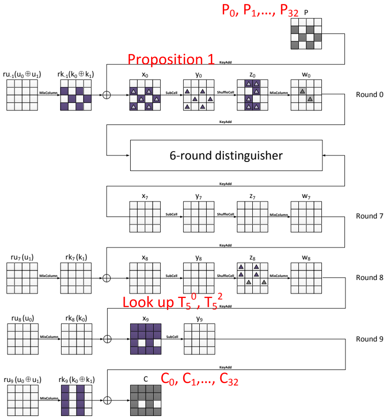
10-Round Attack on Midori64 (Online Phase)

- Define a structure of 2^{24} plaintexts where $P[1, 3, 6, 9, 11, 14]$ takes all the possible values, then we can get 2^{47} pairs. Choose 2^{29} structures to get about 2^{76} pairs. About 2^{68} pairs to verify that $\Delta C[6, 14] = 0$;
- For each of the 2^{68} remaining pairs:
  - Guess $\Delta w_0[5, 10]$, deduce Δy_0. Deduce x_0, then deduce rk_{-1};
  - Deduce $z_0[4, 6, 7, 8, 9, 11]$, Change the value of $w_0[5, 10]$ to be $(0, 1, \cdots, 32)$ and compute their corresponding plaintexts $(P^0, P^1, \cdots, P^{32})$, then get the corresponding ciphertexts;
  - For each of the deduced $rk_{-1}[1, 3, 6, 9, 11, 14]$, compute $rk_9[1, 3]$ and $rk_9[9, 11]$. Look up T_5^0 and T_5^2, deduce $x_9[0, 1, 2, 3]$, $\Delta z_8[0, 1]$ and $x_9[8, 9, 10, 11]$, $\Delta z_8[8, 9]$, deduce $rk_9[0, 2]$ and $rk_9[8, 10]$;



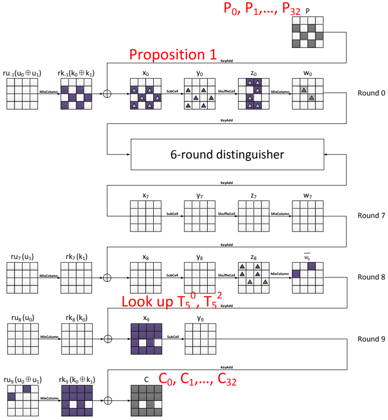
10-Round Attack on Midori64 (Online Phase)

- Define a structure of 2^{24} plaintexts where $P[1, 3, 6, 9, 11, 14]$ takes all the possible values, then we can get 2^{47} pairs. Choose 2^{29} structures to get about 2^{76} pairs. About 2^{68} pairs to verify that $\Delta C[6, 14] = 0$;
- For each of the 2^{68} remaining pairs:
    - Guess $\Delta w_0[5, 10]$, deduce Δy_0. Deduce x_0, then deduce rk_{-1};
    - Deduce $z_0[4, 6, 7, 8, 9, 11]$, Change the value of $w_0[5, 10]$ to be $(0, 1, \cdots, 32)$ and compute their corresponding plaintexts $(P^0, P^1, \cdots, P^{32})$, then get the corresponding ciphertexts;
    - For each of the deduced $rk_{-1}[1, 3, 6, 9, 11, 14]$, compute $rk_9[1, 3]$ and $rk_9[9, 11]$. Look up T_5^0 and T_5^2, deduce $x_9[0, 1, 2, 3]$, $\Delta z_8[0, 1]$ and $x_9[8, 9, 10, 11]$, $\Delta z_8[8, 9]$, deduce $rk_9[0, 2]$ and $rk_9[8, 10]$;
    - Guess $\Delta z_8[6, 14]$, deduce Columns 1 and 3 of Δx_9, then deduce Columns 1 and 3 of rk_9 and x_9. Deduce $ru_9[1, 8]$ and $\overline{w_8}[1, 8]$, so $\chi' = ru_9[1, 8] \oplus \overline{w_8}[1, 8]$ can be deduced. Look up T_6 to deduce y_8 and $ru_7[1, 8]$. Then deduce ru_8;



