APPENDIX A
BRIEF SPECIFICATION OF ANDRX CIPHERS

A. Specification of SIMON

Designed for resource-constrained environments, the SIMON [1] block cipher employs a balanced Feistel structure optimized for hardware efficiency. Following the ARX paradigm (combining Addition, Rotation, and XOR operations) seen in designs like ThreeFish and BLAKE, SIMON's round function utilizes only three elementary operations: bitwise AND, XOR, and circular bit Rotation. This minimalist approach ensures exceptional hardware performance while maintaining cryptographic strength.

The cipher supports multiple configurations parameterized by word size (n bits) and key expansion ratio (m). With a block size of $2n$ bits where $n \in \{16, 24, 32, 48, 64\}$, and key sizes of $m \times n$ bits for $m \in \{2, 3, 4\}$, each variant is denoted as SIMON2n/mn. The round count for every configuration is determined through security analysis based on both block and key dimensions, with specific values detailed in Table I. Each round starts with the state represented by two words: (L_i, R_i). The

TABLE I: SIMON parameters

Variant	Block size	Key size	Word size	Key words	Round
SIMON32	32	64	16	4	32
SIMON48	48	72 96	24	3 4	36 36
SIMON64	64	96 128	32	3 4	42 44
SIMON96	96	96 144	48	2 3	52 54
SIMON128	128	128 192 256	64	2 3 4	68 69 72

left word L_i is passed through a round function F, defined as:

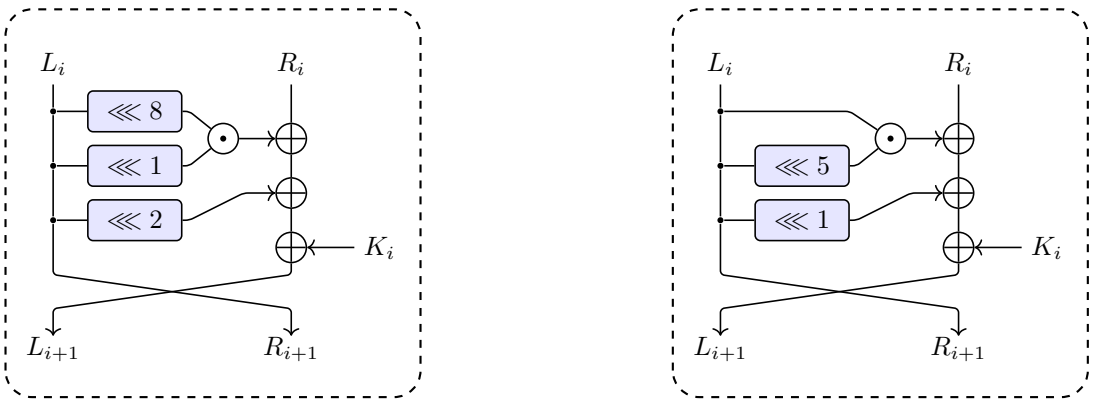
$$F(L_i) = (L_i \lll 1) \wedge (L_i \lll 8) \oplus (L_i \lll 2),$$

where $\lll$ denotes left rotation. The result of $F(L_i)$ is XORed with R_i and the subkey K_i, and then the two words are swapped:

$$L_{i+1} = R_i \oplus F(L_i) \oplus K_i,$$

$$R_{i+1} = L_i.$$

After all rounds are completed, the final state (L_r, R_r) forms the ciphertext. The round function of SIMON is illustrated in Figure 1a. SIMON uses a linear feedback shift register (LFSR)-like process to generate the required r subkeys $K_0, K_1, \ldots, K_{r-1}$. Depending on the length of the secret key (which can consist of 2, 3, or 4 words), Simon employs slightly different key schedule procedures.



(a) SIMON round function (b) Simeck round function

Fig. 1: Round Functions of AndRX Ciphers

The key expansion process initializes the first w words (K_0 through K_{w-1}, where $w \in \{2, 3, 4\}$) directly from the master key. Subsequent round keys K_i for $i = w$ to $r - 1$ are then derived through the recurrence relation:

- For $w = 2$:

$$K_{i+2} = K_i \oplus (K_{i+1} \gg 3) \oplus (K_{i+1} \gg 1) \oplus c \oplus (z_j)_i,$$

- For $w = 3$:
$$K_{i+3} = K_i \oplus (K_{i+1} \gg 3) \oplus (K_{i+2} \gg 1) \oplus c \oplus (z_j)_i,$$
- For $w = 4$:
$$K_{i+4} = K_i \oplus (K_{i+1} \gg 3) \oplus (K_{i+3} \gg 1) \oplus c \oplus (z_j)_i,$$

where $c = 0xff\cdots fc$ is a fixed constant, and $(z_j)_i$ refers to the i-th bit of one of five predefined constant sequences z_0, z_1, z_2, z_3, z_4.

B. Specification of Simeck

The Simeck [2] family of lightweight block ciphers was designed to be hardware-efficient while maintaining strong security. Simeck adopts a simple and compact structure inspired by the SIMON cipher and employs bitwise operations for both encryption and key scheduling. Its primary goal is to provide a small footprint in hardware, making it suitable for constrained environments like IoT devices and embedded systems. The Simeck family includes variants with different block and key sizes: Simeck-32-64, Simeck-48-96, and Simeck-64-128.

Simeck is structured as a balanced Feistel network, where the state is divided into two equal-sized words (L_i, R_i) at the beginning of each round. The round function of Simeck is illustrated in Figure 1b. The round function is designed to be simple and hardware-friendly, operating as follows:

$$f(x) = (x \wedge (x \lll 5)) \oplus (x \lll 1),$$

where $\lll$ denotes a left cyclic shift, $\wedge$ represents bitwise AND, and $\oplus$ is the XOR operation. This function is applied to the left half of the state (L_i), after which the right half (R_i) is XORed with the result of $f(L_i)$ and the subkey K_i. The round operation can be described by:

$$L_{i+1} = R_i \oplus f(L_i) \oplus K_i,$$
$$R_{i+1} = L_i.$$

This process is repeated over T rounds, where T depends on the block size of the Simeck variant in use. The final state after the last round provides the ciphertext.

Simeck uses a compact and efficient key schedule that reuses the round function to generate the required subkeys. The master key is divided into multiple words depending on the key size. A linear feedback shift register (LFSR) structure is used to update the keywords and generate new subkeys for each round. The update rule for the key schedule is given by:

$$K_{i+t} = K_i \oplus (K_{i+t-1} \ggg 3) \oplus (K_{i+t-1} \ggg 1) \oplus z_j,$$

where $\ggg$ denotes a right shift, and z_j is a constant derived from a predefined sequence. This key schedule ensures that subkeys are generated efficiently, minimizing hardware requirements.

APPENDIX B
RELATIONSHIP BETWEEN ORIGINAL SUBKEY AND EQUIVALENT SUBKEY IN SIMON

$$\begin{cases} K^e_1 = K_0 \\ K^e_2 = (K^e_1 \lll 2) \oplus K_1 \\ K^e_3 = K^e_1 \oplus (K^e_2 \lll 2) \oplus K_2 \\ K^e_4 = K^e_2 \oplus (K^e_3 \lll 2) \oplus K_3 \\ \quad\vdots \\ K^e_{r_{in}-1} = K^e_{r_{in}-3} \oplus K_{r_{in}-2} \\ \qquad\qquad \oplus (K^e_{r_{in}-2} \lll 2) \\ K^e_{r_{in}} = K^e_{r_{in}-2} \oplus K_{r_{in}-1} \\ \qquad\qquad \oplus (K^e_{r_{in}-1} \lll 2) \end{cases}$$

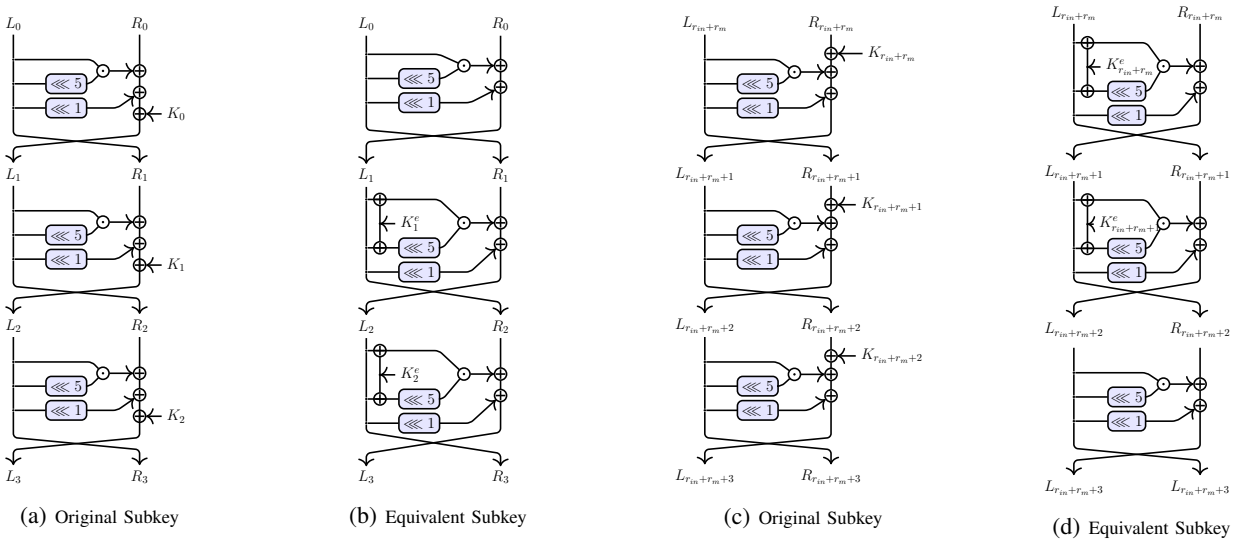
$$\begin{cases} K^e_{r_{tot}-2} = K_{r_{tot}-1} \\ K^e_{r_{tot}-3} = (K^e_{r_{tot}-2} \lll 2) \oplus K_{r_{tot}-2} \\ K^e_{r_{tot}-4} = K^e_{r_{tot}-2} \oplus K_{r_{tot}-3} \\ \qquad\qquad \oplus (K^e_{r_{tot}-3} \lll 2) \\ K^e_{r_{tot}-5} = K^e_{r_{tot}-3} \oplus K_{r_{tot}-4} \\ \qquad\qquad \oplus (K^e_{r_{tot}-4} \lll 2) \\ \quad\vdots \\ K^e_{r_{in}+r_m} = K^e_{r_{in}+r_m+2} \oplus K_{r_{in}+r_m+1} \\ \qquad\qquad \oplus (K^e_{r_{in}+r_m+1} \lll 2) \\ K^e_{r_{in}+r_m-1} = K^e_{r_{in}+r_m+1} \oplus K_{r_{in}+r_m} \\ \qquad\qquad \oplus (K^e_{r_{in}+r_m} \lll 2) \end{cases}$$

APPENDIX C
RELATIONSHIP BETWEEN ORIGINAL SUBKEY AND EQUIVALENT SUBKEY IN SIMECK

The relationships between equivalent subkeys and original subkeys in Simeck Figure 2 are as follows:

$$\begin{cases} K_1^e = K_0 \\ K_2^e = (K_1^e \lll 1) \oplus K_1 \\ K_3^e = K_1^e \oplus (K_2^e \lll 1) \oplus K_2 \\ K_4^e = K_2^e \oplus (K_3^e \lll 1) \oplus K_3 \\ \qquad\qquad \vdots \\ K_{r_{in}-1}^e = K_{r_{in}-3}^e \oplus K_{r_{in}-2} \\ \qquad\qquad\qquad \oplus (K_{r_{in}-2}^e \lll 1) \\ K_{r_{in}}^e = K_{r_{in}-2}^e \oplus K_{r_{in}-1} \\ \qquad\qquad\qquad \oplus (K_{r_{in}-1}^e \lll 1) \end{cases}$$

