Modifications to ULC Block Cipher Against Differential Cryptanalysis

Faris Chandra
School of Electrical Engineering and Informatics
Institute of Technology Bandung
Bandung, Indonesia
23220015@std.stei.itb.sch.id

Yusuf Kurniawan
School of Electrical Engineering and Informatics
Institute of Technology Bandung
Bandung, Indonesia
yusufk@stei.itb.ac.id

Abstract— **With the increasing proliferation of IoT devices, ensuring their security has become a critical challenge. Cryptographic algorithms play a crucial role in securing these devices by providing mechanisms for data protection. This paper focuses on the Ultra Lightweight Cryptosystem (ULC), a block cipher designed for low-power IoT devices, which has been found vulnerable to differential cryptanalysis attacks. We propose modifications to the ULC key schedule algorithm to enhance its security. The modification aims to address the vulnerable component properties of ULC while maintaining its performance. The modified algorithms were evaluated for security through differential cryptanalysis attacks. Results indicate that one of the modified designs, significantly enhance ULC's resistance to differential cryptanalysis without compromising its performance.**

Keywords—IoT; block cipher; cryptography; cryptanalysis; security.

I. INTRODUCTION

The rapid development of the Internet of Things (IoT) has revolutionized our interaction with technology. IoT connects various devices through networks, enabling efficient communication and control. This growth is driven by advancements in wireless technology, increased computing power, and the demand for smarter, more efficient solutions across multiple industrial sectors. IoT applications span healthcare, manufacturing, transportation, and smart homes, all of which require fast and secure communication for their operations.

However, the rapid growth of IoT also brings increased security risks. Cyberattacks targeting IoT devices can cause significant financial and operational damage. For example, the 2016 Distributed Denial of Service (DDoS) attack using malware-infected IoT devices disrupted internet services worldwide, affecting thousands of websites and online services [1]. This incident highlights the vulnerabilities of IoT devices and the urgent need for enhanced security measures. Moreover, IoT devices often have limited computational and energy resources, making the implementation of complex security measures challenging.

Cryptography offers an effective solution for securing IoT devices. It provides mechanisms for data protection through encryption, ensuring that only authorized parties can access sensitive information. Cryptographic algorithms are crucial for maintaining data confidentiality and integrity in IoT networks. The ongoing threats to data security drive continuous research and development in cryptography to find more secure and efficient algorithms. Cryptography is not only used for data protection during transmission but also for authentication, data integrity, and privacy protection. However, due to the limited resources of IoT devices, there is a growing need for lightweight cryptographic solutions that provide robust security without imposing significant computational or power burdens.

In the past five years, significant research has focused on developing lightweight block cipher algorithms. Notable algorithms include SLIM for Internet of Health Things [2], Shadow for IoT nodes [3], LRBC for resource-constrained IoT devices [4], FUTURE, which utilizes an optimal diffusion matrix [5], IVLBC, an involutive cipher for IoT [6]. PIPO for efficient higher-order masking software implementations [7], RAZOR is designed for IoT security [8], LELBC targets low-energy smart agriculture [9], WARP revisits GFN for a lightweight 128-bit cipher [10], while recent efforts aim at ultra-lightweight block ciphers for IoT is the Ultra Lightweight Cryptosystem (ULC) developed by Sliman et al. [11]. ULC is specifically designed for resource-limited devices, common in IoT applications. It aims to provide adequate security while maintaining efficiency in power and computational use. However, further research indicates that ULC is vulnerable to full-round differential cryptanalysis attacks using Mixed-Integer Linear Programming (MILP) [12].

The objective of this research is to modify the ULC algorithm to withstand differential cryptanalysis attacks. By implementing appropriate modifications, we aim to enhance ULC's security. To achieve this, we will analyze the components that make the algorithm susceptible to differential cryptanalysis attacks. Following this analysis, we will modify these components and evaluate the security and performance of the modified algorithm to ensure that the modifications enhance security without compromising its performance.