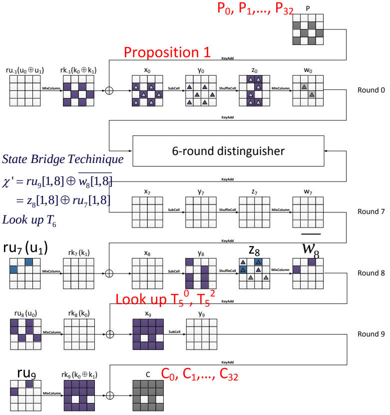
10-Round Attack on Midori64 (Online Phase)

- Define a structure of 2^{24} plaintexts where $P[1, 3, 6, 9, 11, 14]$ takes all the possible values, then we can get 2^{47} pairs. Choose 2^{29} structures to get about 2^{76} pairs. About 2^{68} pairs to verify that $\Delta C[6, 14] = 0$;
- For each of the 2^{68} remaining pairs:
  - Guess $\Delta w_0[5, 10]$, deduce Δy_0. Deduce x_0, then deduce rk_{-1};
  - Deduce $z_0[4, 6, 7, 8, 9, 11]$, Change the value of $w_0[5, 10]$ to be $(0, 1, \cdots, 32)$ and compute their corresponding plaintexts $(P^0, P^1, \cdots, P^{32})$, then get the corresponding ciphertexts;
  - For each of the deduced $rk_{-1}[1, 3, 6, 9, 11, 14]$, compute $rk_9[1, 3]$ and $rk_9[9, 11]$. Look up T_5^0 and T_5^2, deduce $x_9[0, 1, 2, 3]$, $\Delta z_8[0, 1]$ and $x_9[8, 9, 10, 11]$, $\Delta z_8[8, 9]$, deduce $rk_9[0, 2]$ and $rk_9[8, 10]$;
  - Guess $\Delta z_8[6, 14]$, deduce Columns 1 and 3 of Δx_9, then deduce Columns 1 and 3 of rk_9 and x_9. Deduce $ru_9[1, 8]$ and $\overline{w_8}[1, 8]$, so $\chi' = ru_9[1, 8] \oplus \overline{w_8}[1, 8]$ can be deduced. Look up T_6 to deduce y_8 and $ru_7[1, 8]$. Then deduce ru_8;



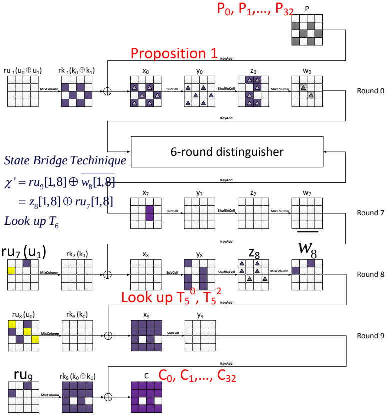
10-Round Attack on Midori64 (Online Phase)

- Define a structure of 2^{24} plaintexts where $P[1, 3, 6, 9, 11, 14]$ takes all the possible values, then we can get 2^{47} pairs. Choose 2^{29} structures to get about 2^{76} pairs. About 2^{68} pairs to verify that $\Delta C[6, 14] = 0$;
- For each of the 2^{68} remaining pairs:
  - Guess $\Delta w_0[5, 10]$, deduce Δy_0. Deduce x_0, then deduce rk_{-1};
  - Deduce $z_0[4, 6, 7, 8, 9, 11]$, Change the value of $w_0[5, 10]$ to be $(0, 1, \cdots, 32)$ and compute their corresponding plaintexts $(P^0, P^1, \cdots, P^{32})$, then get the corresponding ciphertexts;
  - For each of the deduced $rk_{-1}[1, 3, 6, 9, 11, 14]$, compute $rk_9[1, 3]$ and $rk_9[9, 11]$. Look up T_5^0 and T_5^2, deduce $x_9[0, 1, 2, 3]$, $\Delta z_8[0, 1]$ and $x_9[8, 9, 10, 11]$, $\Delta z_8[8, 9]$, deduce $rk_9[0, 2]$ and $rk_9[8, 10]$;
  - Guess $\Delta z_8[6, 14]$, deduce Columns 1 and 3 of Δx_9, then deduce Columns 1 and 3 of rk_9 and x_9. Deduce $ru_9[1, 8]$ and $\overline{w_8}[1, 8]$, so $\chi' = ru_9[1, 8] \oplus \overline{w_8}[1, 8]$ can be deduced. Look up T_6 to deduce y_8 and $ru_7[1, 8]$. Then deduce ru_8;