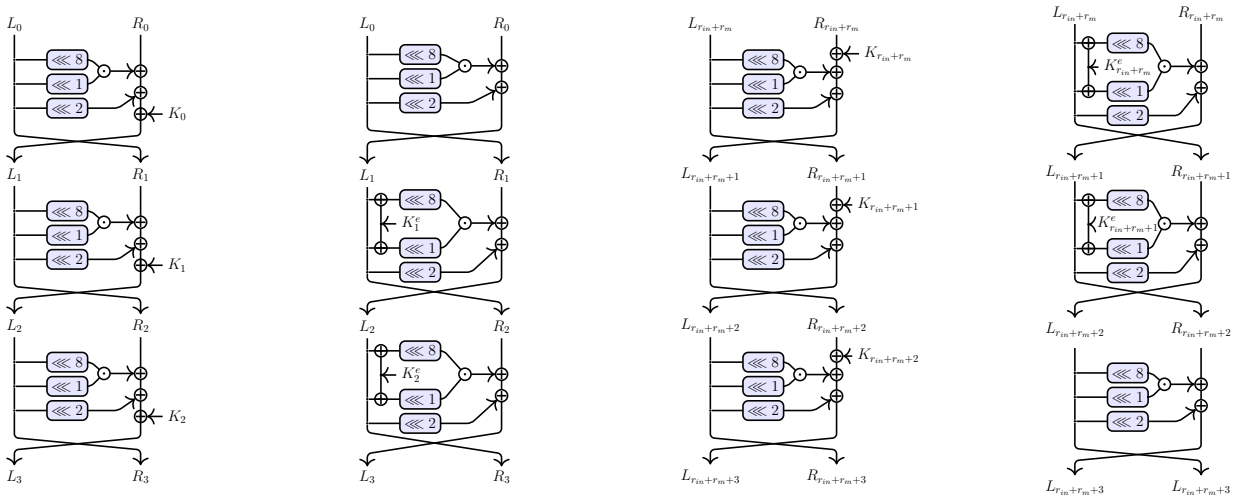
$$\begin{cases} K_{r_{tot}-2}^e = K_{r_{tot}-1} \\ K_{r_{tot}-3}^e = (K_{r_{tot}-2}^e \lll 1) \oplus K_{r_{tot}-2} \\ K_{r_{tot}-4}^e = K_{r_{tot}-2}^e \oplus K_{r_{tot}-3} \\ \qquad\qquad\qquad \oplus (K_{r_{tot}-3}^e \lll 1) \\ K_{r_{tot}-5}^e = K_{r_{tot}-3}^e \oplus K_{r_{tot}-4} \\ \qquad\qquad\qquad \oplus (K_{r_{tot}-4}^e \lll 1) \\ \qquad\qquad \vdots \\ K_{r_{in}+r_m}^e = K_{r_{in}+r_m+2}^e \oplus K_{r_{in}+r_m+1} \\ \qquad\qquad\qquad \oplus (K_{r_{in}+r_m+1}^e \lll 1) \\ K_{r_{in}+r_m-1}^e = K_{r_{in}+r_m+1}^e \oplus K_{r_{in}+r_m} \\ \qquad\qquad\qquad \oplus (K_{r_{in}+r_m}^e \lll 1) \end{cases}$$



(a) Original Subkey (b) Equivalent Subkey (c) Original Subkey (d) Equivalent Subkey

Fig. 2: Relationship between original subkey and equivalent subkey in Simeck

APPENDIX D
NECESSARY ALGORITHMS TO CONSTRUCT OUR AUTOMATED TOOL



(a) Original Subkey (b) Equivalent Subkey (c) Original Subkey (d) Equivalent Subkey

Fig. 3: Relationship between original subkey and equivalent subkey in SIMON

Algorithm 1: $\texttt{CSP}_m$ model of differential propagation through E_m for SIMON

Input: The integer number r_m
Output: $\texttt{CSP}_m$

Declare an empty CSP model $\mathcal{M}$;
/* Declaration of CP variables */
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{x}_{r,0}[i] \in \{0,1\},\ \texttt{x}_{r,1}[i] \in \{0,1\} :\ 0 \le r \le r_m,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{y}_{r,0}[i] \in \{0,1\} :\ 0 \le r \le r_m - 1,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{z}_{r,0}[i] \in \{0,1\} :\ 0 \le r \le r_m - 1,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{p}_{r,0}[i] \in \{0,1\} \in \{0,1\} :\ 0 \le r \le r_m - 1,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow 0 \le \texttt{p} \le (2n-1)$;
/* Rotation and AND operation */
for $r = 0, \ldots, r_m - 1$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{z}_{r,0} = \texttt{x}_{r,0} \lll 2$;
end
for $r = 0, \ldots, r_m - 1$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{Rotation-AND}(\texttt{x}_{r,0}, \texttt{y}_{r,0})$;
end
/* XOR operation */
for $r = 0, \ldots, r_m - 1,\ i = 0, \ldots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{XOR}(\texttt{y}_{r,0}[i], \texttt{z}_{r,0}[i], \texttt{x}_{r,1}[i], \texttt{x}_{r+1,0}[i])$;
end
/* Equality condition */
for $r = 0, \ldots, r_m - 1,\ i = 0, \ldots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{x}_{r,0}[i] = \texttt{x}_{r+1,1}[i])$;
end
/* Probability calculation */
for $r = 0, \ldots, r_m - 1,\ i = 0, \ldots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{AUX}(\texttt{x}_{r,0}[i], \texttt{x}_{r,0}[i+t \pmod n)], \texttt{x}_{r,0}[i+2t \pmod n)], \texttt{p}_{r,0}[i])$;
end
$\mathcal{M}.\texttt{con} \leftarrow \texttt{p} = \sum_{r=0}^{r_m-1} \sum_{i=0}^{n-1} (2 * \texttt{x}_{r,0}[i] - \texttt{p}_{r,0}[i])$;
/* Constraints on input-output difference */
$\mathcal{M}.\texttt{con} \leftarrow \sum_{i=0}^{n-1} (\texttt{x}_{0,0}[i] + \texttt{x}_{0,1}[i])\ \ != 0$;
$\mathcal{M}.\texttt{con} \leftarrow \sum_{i=0}^{n-1} (\texttt{x}_{r_m,0}[i] + \texttt{x}_{r_m,1}[i])\ \ != 0$;
return $\mathcal{M}$;

Algorithm 2: $\texttt{CSP}_{out}^{dp}$ model of deterministic differential transition through E_{out} for SIMON

Input: $\texttt{CSP}_m.\texttt{var}$, and the integer number r_{out}
Output: $\texttt{CSP}_{out}^{dp}$

Declare an empty **CSP** model $\mathcal{M}$;
/* Declaration of CP variables */
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{dx}_{r,0}^{f}[i] \in \{-1,0,1\},\ \texttt{dx}_{r,0}^{f}[i] \in \{-1,0,1\} :\ 0 \le r \le r_{out},\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{dy}_{r,0}^{f}[i] \in \{-1,0,1\} :\ 0 \le r \le r_{out}-1,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{dz}_{r,0}^{f}[i] \in \{-1,0,1\} :\ 0 \le r \le r_{out}-1,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{dw}_{r,0}^{f}[i] \in \{-1,0,1\} :\ 0 \le r \le r_{out}-1,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{dp}_{r,0}^{f}[i] \in \{-1,0,1\},\ \texttt{dq}_{r,0}^{f}[i] \in \{-1,0,1\} :\ 0 \le r \le r_{out}-1,\ 0 \le i \le (n-1)\}$;
/* Boundary conditions */
for $i = 0, \ldots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{dx}_{0,0}^{f}[i] = \texttt{x}_{r_m,0}[i])$;
end
for $i = 0, \ldots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{dx}_{0,1}^{f}[i] = \texttt{x}_{r_m,1}[i])$;
end
/* Rotation */
for $r = 0, \ldots, r_{out}-1$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{dy}_{r,0}^{f} = \texttt{dx}_{r,0}^{f} \lll 8$;
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{dz}_{r,0}^{f} = \texttt{dx}_{r,0}^{f} \lll 1$;
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{dw}_{r,0}^{f} = \texttt{dx}_{r,0}^{f} \lll 2$;
end
/* AND operation */
for $r = 0, \ldots, r_{out}-1,\ i = 0, \ldots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{AND}_{\texttt{D}}(\texttt{dy}_{r,0}^{f}[i], \texttt{dz}_{r,0}^{f}[i], \texttt{dp}_{r,0}^{f}[i])$;
end
/* XOR operation */
for $r = 0, \ldots, r_{out}-1,\ i = 0, \ldots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{XOR}_{\texttt{D}}(\texttt{dp}_{r,0}^{f}[i], \texttt{dw}_{r,0}^{f}[i], \texttt{dq}_{r,0}^{f}[i])$;
end
for $r = 0, \ldots, r_{out}-1,\ i = 0, \ldots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{XOR}_{\texttt{D}}(\texttt{dq}_{r,0}^{f}[i], \texttt{dx}_{r,1}^{f}[i], \texttt{dx}_{r+1,0}^{f}[i])$;
end
/* Equality condition */
for $r = 0, \ldots, r_{out}-1,\ i = 0, \ldots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{dx}_{r,0}^{f}[i] = \texttt{dx}_{r+1,1}^{f}[i])$;
end
return $\mathcal{M}$;

Algorithm 3: $\texttt{CSP}_{out}^{isk}$ model to find k_{out} through E_{out} for SIMON

Input: $\texttt{CSP}_m.\texttt{var}$, $\texttt{CSP}_{out}^{dp}$, and the integer number r_{out}
Output: $\texttt{CSP}_{out}^{isk}$