II. ULTRA LIGHTWEIGHT CRYPTOSYSTEM (ULC)

The ULC block cipher was proposed by L. Sliman et al [11] to address the unique security and performance needs of Internet of Things (IoT) devices. The primary objective of ULC is to provide a high level of security while maintaining low resource consumption, making it particularly suitable for devices with limited computational power and memory. ULC integrates several advanced cryptographic techniques, including bit-slicing, wide-trail strategy (WTS), and involutive methods, which together enhance its efficiency and security. Operating on a 64-bit block size, ULC employs a 15-round Substitution-Permutation Network (SPN) structure. Each round involves a key addition, a bit-slice S-box substitution, and an involutive bit permutation.

A. Key Schedule

The key schedule in ULC is designed to generate a series of round keys from an initial 80-bit key. The process involves three main steps: applying an S-box to the last 4 bits of the 80-bit key, rotating the entire key left by 61 bits, and extracting

TABLE I. ULC KEY SCHEDULE

Algorithm 1 : Key schedule

Input : $K = k_{79}\, k_{78} \ldots k_2\, k_1\, k_0$
Output : $K^r = k_{63}^r\, k_{62}^r \ldots k_2^r\, k_1^r\, k_0^r$ where $i \le r \le 16$
for $r = 1$ to 16 **do**
$(k_{79}\, k_{78}\, k_{77}\, k_{76}) = S((k_{79}\, k_{78}\, k_{77}\, k_{76}))$
$K = K <<< 61$
$K^r = (k_{79}\, k_{78} \ldots k_{18}\, k_{17}\, k_{16})$
end

the last 64 bits to be used as the round key. This procedure is repeated for each of the 16 rounds to produce the required round keys as shown at Table I below.

B. Encryption

The encryption process of ULC consists of 15 rounds of transformations. Each round includes a key addition, an S-box substitution, and an involutive permutation as shown at Table II below. The S-box employed is a 4x4 non-linear function designed to enhance security through confusion and diffusion properties (Table III). The permutation step redistributes the bits across the block to further enhance security (Table IV). The final round involves an additional key addition.

C. Differential Cryptanalysis on ULC

Two years after ULC was proposed, Kaur et al [12] perform a comprehensive cryptanalysis attack on it. The attack aimed to exploit the non-uniform behavior of input-output differentials in the cipher using differential cryptanalysis, a technique requiring high-probability differential characteristics for successful key recovery attacks. This attack utilized MILP to address the problem. The process of finding differential characteristics involves two main steps:

1. Calculating the minimum number of active S-boxes.
2. Identifying high-probability differential characteristics.

The attack began by using SageMath to compute the linear inequalities required for minimizing the number of active S-boxes in ULC. Initially, 267 inequalities were generated, which were then reduced to 21 using a MILP-based reduction algorithm. This reduction is crucial for constructing an efficient MILP model. To optimize the probability of differential characteristics, the differential probabilities of the S-boxes were considered. For ULC, the possible probabilities are $1, 2^{-2}$, and 2^{-3}, represented by two binary variables (p_0, p_1). SageMath was used to compute 543 linear inequalities, which were reduced to 19 using Gurobi. These minimized inequalities define the relationships between differential characteristics and their propagation through the S-box and permutation layers. Consequently, the differential characteristic for the 15-round ULC was identified with a probability of 2^{-45} and a minimum of 15 active. The detailed differential characteristic for the 15 rounds is on Table V below.

From Table V, we can see that the 15-round ULC's differential characteristics were optimized using MILP techniques. The analysis revealed that at least 15 active S-boxes are involved and the overall probability for differentials in the 15-round ULC is 2^{-45}. This value is significantly higher than 2^{-80}, the probability of success for an exhaustive search [13], indicating that ULC is vulnerable to differential attacks.

TABLE II. ULC ENCRYPTION

Algorithm 2 : Encryption

Input : $X = x_{63}\, x_{62} \ldots x_2\, x_1\, x_0$
$K^r = k_{63}^r\, k_{62}^r \ldots k_2^r\, k_1^r\, k_0^r$ where $i \le r \le 16$
Output : $Y = y_{63}\, y_{62} \ldots y_2\, y_1\, y_0$
for $r = 1$ to 15 **do**
$X = X \oplus K^r$
for $j = 1$ to 15 **do**
$X_{[4*j+3*,4*j]} = S(X_{[4*j+3*,4*j]})$
end
$X = P(X)$
end
$Y = X \oplus K^{16}$