10-Round Attack on Midori64 (Online Phase)

- Define a structure of 2^{24} plaintexts where $P[1, 3, 6, 9, 11, 14]$ takes all the possible values, then we can get 2^{47} pairs. Choose 2^{29} structures to get about 2^{76} pairs. About 2^{68} pairs to verify that $\Delta C[6, 14] = 0$;
- For each of the 2^{68} remaining pairs:
  - Guess $\Delta w_0[5, 10]$, deduce Δy_0. Deduce x_0, then deduce rk_{-1};
  - Deduce $z_0[4, 6, 7, 8, 9, 11]$, Change the value of $w_0[5, 10]$ to be $(0, 1, \cdots, 32)$ and compute their corresponding plaintexts $(P^0, P^1, \cdots, P^{32})$, then get the corresponding ciphertexts;
  - For each of the deduced $rk_{-1}[1, 3, 6, 9, 11, 14]$, compute $rk_9[1, 3]$ and $rk_9[9, 11]$. Look up T_5^0 and T_5^2, deduce $x_9[0, 1, 2, 3]$, $\Delta z_8[0, 1]$ and $x_9[8, 9, 10, 11]$, $\Delta z_8[8, 9]$, deduce $rk_9[0, 2]$ and $rk_9[8, 10]$;
  - Guess $\Delta z_8[6, 14]$, deduce Columns 1 and 3 of Δx_9, then deduce Columns 1 and 3 of rk_9 and x_9. Deduce $ru_9[1, 8]$ and $\overline{w_8}[1, 8]$, so $\chi' = ru_9[1, 8] \oplus \overline{w_8}[1, 8]$ can be deduced. Look up T_6 to deduce y_8 and $ru_7[1, 8]$. Then deduce ru_8;
  - Using these keys and T_7 to deduce Δe_{out} from the ciphertexts, Check whether it lies in the precomputation table T_4. If not, try another one. If so, we check whether $ru_2[0, 9, 14]||ru_3[1]$ matches $ru_8[0, 9, 14]||ru_7[1]$.



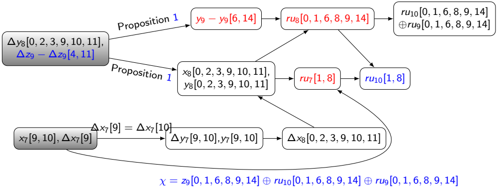
Complexity Analysis

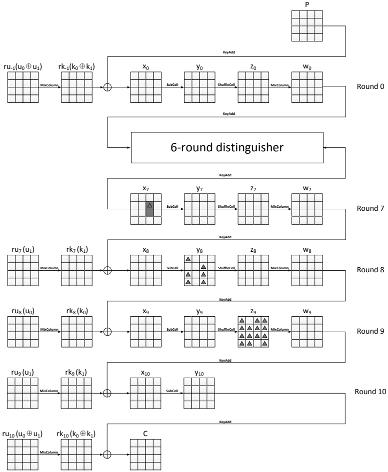
- In the precomputation phase, in order to construct T_4, we need to perform 2^{104} partial encryptions on 33 messages.
- In the online phase, we need to perform 2^{20+68} partial encryptions on 33 messages.
- With data/time/memory tradeoff, the adversary only need to precompute a fraction of $2^{-8.5}$ of possible sequences, and in the online phase, repeat the attack $2^{8.5}$ times to offset the probability of the failure. Otherwise, Using the relation of $ru_3[1]$, the attack can be divided into 2^4 weak-key attacks.
- In total, the time complexity of this attack is $2^{99.5}$ 10-round Midori64 encryptions, the data complexity is $2^{61.5}$ chosen-plaintexts and the memory complexity is $2^{92.7}$ 64-bit blocks.