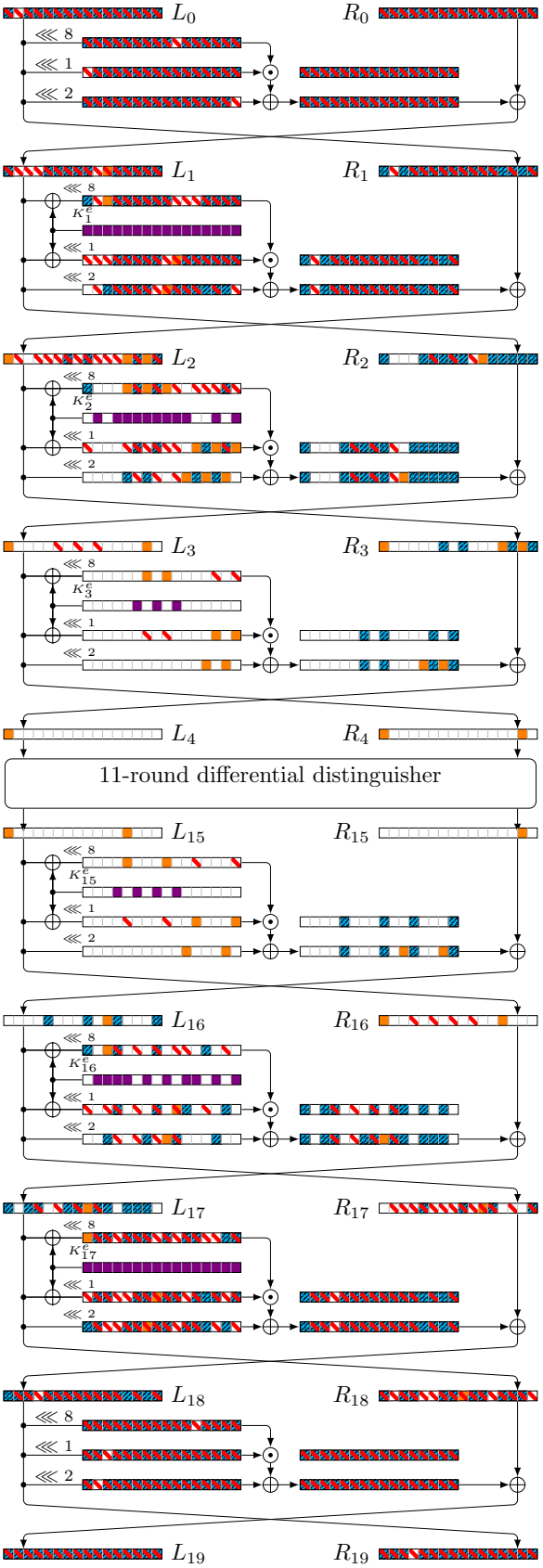
Declare an empty CSP model $\mathcal{M}$;
/* Declaration of CP variables */
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{kdx}_{r,0}^{f}[i] \in \{0,1\},\ \texttt{kdxu}_{r,1}^{f}[i] \in \{0,1\} :\ 0 \le r \le r_{out},\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{kdy}_{r,0}^{f}[i],\ \texttt{kdz}_{r,0}^{f}[i],\ \texttt{kdw}_{r,0}^{f}[i] \in \{0,1\} :\ 0 \le r \le r_{out}-1,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{kdp}_{r,0}^{f}[i],\ \texttt{kdq}_{r,0}^{f}[i] \in \{0,1\} :\ 0 \le r \le r_{out}-1,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{kdy1}_{r,0}^{f}[i],\ \texttt{kdz1}_{r,0}^{f}[i],\ \texttt{kdw1}_{r,0}^{f}[i] \in \{0,1\} :\ 0 \le r \le r_{out}-1,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{kx}_{r,0}^{f}[i] \in \{0,1\},\ \texttt{kx}_{r,1}^{f}[i] \in \{0,1\} :\ 0 \le r \le r_{out},\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{ky}_{r,0}^{f}[i],\ \texttt{kz}_{r,0}^{f}[i],\ \texttt{kw}_{r,0}^{f}[i] \in \{0,1\} :\ 0 \le r \le r_{out}-1,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{kp}_{r,0}^{f}[i] \in \{0,1\},\ \texttt{kq}_{r,0}^{f}[i] \in \{0,1\} :\ 0 \le r \le r_{out}-1,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{ky1}_{r,0}^{f}[i],\ \texttt{kz1}_{r,0}^{f}[i],\ \texttt{kw1}_{r,0}^{f}[i] \in \{0,1\} :\ 0 \le r \le r_{out}-1,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{ikf}_{r}^{f}[i] \in \{0,1\} :\ 0 \le r \le r_{out}-1,\ 0 \le i \le (n-1)\}$;
/* Boundary conditions */
for $i = 0, \dots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{if}\ \texttt{dx}_{0,0}^{f}[i] \ge 0\ \texttt{then}\ \texttt{kdx}_{0,0}^{f}[i] = 0\ \texttt{else}\ \texttt{kdx}_{0,0}^{f}[i] = 1)$;
end
for $i = 0, \dots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{if}\ \texttt{dx}_{0,1}^{f}[i] \ge 0\ \texttt{then}\ \texttt{kdx}_{0,1}^{f}[i] = 0\ \texttt{else}\ \texttt{kdx}_{0,1}^{f}[i] = 1)$;
end
for $i = 0, \dots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{kx}_{0,0}^{f}[i] = 0)\ \cup\ (\texttt{kx}_{0,1}^{f}[i] = 0)$;
end
/* Difference and value determination through XOR operation */
for $r = 0, \dots, r_{out}-1,\ i = 0, \dots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{XOR}_{\texttt{Diff}}(\texttt{dx}_{r+1,0}^{f}[i], \texttt{dq}_{r,0}^{f}[i], \texttt{dx}_{r,1}^{f}[i], \texttt{kdx}_{r+1,0}^{f}[i], \texttt{kdq}_{r,0}^{f}[i], \texttt{kdx}_{r,1}^{f}[i])$;
end
for $r = 0, \dots, r_{out}-1,\ i = 0, \dots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{XOR}_{\texttt{Diff}}(\texttt{dp}_{r,0}^{f}[i], \texttt{dw}_{r,0}^{f}[i], \texttt{dq}_{r,0}^{f}[i], \texttt{kdp}_{r,0}^{f}[i], \texttt{kdw}_{r,0}^{f}[i], \texttt{kdq}_{r,0}^{f}[i])$;
end
for $r = 0, \dots, r_{out}-1,\ i = 0, \dots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{XOR}_{\texttt{Value}}(\texttt{kx}_{r,1}^{f}[i], \texttt{kx}_{r+1,0}^{f}[i], \texttt{kq}_{r,0}^{f}[i])\ \cup\ \texttt{XOR}_{\texttt{Value}}(\texttt{kq}_{r,0}^{f}[i], \texttt{kp}_{r,0}^{f}[i], \texttt{kw}_{r,0}^{f}[i])$;
end
/* Difference and value determination through AND operation */
for $r = 0, \dots, r_{out}-1,\ i = 0, \dots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{AND}_{\texttt{DiffUValue}}(\texttt{kdp}_{r,0}^{f}[i], \texttt{kp}_{r,0}^{f}[i], \texttt{dy}_{r,0}^{f}[i], \texttt{dz}_{r,0}^{f}[i], \texttt{ky}_{r,0}^{f}[i], \texttt{kz}_{r,0}^{f}[i], \texttt{kdy}_{r,0}^{f}[i], \texttt{kdz}_{r,0}^{f}[i])$;
end
/* Difference and value determination through Rotation operation */
for $r = 0, \dots, r_{out}-1$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{kdy1}_{r,0}^{f} = \texttt{kdy}_{r,0}^{f} \ggg 8)\ \cup\ (\texttt{kdz1}_{r,0}^{f} = \texttt{kdz}_{r,0}^{f} \ggg 1)\ \cup\ (\texttt{kdw1}_{r}^{0} = \texttt{kdw}_{r,0}^{f} \ggg 2)$;
end
for $r = 0, \dots, r_{out}-1$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{ky1}_{r,0}^{f} = \texttt{ky}_{r,0}^{f} \ggg 8)\ \cup\ (\texttt{kz1}_{r,0}^{f} = \texttt{kz}_{r,0}^{f} \ggg 1)\ \cup\ (\texttt{kw1}_{r,0}^{f} = \texttt{kw}_{r,0}^{f} \ggg 2)$;
end
for $r = 0, \dots, r_{out}-1,\ i = 0, \dots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{if}\ \texttt{kdy1}_{r,0}^{f}[i] = \texttt{kdz1}_{r,0}^{f}[i] = \texttt{kdw1}_{r,0}^{f}[i] = \texttt{kdx}_{r,0}^{f}[i] = 0\ \texttt{then}\ \texttt{kdx}_{r+1,1}^{f}[i] = 0\ \texttt{else}\ \texttt{kdx}_{r+1,1}^{f}[i] = 1)$;
end
for $r = 0, \dots, r_{out}-1,\ i = 0, \dots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{if}\ \texttt{ky1}_{r,0}^{f}[i] = \texttt{kz1}_{r,0}^{f}[i] = \texttt{kw1}_{r,0}^{f}[i] = \texttt{kx}_{r,0}^{f}[i] = 0\ \texttt{then}\ \texttt{kx}_{r+1,1}^{f}[i] = 0\ \texttt{else}\ \texttt{kx}_{r+1,1}^{f}[i] = 1)$;
end
/* Finding involved subkey bits */
for $r = 0, \dots, r_{out}-1,\ i = 0, \dots (n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{if}\ \texttt{kx}_{r,1}^{f}[i] = 1\ \texttt{then}\ \texttt{ikf}_{r}[i] = 1\ \texttt{else}\ \texttt{ikf}_{r}[i] = 0)$;
end
return $\mathcal{M}$;

Algorithm 4: $\texttt{CSP}_{out}^{eq\cup skg}$ model to find exact k_{out} through E_{out} for SIMON using selective key guessing in equivalent subkey model

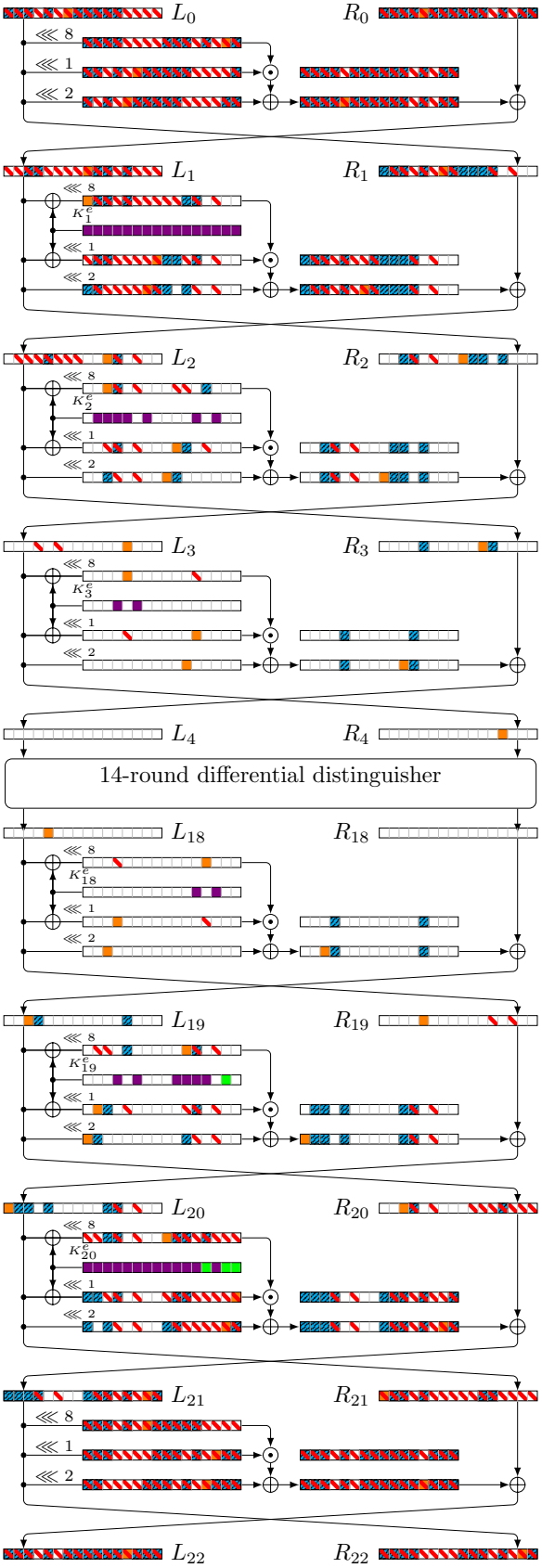
Input: $\texttt{CSP}_m.\texttt{var}$, $\texttt{CSP}_{out}^{dp}$, $\texttt{CSP}_{out}^{isk}$ and the integer number r_{out}
Output: $\texttt{CSP}_{out}^{eq\cup skg}$

Declare an empty **CSP** model $\mathcal{M}$;
/* Declaration of CP variables */
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{ikf}_r^{eq}[i] \in \{0,1\} :\ 0 \le r \le r_{out}-2,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\texttt{ikf}_r^{eq\cup skg}[i] \in \{0,1\} :\ 1 \le r \le r_{out}-2,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\widetilde{\texttt{ky}}_{r,0}^{f}[i],\ \widetilde{\texttt{kz}}_{r,0}^{f}[i] \in \{0,1,2\} :\ 1 \le r \le r_{out}-2,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\widetilde{\texttt{ky1}}_{r,0}^{f}[i],\ \widetilde{\texttt{kz1}}_{r,0}^{f}[i] \in \{0,1,2\} :\ 1 \le r \le r_{out}-2,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\widehat{\texttt{ky}}_{r,0}^{f}[i],\ \widehat{\texttt{kz}}_{r,0}^{f}[i] \in \{0,1\} :\ 1 \le r \le r_{out}-2,\ 0 \le i \le (n-1)\}$;
$\mathcal{M}.\texttt{var} \leftarrow \{\widehat{\texttt{ky1}}_{r,0}^{f}[i],\ \widehat{\texttt{kz1}}_{r,0}^{f}[i] \in \{0,1\} :\ 1 \le r \le r_{out}-2,\ 0 \le i \le (n-1)\}$;
/* Involved key in equivalent subkey model */
for $r = 0,\ldots,r_{out}-2,\ i = 0,\ldots(n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{if } \texttt{ky1}_{r,0}^{f}[i] = \texttt{kz1}_{r,0}^{f}[i] = 0 \texttt{ then } \texttt{ikf}_r^{eq}[i] = 0 \texttt{ else } \texttt{ikf}_r^{eq}[i] = 1)$;
end
/* Updated involved key bits using selective key guessing in equivalent subkey model */
for $r = 1,\ldots,r_{out}-2,\ i = 0,\ldots(n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{SKG}_1(\texttt{dy}_{r,0}^{f}[i], \texttt{dz}_{r,0}^{f}[i], \texttt{ky}_{r,0}^{f}[i], \texttt{kz}_{r,0}^{f}[i], \widetilde{\texttt{ky}}_{r,0}^{f}[i], \widetilde{\texttt{kz}}_{r,0}^{f}[i])$;
end
for $r = 1,\ldots,r_{out}-2$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\widetilde{\texttt{ky1}}_{r,0}^{f} = \widetilde{\texttt{ky}}_{r,0}^{f} \ggg 8)\ \cup\ (\widetilde{\texttt{kz1}}_{r,0}^{f} = \widetilde{\texttt{kz}}_{r,0}^{f} \ggg 1)$;
end
for $r = 1,\ldots,r_{out}-2,\ i = 0,\ldots(n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow \texttt{SKG}_2(\widetilde{\texttt{ky}}_{r,0}^{f}[i], \widetilde{\texttt{kz}}_{r,0}^{f}[i], \widetilde{\texttt{kz1}}_{r,0}^{f}[i+8], \widetilde{\texttt{ky1}}_{r,0}^{f}[i+1], \widehat{\texttt{ky}}_{r,0}^{f}[i], \widehat{\texttt{kz}}_{r,0}^{f}[i])$;
end
for $r = 1,\ldots,r_{out}-2$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\widehat{\texttt{ky1}}_{r,0}^{f} = \widehat{\texttt{ky}}_{r,0}^{f} \ggg 8)\ \cup\ (\widehat{\texttt{kz1}}_{r,0}^{f} = \widehat{\texttt{kz}}_{r,0}^{f} \ggg 1)$;
end
for $r = 1,\ldots,r_{out}-2,\ i = 0,\ldots(n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{if } \widehat{\texttt{ky1}}_{r,0}^{f}[i] = \widehat{\texttt{kz1}}_{r,0}^{f}[i] = 0 \texttt{ then } \texttt{ikf}_r^{eq\cup skg}[i] = 0 \texttt{ else } \texttt{ikf}_r^{eq\cup skg}[i] = 1)$;
end
for $i = 0,\ldots(n-1)$ **do**
  $\mathcal{M}.\texttt{con} \leftarrow (\texttt{ikf}_0^{eq\cup skg}[i] = \texttt{ikf}_0^{eq}[i])$;
end
return $\mathcal{M}$;

