Deep Learning-Based Rotational-XOR Distinguishers for AND-RX Block Ciphers: Evaluations on Simeck and Simon

AMIRHOSSEIN EBRAHIMI[1], DAVID GERAULT[2], PAOLO PALMIERI[1]

1. School of Computer Science & IT, University College Cork
2. Cryptography Research Centre, Technology Innovation Institute, Abu Dhabi, UAE

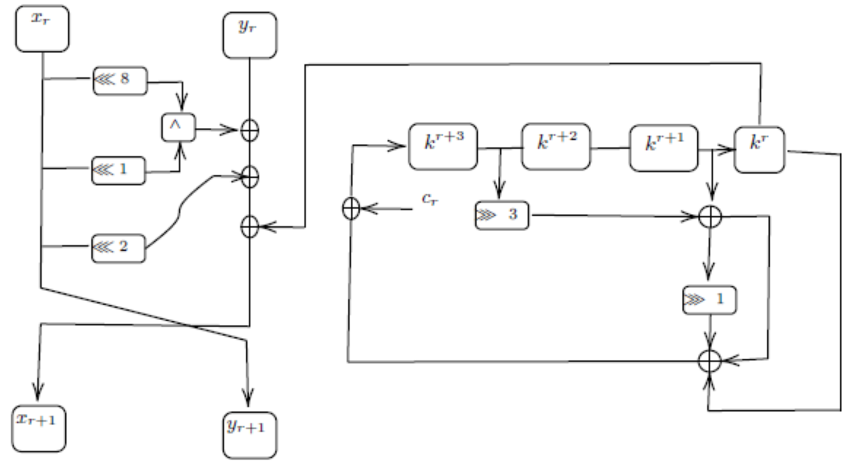
Outline

1. Background and Context
2. Preliminaries
3. Problem Statement and Objectives
4. Methodology
5. Results
6. Discussion and Implications
7. Future Work
8. Conclusion

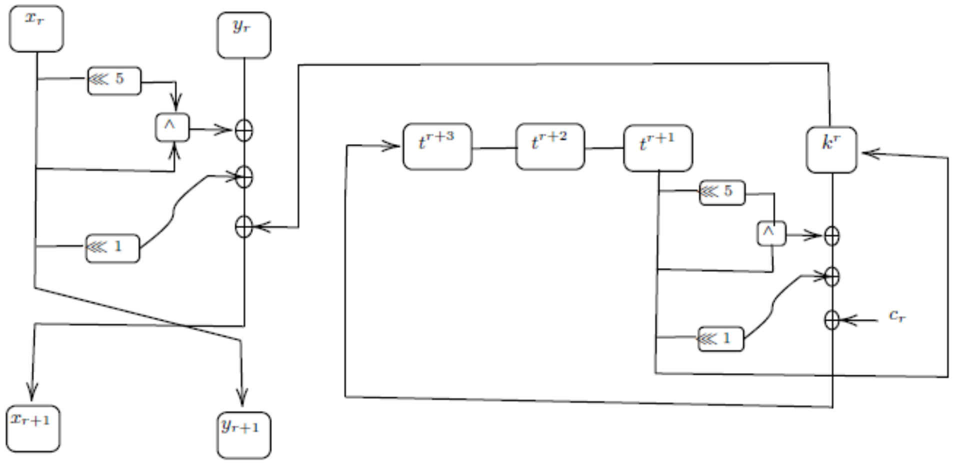
Prior Work and Contextual Landscape

- **2013 - 2015**: Simon & Simeck ciphers
- **2016**: Introducing the Rotational-Xor (RX) cryptanalysis
- **2017 - 2018**: First automated attacks of Simon and Simeck
- **2019**: Gohr's paper on Deep learning based cryptanalysis
- **2021**: Interpretation of neural network distinguishers
- **2022**: rotational-xor cryptanalysis of simon-like block ciphers
- **2022**: automated finding of good input differences

AND-RX Cipher (Simon-like)



Simon cipher for $m = 4$



Simeck cipher

$$R(x, y) = (y \oplus f(x) \oplus k, x)$$

$$f(x) = ((x \lll a) \wedge (x \lll b)) \oplus x \lll c$$

RX Cryptanalysis

- Technique to analyze the security of symmetric algorithms
- Focuses on rotational pairs of plaintext and ciphertext
- Extension of rotational cryptanalysis to handle XOR operations with constants

(δ,γ)-Rotational-Xor-difference: represents a rotational pair with rotation γ under translation δ, i.e., $(x, (x \ll \gamma) \oplus \delta)$
The method seeks to analyze the propagation of RX-differences through the cryptographic primitive.