Outline

1. Preliminaries
2. Meet-in-the-Middle Attack on 10-Round Midori64
3. Meet-in-the-Middle Attack on 11-Round Midori64
4. Meet-in-the-Middle Attack on 12-Round Midori64
5. Conclusions

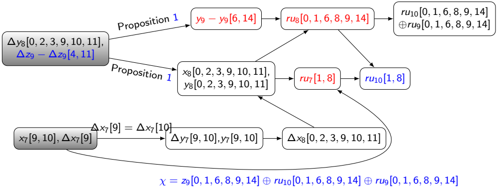
Meet-in-the-Middle Attack on 11-Round Midori64

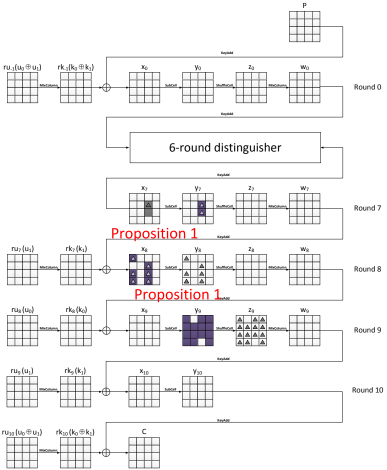




- The precomputation phase is the same as the 10-round attack, except the online phase and the tables for online phase.
- T_6 (State-Bridge Technique)

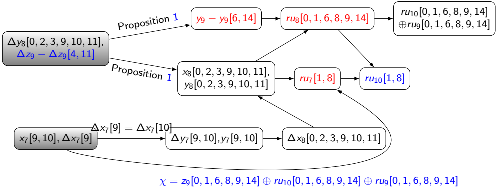
Meet-in-the-Middle Attack on 11-Round Midori64

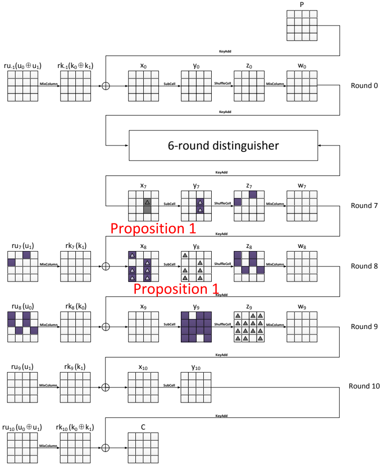




- The precomputation phase is the same as the 10-round attack, except the online phase and the tables for online phase.
- T_6 (State-Bridge Technique)

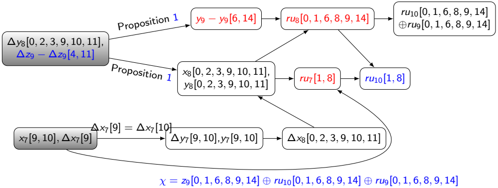
Meet-in-the-Middle Attack on 11-Round Midori64

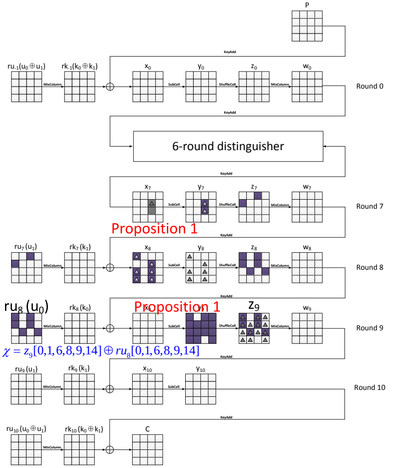




- The precomputation phase is the same as the 10-round attack, except the online phase and the tables for online phase.
- T_6 (State-Bridge Technique)