APPENDIX E
FIGURES RELATED TO DIFFERENTIAL MITM ATTACKS ON SIMON AND SIMECK

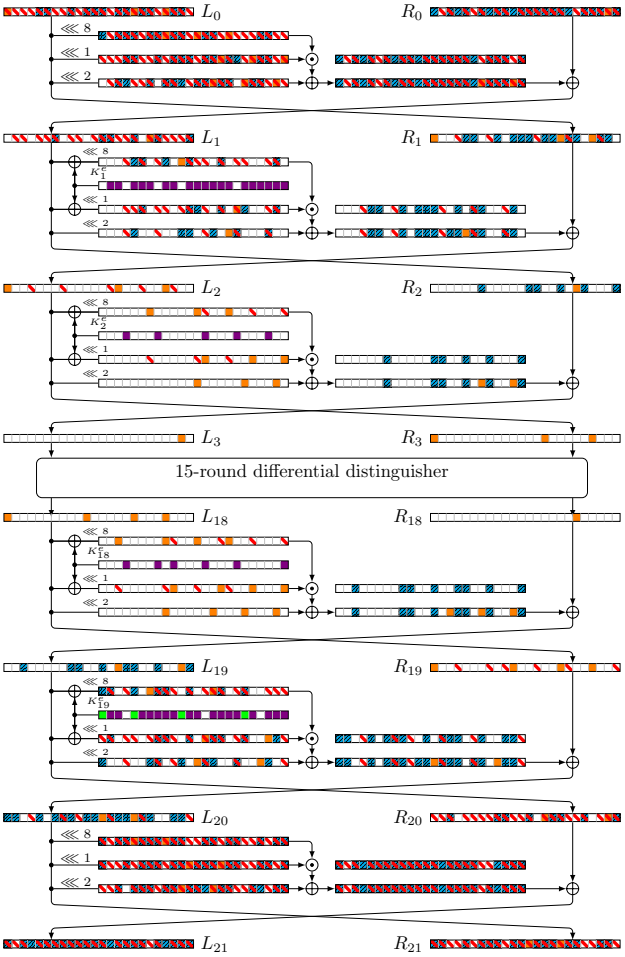


(a) Differential MITM attack on 19-round SIMON-32-64.

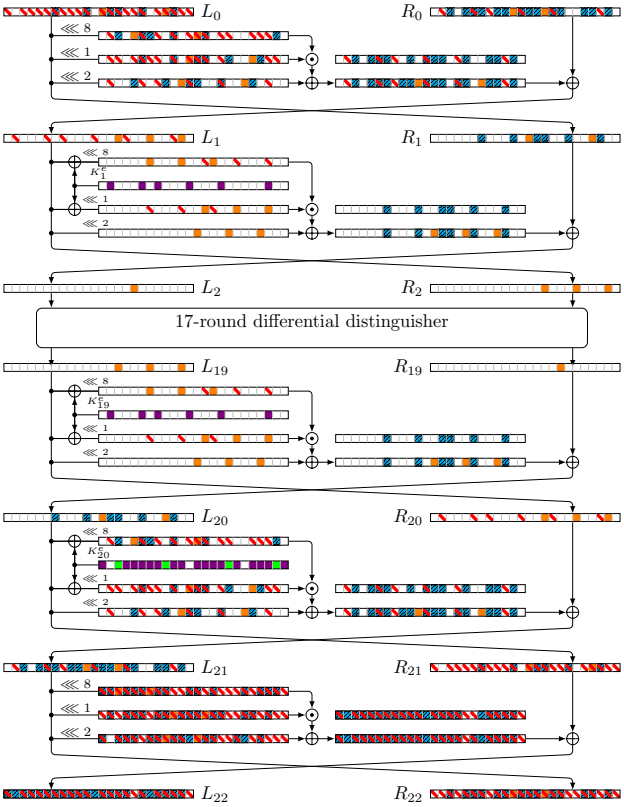


(b) Differential MITM attack on 22-round SIMON-32-64.

Fig. 4: Differential MITM attack on SIMON-32-64

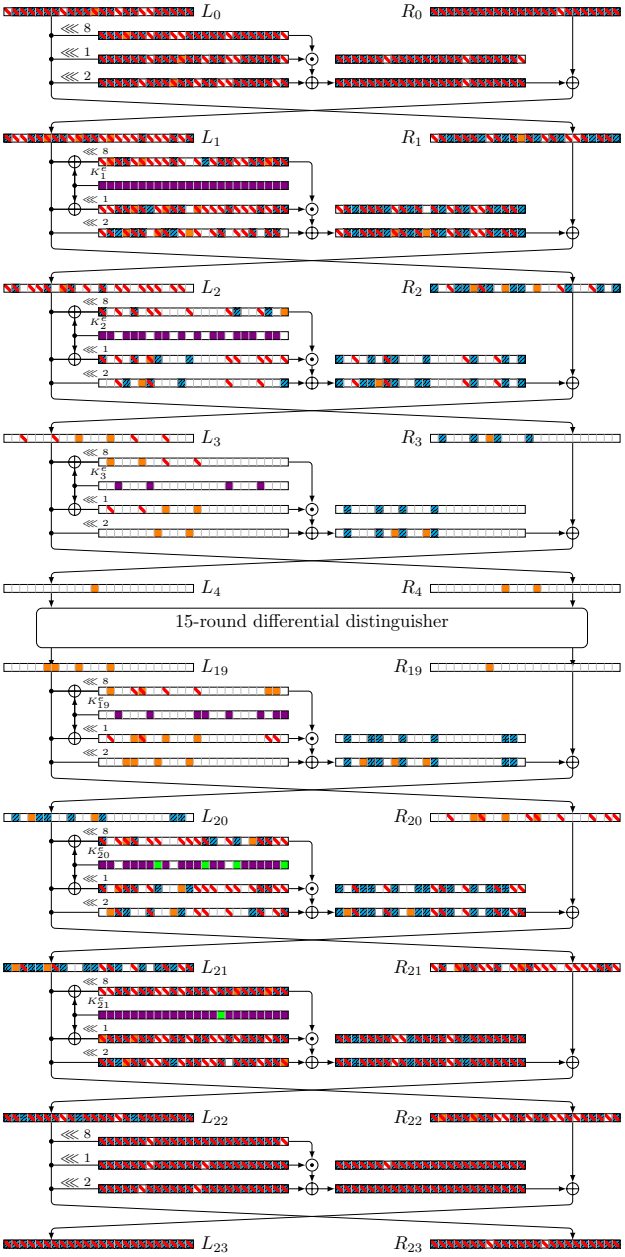


(a) Differential MITM attack on 21-round SIMON-48-72.

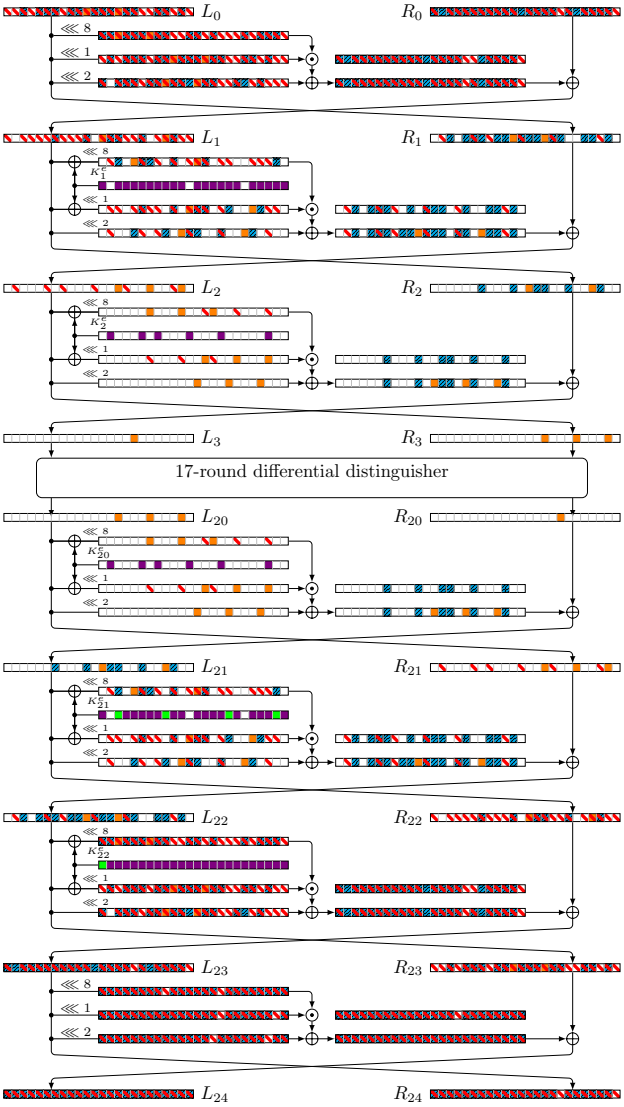


(b) Differential MITM attack on 22-round SIMON-48-72.