Deep Learning in Symmetric Cryptography

- Gohr applied deep learning techniques for cryptanalysis, specifically for attacking the Speck cipher
- Deep learning can outperform traditional counterparts in differential cryptanalysis

Algorithm 1 DL-based Differential Distinguisher for r rounds of Speck32/64

1: **Input:** r (number of rounds), AI machine, (C_0, C_1)
2: **Output:** Trained AI machine, differential distinguisher status
3: Generate 10^7 plaintext pairs (P_0, P_1) with $\Delta = (L_0 \oplus L_1, R_0 \oplus R_1) = (0x0040, 0x0000)$
4: Randomly allocate 10^7 labels $Y \in_r \{0, 1\}$ to the pairs
5: **for** each pair (P_0, P_1) with label Y **do**
6:     **if** $Y = 0$ **then**
7:         $P_1 \leftarrow P_1 \in_r \{0, 1\}^{32}$
8:     Encrypt the pairs with r rounds of Speck32/64 to get ciphertext pairs (C_0, C_1)
9:     Store (C_0, C_1) with corresponding labels in a dataset
10: Train DL-distinguisher using the dataset and their corresponding labels
11: Repeat steps 3-11 for another 10^6 pairs for testing
12: Measure the accuracy of the DL-based distinguisher
13: **if** accuracy $> 50\%$ **then**
14:     The machine is a DL-based differential distinguisher

Deep Learning in Symmetric Cryptography (Cont.)

- Bellini et al. presented an alternative approach for finding the best input difference that does not rely on neural networks

Algorithm 2 Evolutionary optimizer [5]

$init\_population \leftarrow$ [RandomInt$(0, 2^n - 1)$ for 1024 times]
Sort $init\_population$ by $\tilde{b}_t(\cdot)$ in descending order
$curr\_population \leftarrow$ first P elements of $init\_population$
for $iter \leftarrow 0$ to 50 **do**
  $cand \leftarrow [\ ]$
  for $i \leftarrow 0$ to $P-1$ **do**
    for $j \leftarrow i+1$ to $P-1$ **do**
      if RandomFloat$(0,1) < p_m$ **then**
        $m \leftarrow 1$
      else
        $m \leftarrow 0$
      Add $curr\_population_i \oplus curr\_population_j \oplus (m \lll \text{RandomInt}(0, n-1))$ to $cand$
  Sort $cand$ by $\tilde{b}_t(\cdot)$ in descending order
  $curr\_population \leftarrow$ first P elements of $cand$
return $cand$

$$\tilde{b}^t(\Delta) = \left| \sum_{j=0}^{n-1} 2 \cdot \frac{\sum_{i=0}^{t}(E_{K_i}(X_i) \oplus E_{K_i}(X_i \oplus \Delta))_j}{t} - 1 \right|$$

Challenges in RX Cryptanalysis

- More parameters
- Limitations of weak-key models
- Unexplored potential of AI

DL-based RX Distinguisher & Approximate RX Bias Score

- **DL-based RX Distinguisher:** An AI machine is trained to distinguish rotational-XOR (RX) patterns in ciphertext pairs. The machine's accuracy is measured and if it's greater than or equal to 50%, it is designated as an RX distinguisher.

Approximate RX Bias Score: Let $E: \mathbb{F}_2^n \times \mathbb{F}_2^k \rightarrow \mathbb{F}_2^n$ be a block cipher, then The Approximate RX Bias Score for δ, denoted by $\tilde{b}^t(\delta, \gamma)$, is defined as the sum of the biases of each bit position j in the output RX-difference, computed over t samples.

$$\tilde{b}^t(\delta,\gamma) = \left| \sum_{j=0}^{n-1} 2 \cdot \frac{\sum_{i=0}^{t}((E_{K_i}(X_i)) \oplus E_{(K_i \lll \gamma)\oplus\delta}((X_i \lll \gamma) \oplus \delta)))_j}{t} - 1 \right|$$

DL-based RX Distinguisher & Approximate RX Bias Score (Cont.)

- We observed a positive correlation between bias score and the accuracy of the distinguisher, with some outliers due to variations in the cipher structure.