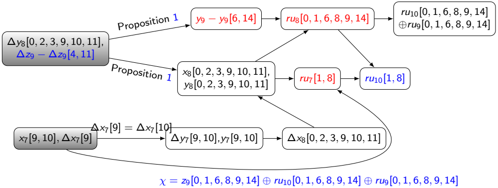
Meet-in-the-Middle Attack on 11-Round Midori64

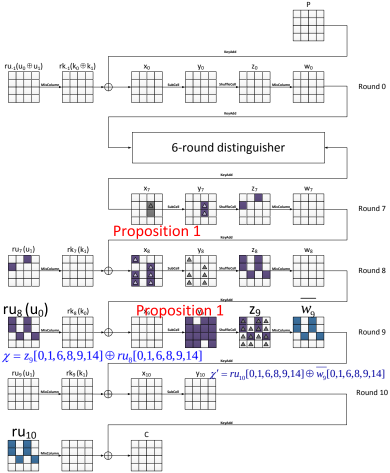




- The precomputation phase is the same as the 10-round attack, except the online phase and the tables for online phase.
- T_6 (State-Bridge Technique)

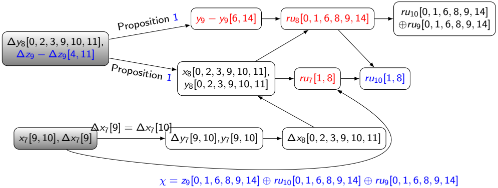
Meet-in-the-Middle Attack on 11-Round Midori64



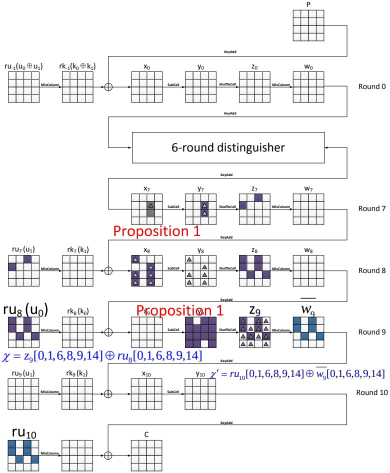


- The precomputation phase is the same as the 10-round attack, except the online phase and the tables for online phase.
- T_6 (State-Bridge Technique)
- Online Phase.

Meet-in-the-Middle Attack on 11-Round Midori64



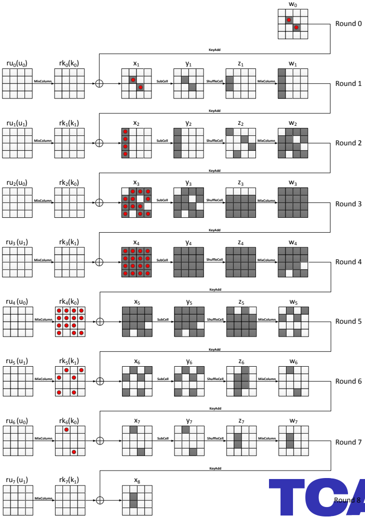
- The time complexity of this attack is 2^{122} 11-round Midori64 encryptions, the data complexity is 2^{53} chosen-plaintexts and the memory complexity is $2^{89.2}$ 64-bit blocks.