Fig. 5: Differential MITM attack on SIMON-48-72

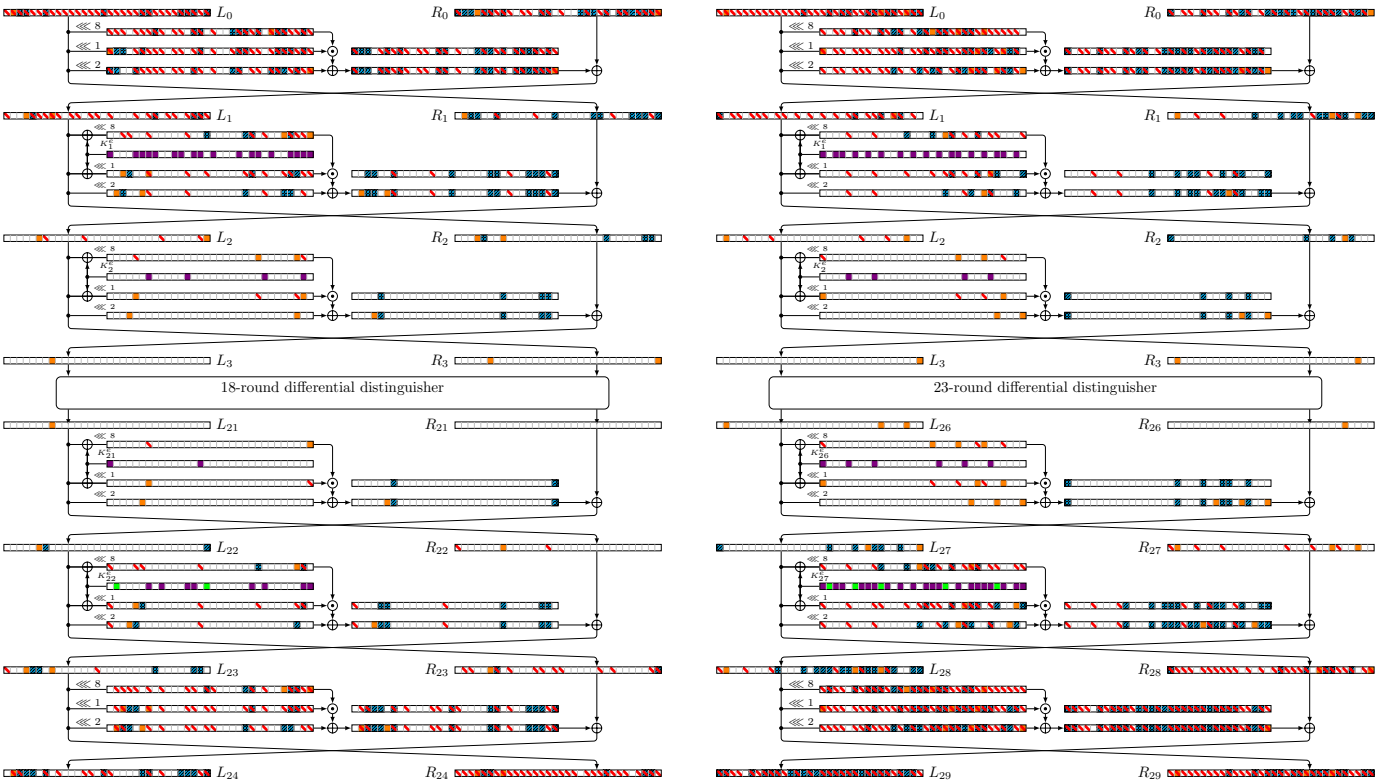


(a) Differential MITM attack on 23-round SIMON-48-96.



(b) Differential MITM attack on 24-round SIMON-48-96.

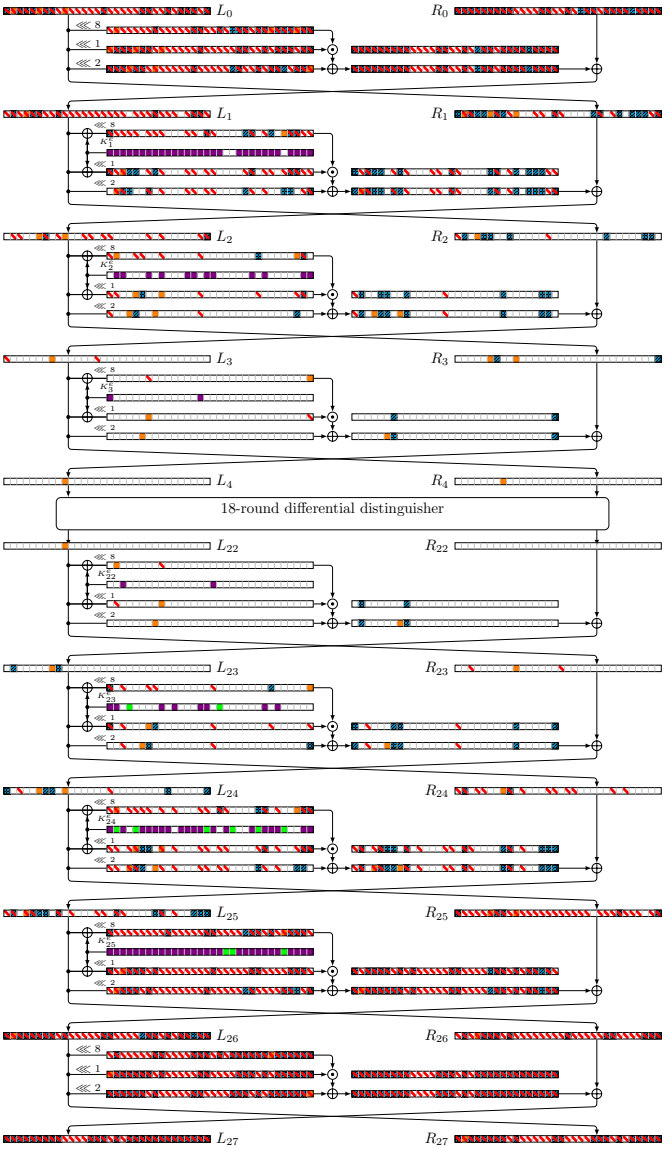
Fig. 6: Differential MITM attack on SIMON-48-96



(a) Differential MITM attack on 24-round SIMON-64-96.

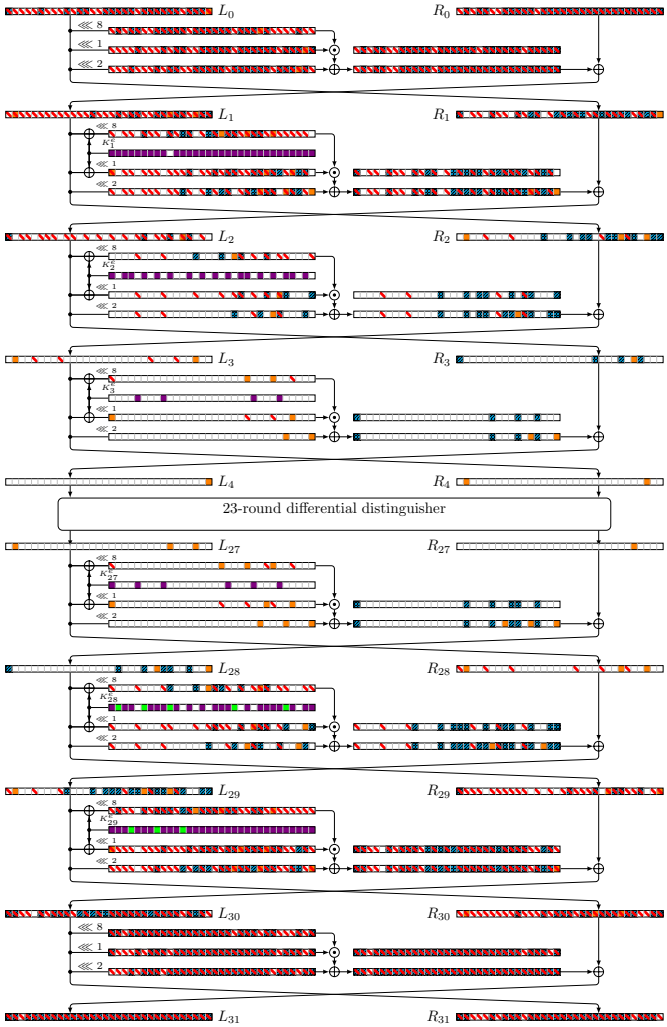
(b) Differential MITM attack on 29-round SIMON-64-96.

Fig. 7: Differential MITM attack on SIMON-64-96



Difference required Value required 1 any
Key bits skipped by SKG technique Involved subkey bits

(a) Differential MITM attack on 27-round SIMON-64-128.



Difference required Value required 1 any
Key bits skipped by SKG technique Involved subkey bits

(b) Differential MITM attack on 31-round SIMON-64-128.

Fig. 8: Differential MITM attack on SIMON-64-128

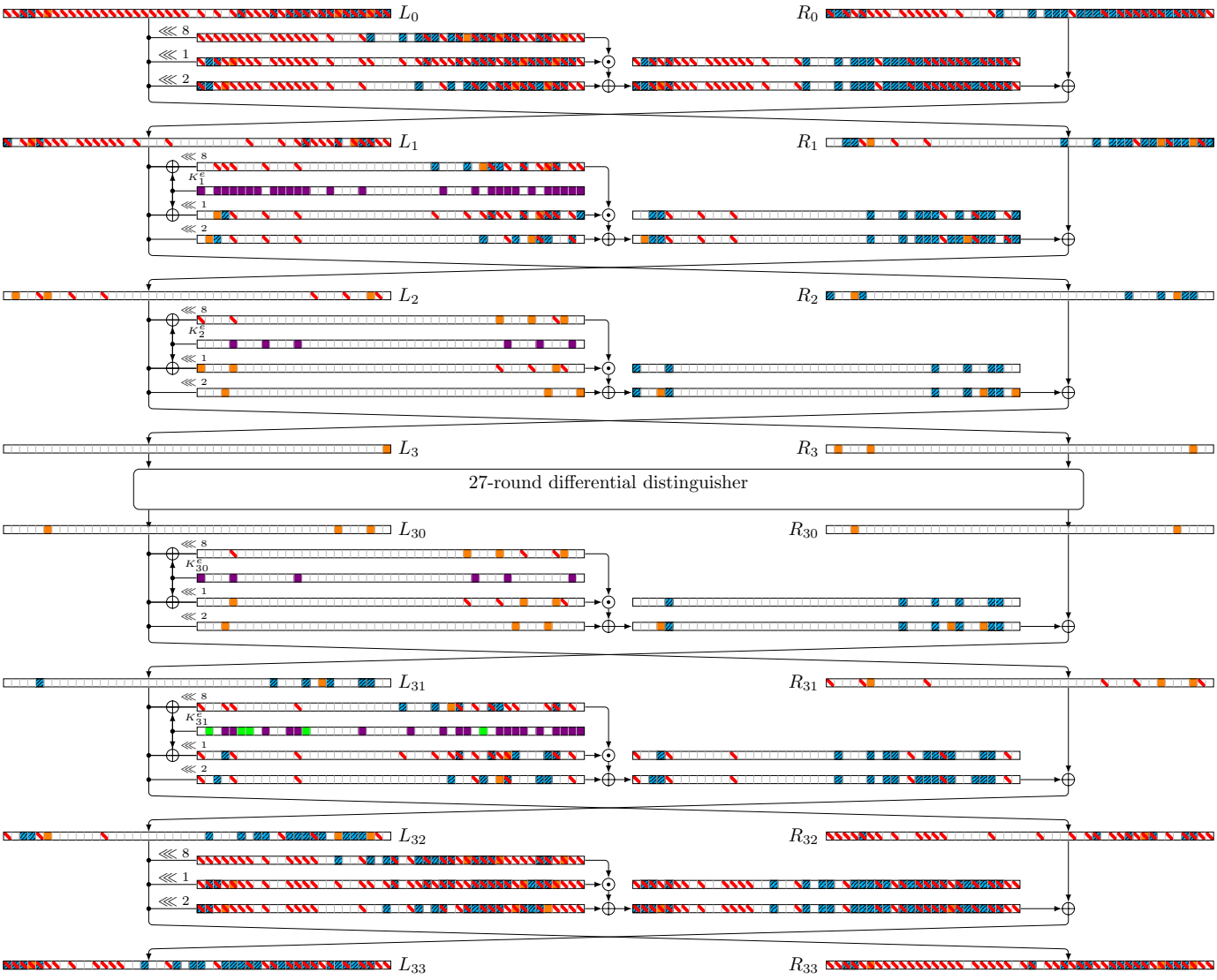


Fig. 9: Differential MITM attack on 33-round SIMON-96-144.

Fig. 10: Differential MITM attack on 30-round SIMON-96-96.