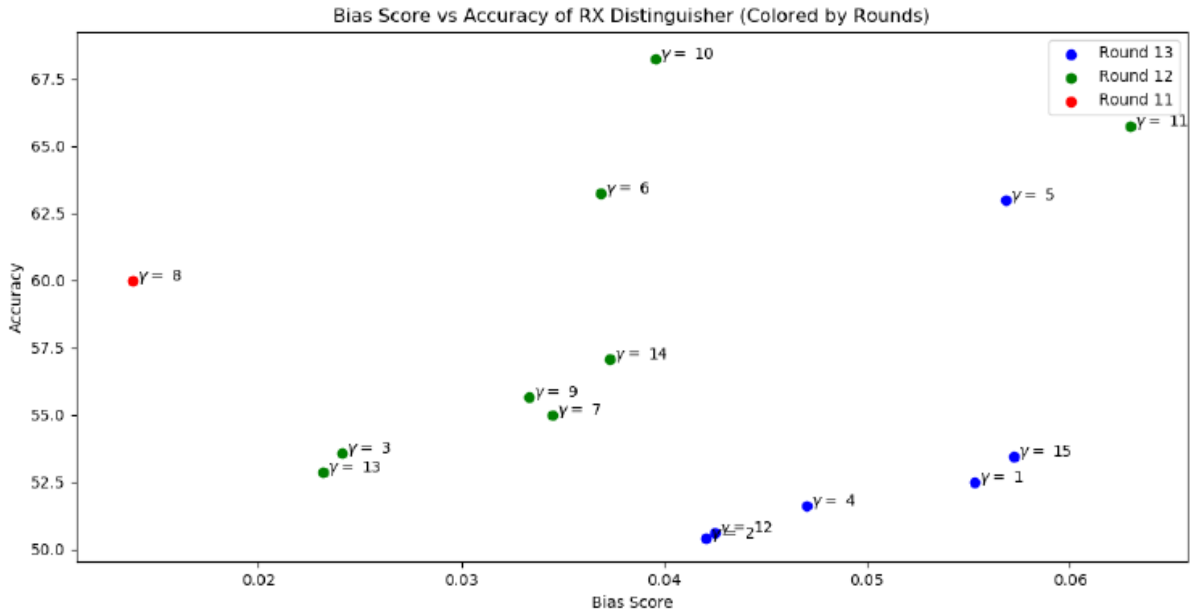


Fig. 2: Scatter plot of Bias Score vs Accuracy of RX Distinguisher (Colored by Rounds)

Evolutionary Optimization of Deep Learning RX Differential Distinguishers

- **Goal:** To adapt the evolutionary-based search algorithm to explore a more extensive set of candidate RX pairs, accounting for the rotational offset γ and the XOR translation δ.
- **Search Strategy:** First-time simultaneous search for optimal δ and γ parameters.
- **Methodology:** The algorithm starts with a population of randomly generated input differences and corresponding rotational offsets. For each of them, an approximate RX bias score is computed. The algorithm iteratively refines this population, maintaining top performers.
- **Outcome:** The algorithm returns a list of 32 input differences for each round and computes a weighted cumulative RX bias score.
- **Advancement:** This approach allows for identifying effective RX distinguishers for full-key classes, a notable improvement over previous methods.

Evolutionary Optimization of Deep Learning RX Differential Distinguishers (Cont.)

Algorithm 4 Evolutionary optimizer for RX differential distinguishers

$init\_population \leftarrow$ [RandomInt$(0, 2^n - 1)$||RandomInt$(1, n-1)$ for 1024 times]
Sort $init\_population$ by $\tilde{b}^t_{(\delta,\gamma)}(\cdot)$ in descending order
$curr\_population \leftarrow$ first P elements of $init\_population$
for $iter \leftarrow 0$ to 50 **do**
    $cand \leftarrow [\ \ ]$
    for $i \leftarrow 0$ to $P-1$ **do**
        for $j \leftarrow i+1$ to $P-1$ **do**
            $m_\gamma \leftarrow 1$
            if RandomFloat$(0,1) < p_m$ **then**
                $m_\delta \leftarrow 1$
            else
                $m_\delta \leftarrow 0$
            Add $(((curr\_population_{i,\delta} \lll \text{RandomInt}(0, n-1)) \oplus curr\_population_{j,\delta} \oplus m_\delta)||(curr\_population_{i,\gamma} \oplus m_\gamma)$ to $cand$
    Sort $cand$ by $\tilde{b}_t(\cdot)$ in descending order
    $curr\_population \leftarrow$ first P elements of $cand$
return $cand$

Results

Table 1: Comparison of related-key DL-based distinguishers for Simeck. RX: Rotational-Xor cryptanalysis, RD: Related-key Differential cryptanalysis. The Combined Accuracy Score [11] for m pairs is $\frac{1}{1+\prod_{i=1}^{m}\frac{1-p_i}{p_i}}$