Outline

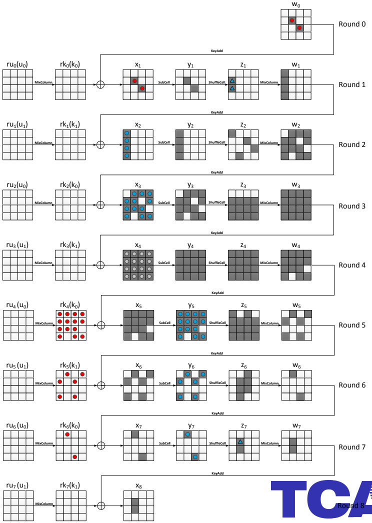
Meet-in-the-Middle Attack on 12-Round Midori64

- Precomputation phase:
  - By guessing the 58 nibble-parameters, we can deduce Δe_{out} from 2-δ-set;



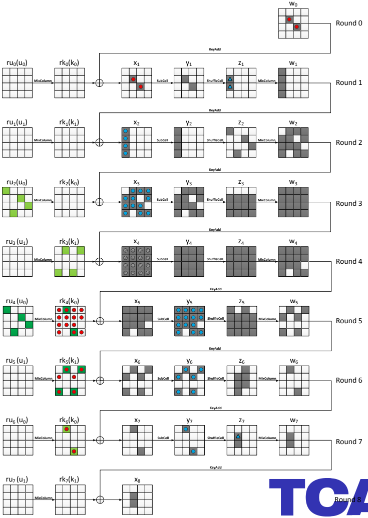
Meet-in-the-Middle Attack on 12-Round Midori64

- Precomputation phase:
  - By guessing the 58 nibble-parameters, we can deduce Δe_{out} from 2-δ-set;
  - If a pair of messages conforms to the truncated differential trail, the above 58 nibble-parameters are determined by the 41 nibble- parameters.



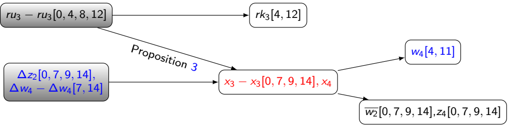
Meet-in-the-Middle Attack on 12-Round Midori64

- Precomputation phase:
  - By guessing the 58 nibble-parameters, we can deduce Δe_{out} from 2-δ-set;
  - If a pair of messages conforms to the truncated differential trail, the above 58 nibble-parameters are determined by the 41 nibble- parameters.
  - There are 10 key-relations in this distinguisher, then 58 nibble-parameters can take about 2^{124} values.

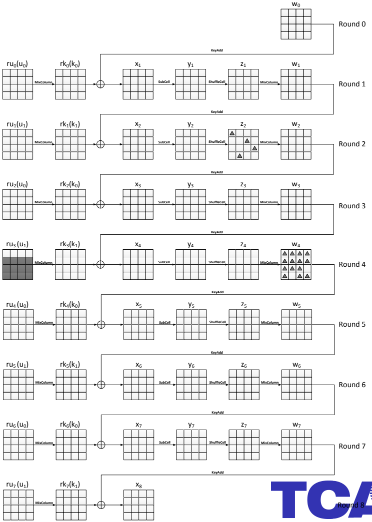


Meet-in-the-Middle Attack on 12-Round Midori64

- Table T_1 (State-Bridge Technique).

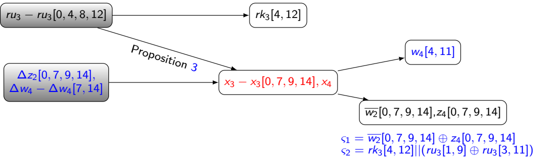


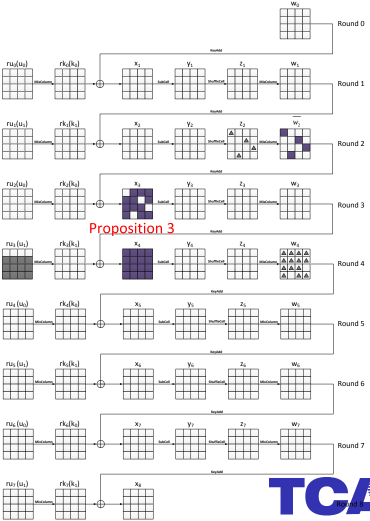
$\varsigma_1 = \overline{w_2}[0,7,9,14] \oplus z_4[0,7,9,14]$
$\varsigma_2 = rk_3[4,12] || (ru_3[1,9] \oplus ru_3[3,11])$



Meet-in-the-Middle Attack on 12-Round Midori64

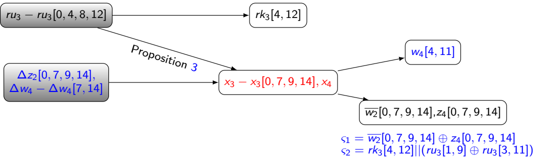
- Table T_1 (State-Bridge Technique).