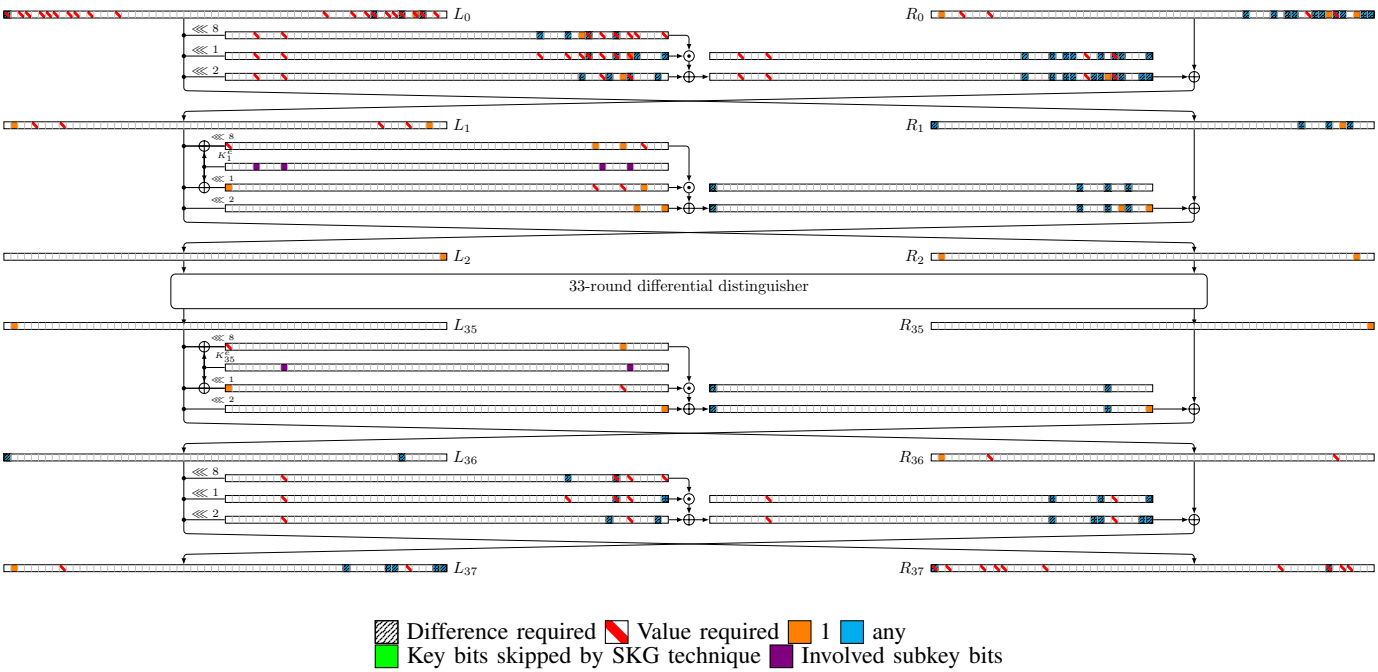


Fig. 11: Differential MITM attack on 37-round SIMON-128-128.

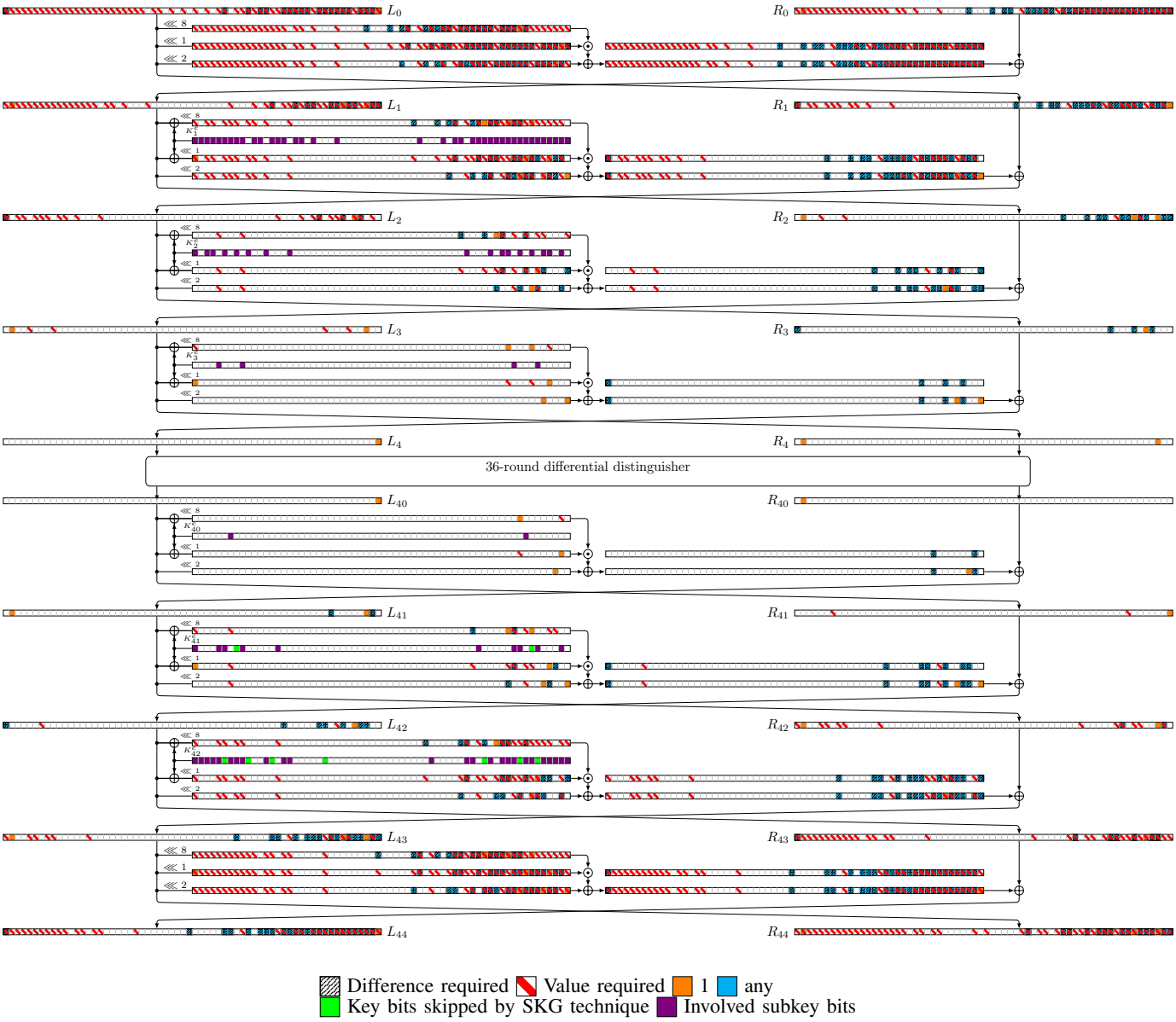


Fig. 12: Differential MITM attack on 44-round SIMON-128-192.

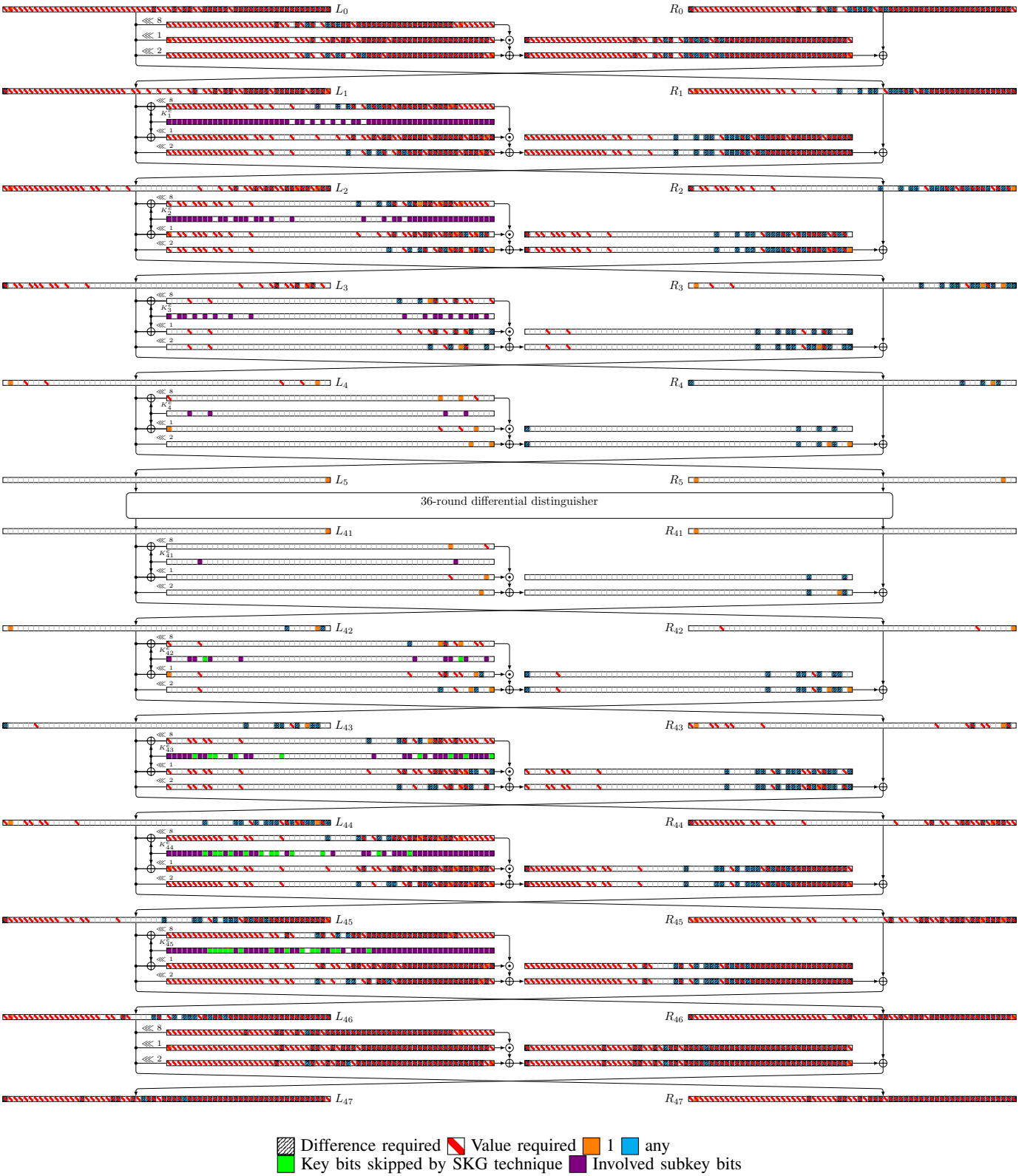


Fig. 13: Differential MITM attack on 47-round SIMON-128-256.

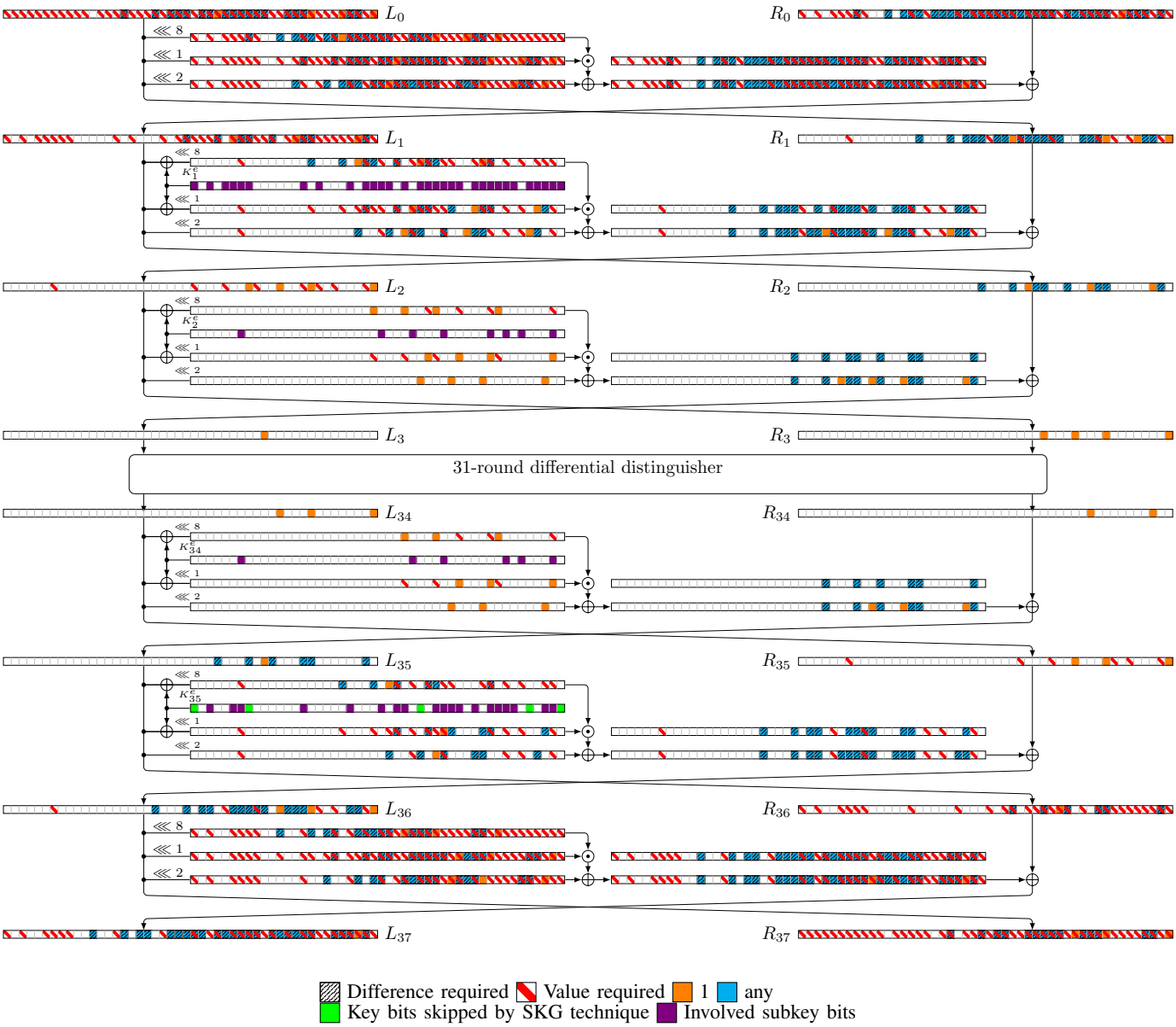


Fig. 14: Differential MITM attack on 37-round SIMON-96-144.