	Round	Combined Accuracy Score	Pairs	Attack Type	Ref.
Simeck 32/64	13	0.9950	8	RD	[17]
	14	0.6679	8	RD	[17]
	15	0.5573	8	RD	[17]
	15	**0.5134**	**1**	**RX**	**This Work**
	15	**0.5475**	**8**	**RX**	**This Work**
Simeck 64/128	18	0.9066	8	RD	[17]
	19	0.7558	8	RD	[17]
	20	0.6229	8	RD	[17]
	20	**0.5212**	**1**	**RX**	**This Work**
	20	**0.6338**	**8**	**RX**	**This Work**

Table 2: Comparison of the RX distinguishers for different versions of Simeck

Cipher	Rounds	Data Complexity	Size of Weak Key Class	DL-based	Ref.
Simeck32/64	**15**	$\mathbf{2^{20}}$	**Full**	**Yes**	**This Work**
	15	2^{18}	2^{44}	No	[18]
	19	2^{24}	2^{30}	No	[18]
	20	2^{26}	2^{30}	No	[18]
Simeck48/96	**17**	$\mathbf{2^{20}}$	**Full**	**Yes**	**This Work**
	16	2^{18}	2^{68}	No	[18]
	18	2^{22}	2^{66}	No	[18]
	19	2^{24}	2^{62}	No	[18]
	27	2^{44}	2^{46}	No	[18]
Simeck64/128	**20**	$\mathbf{2^{20}}$	**Full**	**Yes**	**This Work**
	25	2^{34}	2^{80}	No	[18]
	34	2^{56}	2^{58}	No	[18]

17. Lu, J., Liu, G., Sun, B., Li, C., Liu, L.: Improved (related-key) differential-based neural distinguishers for simon and simeck block ciphers. Cryptology ePrint Archive (2022)

18. Lu, J., Liu, Y., Ashur, T., Sun, B., Li, C.: Improved rotational-xor cryptanalysis of simon-like block ciphers. IET Information Security 16(4), 282–300 (2022)

Results (Cont.)

Table 3: Summary of the optimal RX input differences (δ), key differences (δ_{key}), rotational offsets (γ), the number of rounds, and distinguisher accuracy for different versions of Simeck block ciphers.

Cipher Version	δ	δ_{key}	γ	Number of Rounds	Accuracy
Simeck32/64	$(0, 0x0002)$	0002	1	15	51.34
				14	57.08
				13	70.57
Simeck48/96	$(0, 0x000002)$	0002	1	17	52.06
				16	57.67
				15	69.85
Simeck64/128	$(0, 0x00000002)$	0002	1	20	52.12
				19	57.01
				18	70.15

Table 4: Summary of the optimal RX input differences (δ), key differences (δ_k), rotational offsets (γ), the number of rounds, and distinguisher accuracy for different versions of Simon block ciphers.

Cipher Version	δ	δ_k	γ	Number of Rounds	Accuracy
Simon32/64	$(0x0, 0x0002)$	0002	3	11	54.45
				10	74.11
				9	98.48
Simon64/128	$(0x0, 0x0)$	0000	30	13	51.51
				12	73.15
				11	98.5
Simon128/256	$(0x0, 0x0)$	0000	60	16	50.62
				15	72.26
				14	96.87

Impact of Diffusion Layer

- **Goal:** Improve AND-RX cipher security via ideal shift parameters (a, b, c) in $f(x) = ((x \ll a) \wedge (x \ll b)) \oplus x \ll c$.

- **Approach:** Used evolutionary algorithm to test various (a, b, c) combinations, finding highest bias scores and optimal resistance against attacks.

- **Result:** The shift set (4, 6, 3) offers superior resistance, with no distinguisher found for > 13 rounds.

- **Implication:** These configurations impact cipher's resilience against attacks, but hardware efficiency must also be considered in design.

Impact of Diffusion Layer (Cont.)

Table 5: Optimal rotation sets for AND-RX ciphers with non-linear key schedule and $n = 32$ determined by the evolutionary algorithm

Rotation Set	Highest Cumulative Bias	Highest Round Distinguisher
(4, 6, 3)	14.32	13
(4, 5, 7)	17.28	14
(6, 7, 4)	17.99	14
(3, 7, 2)	18.25	14
(3, 5, 6)	18.67	14
(3, 6, 1)	18.95	14

Conclusion & Future works

- **Summary:**
  - Explored deep learning's role in RX cryptanalysis of AND-RX ciphers, especially Simon and Simeck families.
  - Discovered related-key model distinguishers against conventional weak-key models.
  - Identified optimal RX input differences, key differences, and rotational offsets.
  - Presented a new method to optimize diffusion layers in AND-RX ciphers and identified several optimal rotation sets for Simeck-like ciphers.

- **Future Work:**
  - Weak-key attack
  - Implementing key recovery attack

Thank you

Question ?