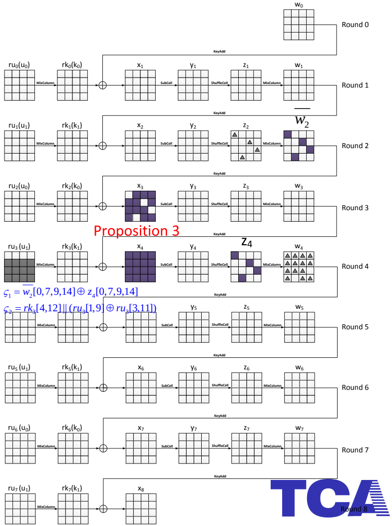




Meet-in-the-Middle Attack on 12-Round Midori64

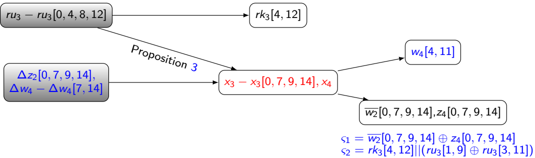
- Table T_1 (State-Bridge Technique).

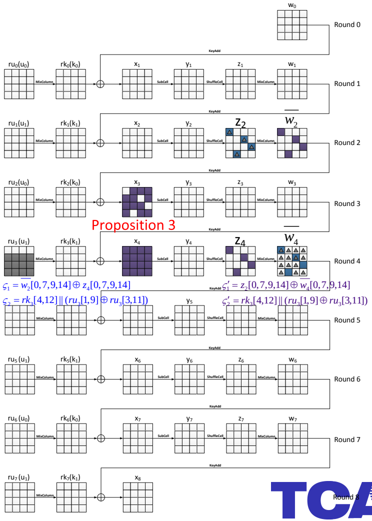




Meet-in-the-Middle Attack on 12-Round Midori64

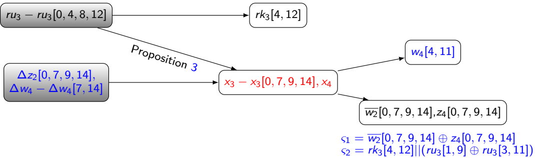
- Table T_1 (State-Bridge Technique).
- At the construction of Table T_4.

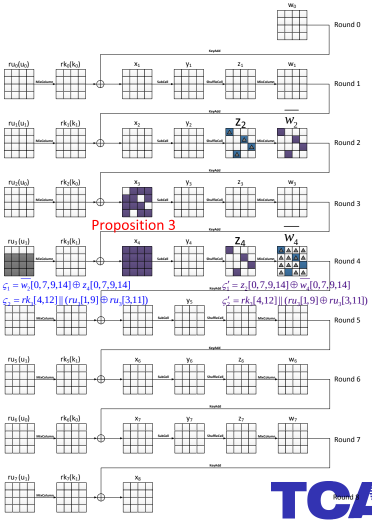




Meet-in-the-Middle Attack on 12-Round Midori64

- Table T_1 (State-Bridge Technique).
- At the construction of Table T_4.
- The online phase is almost the same as the 11-round attack.





Outline

Conclusions

- In this paper, we discussed the security of Midori64 against meet-in-the-middle attacks. To the best of our knowledge, this is the best attack on Midori64 in the single-key setting.
- The differential enumeration, key-bridging and key-dependent sieve techniques are used.
- We propose the state-bridge technique to use some key relations that are quite complicated and divided by some rounds to achieve the complexity lower bound.

Thank you for your attension!