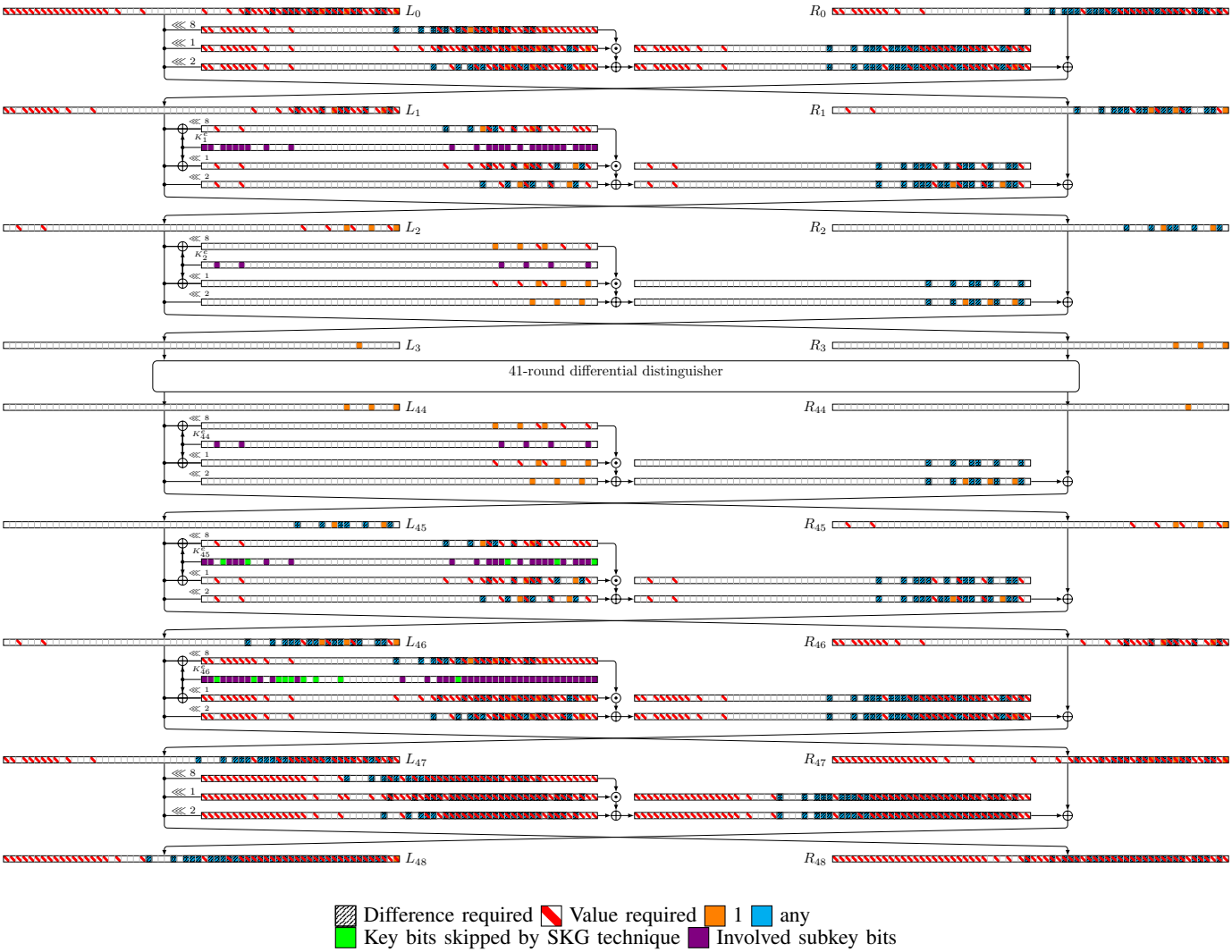


Fig. 15: Differential MITM attack on 48-round SIMON-128-192.

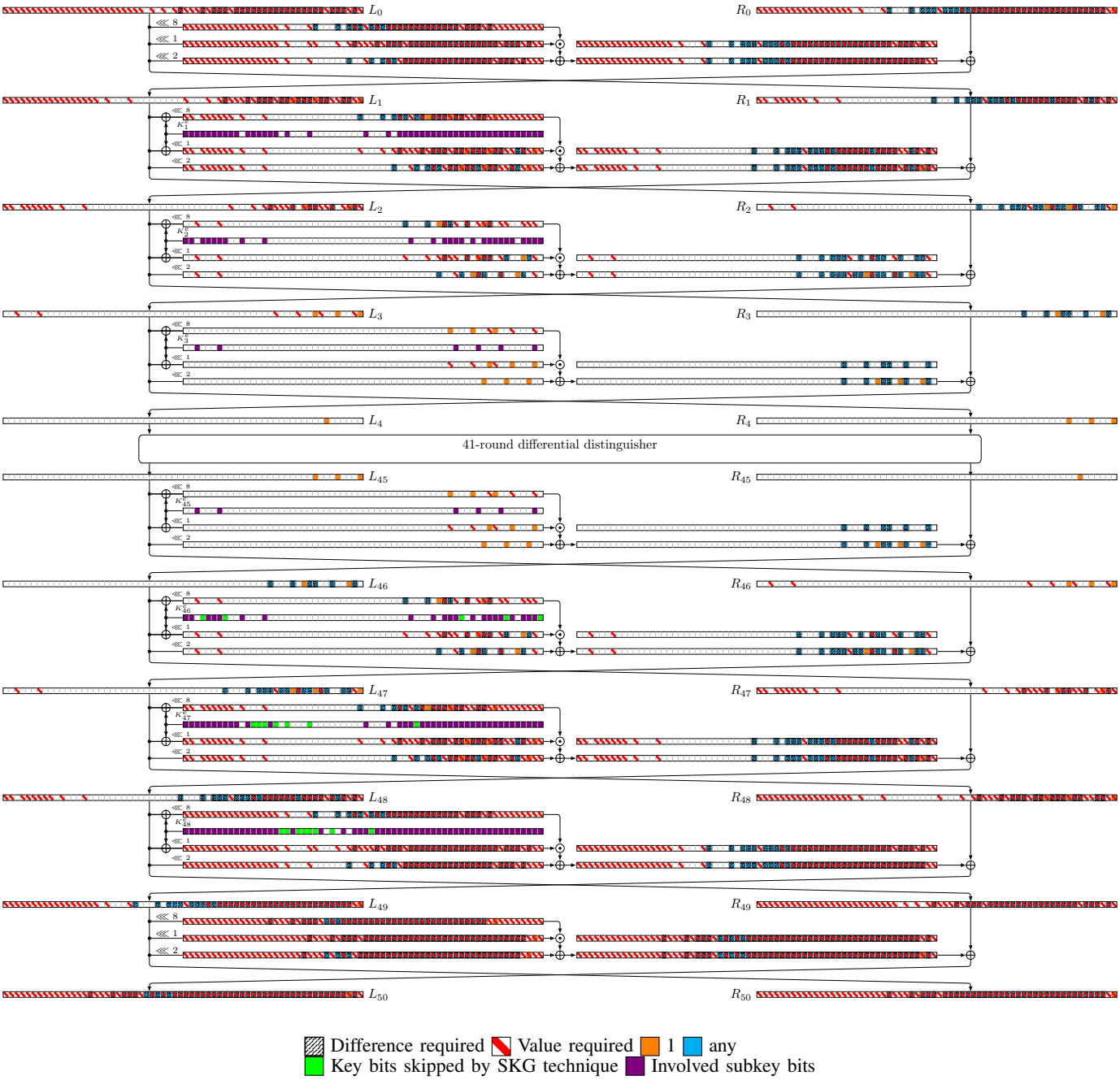
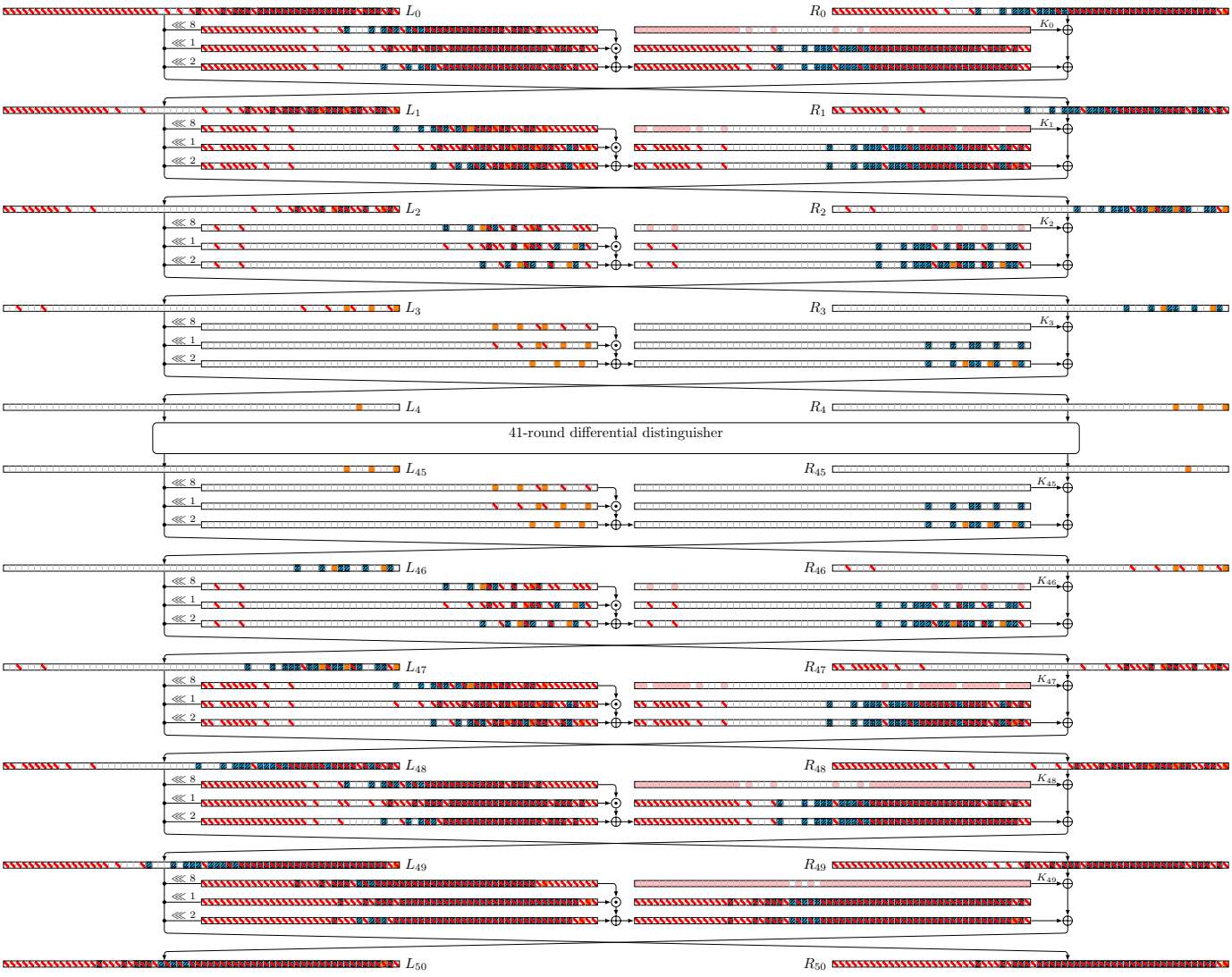


Fig. 16: Differential MITM attack on 50-round SIMON-128-256.



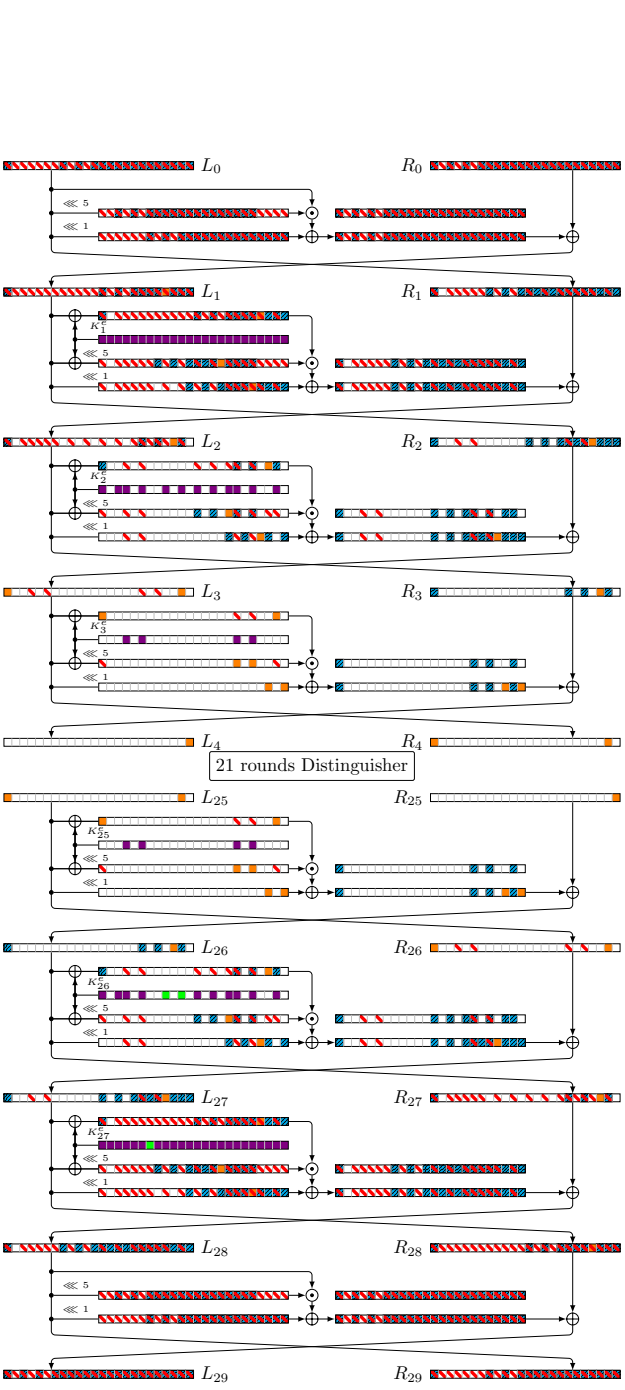
Difference required Value required 1 any Involved subkey bits

Fig. 17: Differential MITM attack on 50-round SIMON-128-256 original subkey.

(a) Differential MITM attack on 21-round Simeck-32-64.

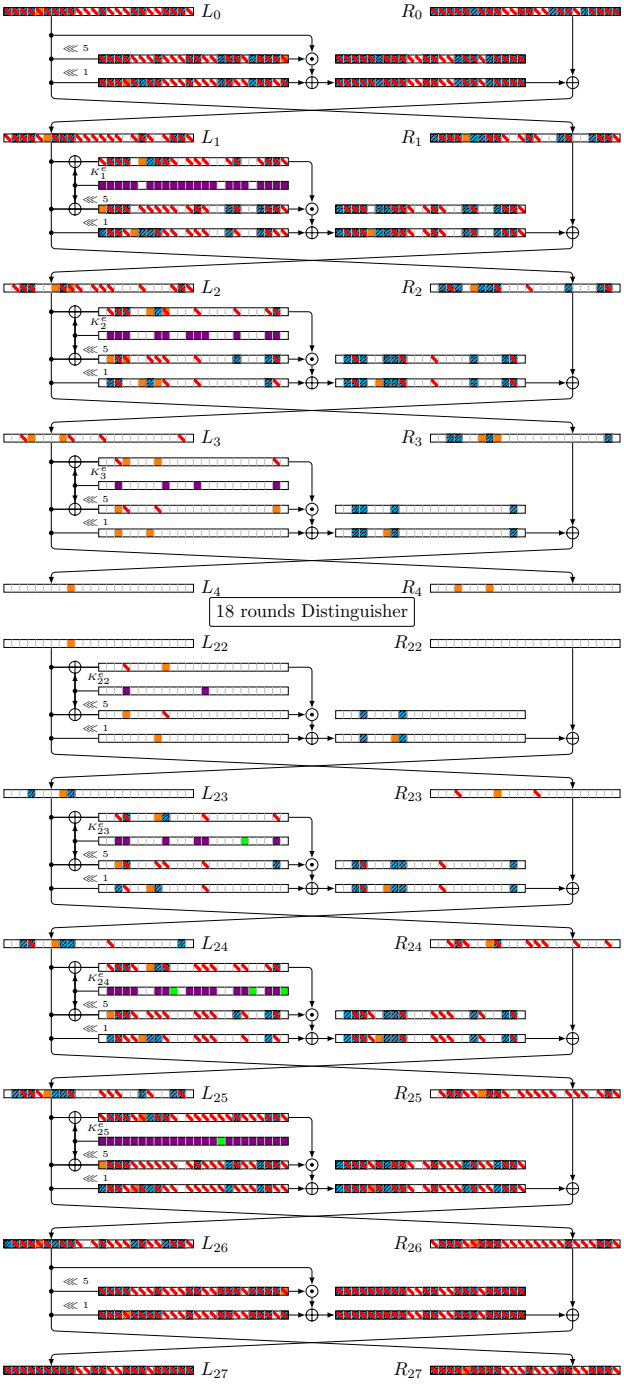
(b) Differential MITM attack on 20-round Simeck-32-64.

Fig. 18: Differential MITM attack on Simeck-32-64



Difference required Value required 1 any
Key bits skipped by SKG technique Involved subkey bits

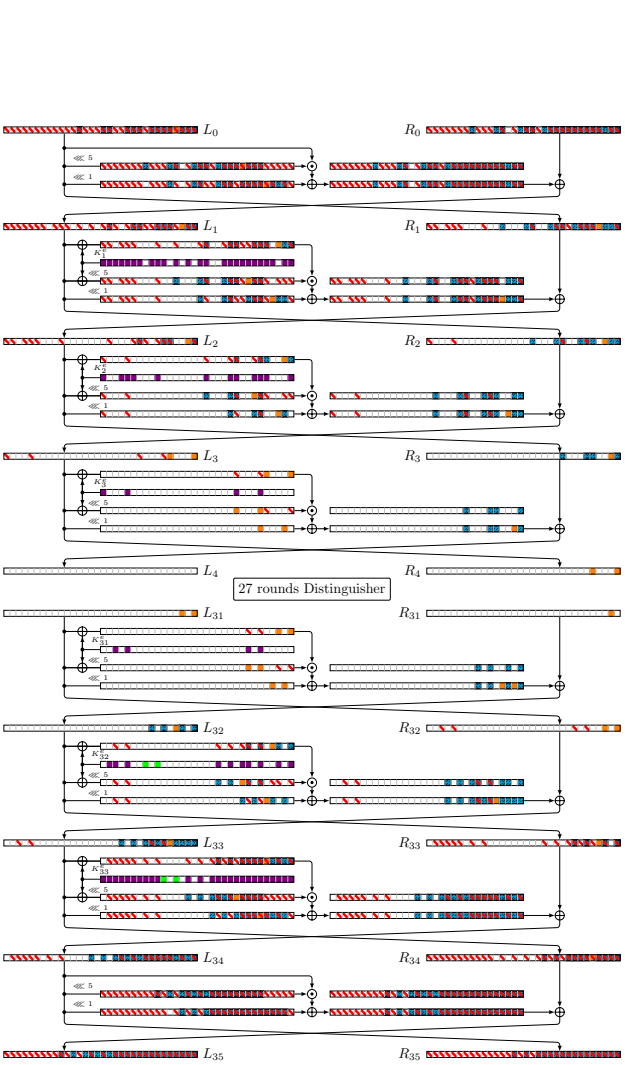
(a) Differential MITM attack on 29-round Simeck-48-96.



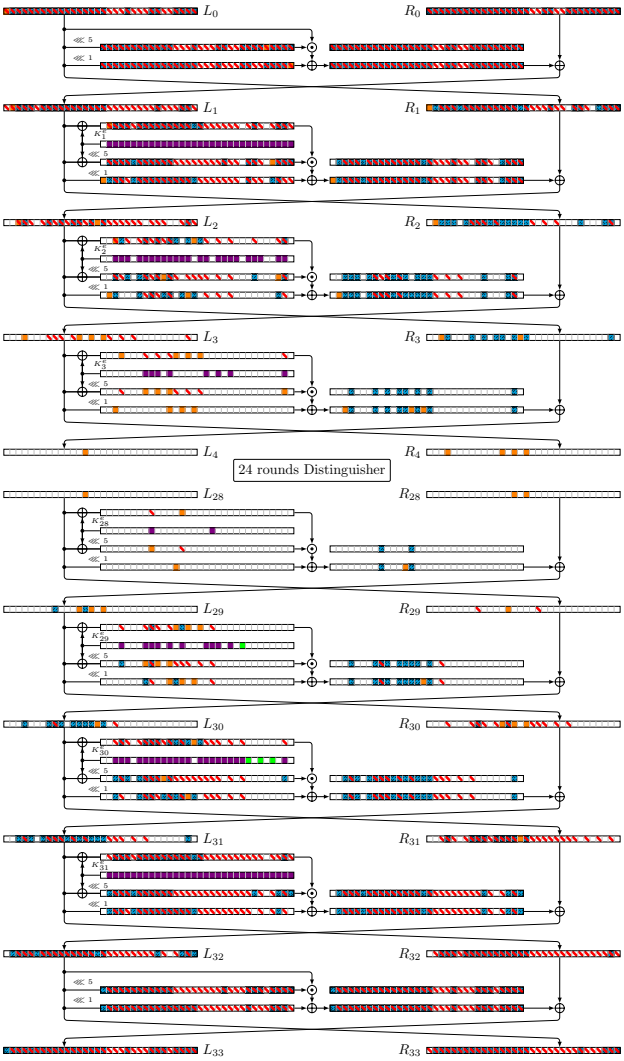
Difference required Value required 1 any
Key bits skipped by SKG technique Involved subkey bits

(b) Differential MITM attack on 27-round Simeck-48-96.

Fig. 19: Differential MITM attack on Simeck-48-96



(a) Differential MITM attack on 35-round Simeck-64-128.



(b) Differential MITM attack on 33-round Simeck-64-128.

Fig. 20: Differential MITM attack on Simeck-64-128

REFERENCES

[1] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK lightweight block ciphers. In *DAC 2015*, pages 175:1–175:6. ACM, 2015.

[2] Gangqiang Yang, Bo Zhu, Valentin Suder, Mark D. Aagaard, and Guang Gong. The simeck family of lightweight block ciphers. In *CHES 2015*, volume 9293 of *LNCS*, pages 307–329. Springer, 2015.