TABLE III. ULC S-BOX

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	6	5	C	A	1	E	7	9	B	0	3	D	8	F	4	2

TABLE IV. ULC PERMUTATION

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P(x)$	63	59	55	51	47	43	39	35	31	27	23	19	15	11	7	3
x	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
$P(x)$	62	58	54	50	46	42	38	34	30	26	22	18	14	10	6	2
x	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**
$P(x)$	61	57	53	49	45	41	37	33	29	25	21	17	13	9	5	1
x	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**
$P(x)$	60	56	52	48	44	40	36	32	28	24	20	16	12	8	4	0

III. MODIFICATION ON ULC

In this chapter, modifications will be made to the ULC block cipher. The first step involves identifying the components that constitute vulnerabilities within ULC, rendering it susceptible to differential cryptanalysis. Once these weaknesses have been identified, appropriate modifications to these components will be determined. The goal is to enhance the security of the ULC block cipher against differential cryptanalysis by addressing the identified weaknesses and implementing the necessary changes. This process ensures that the modified ULC block cipher is more robust and resistant to cryptographic attacks, thereby improving its overall security and maintaining the performance.

TABLE V. 15 ROUND DIFFERENTIAL CHARACTERISTICS ULC

Round	Input Difference	Probability
1	0x0000000000005000	2^0
2-15	0x0000000000008000	2^{-3} each round
16	0x0000000000008880	2^{-3}

A. Component Analysis

Component analysis was conducted by examining the results of the attack performed by Kaur et al [12]. In that attack, we can identify the components that constitute the weaknesses of ULC by looking at this indicators: the total number of active S-boxes and differential trail. The analysis revealed that there are 15 active S-boxes found in the identified differential trail, indicating that there is only single active S-box in each round. The component directly related to the number of active S-boxes in each round is the permutation. Previous research has designed many permutation to increase the number of active S-boxes in each round. One such example is the block cipher PRESENT by A. Bogdanov et al. [14], whose permutation component ensures at least 10 active S-boxes in 5 rounds. This is possible because the permutation in PRESENT ensures that each input bit to an S-box comes from 4 different S-boxes, and each output bit from one S-box goes to 4 different S-boxes in the next round.

Subsequently, S. Banik, et al. [15] explained that the situation of having single active S-box in a 1-round differential trail is caused by bit permutation allowing a 1-to-1 bit transition from some S-boxes in one round to propagate to some S-boxes in the next round, which would again produce a 1-to-1 bit transition. Then, to address the issue of having only one active S-box in a single round they developed a permutation by using the "Bad Output must go to Good Input" (BOGI) strategy. This permutation resolves the issue by ensuring that a bad output bit from an S-box (an output bit that can cause a single-bit transition) does not become a bad input bit in the next round (an input bit that can result from a single-bit transition), thereby preventing the continuous propagation of a single active bit path across rounds. This is achieved through an injective mapping from bad outputs to good inputs (input bits that do not lead to single-bit transitions), eliminating the possibility of recurring 1-to-1 bit transitions that involve only one active S-box per round.

Next, we look at the differential trail of the attack. In the differential trail shown in Table V, we can see the pattern $0x0 \ldots 8000$ to $0x0 \ldots 8000$ being used repeatedly in each round. After analyzing the permutation of ULC, it was found that this pattern is formed because there is a loop in the ULC permutation at bit positions {48 49 50 51}. The permutation in ULC maps the 51st bit to the 48th bit, causing the bits to cycle within this nibble. An attacker can choose the input difference mapping 1000_b to the output difference 0001_b in the ULC S-box with a probability of 2^{-3}. When the output difference enters the ULC permutation, the 51st bit will map back to the 48th bit, forming a loop that can be exploited by the attacker to create a repeating 1-round differential pattern. We also identified potential loops at bit positions {12 13 14 15}, {24 25 26 27}, and {36 37 38 39}.

TABLE VI. SECURITY COMPARISON

Block Cipher	Active S-box	Probability	Time
ULC	15	2^{-45}	121s
Swap Modification ULC	17	2^{-49}	872s

TABLE VII. DIFFERENTIAL TRAIL SWAP MODIFICATION ULC

Round	Input Difference	Probability
1	0x0000000000005000	2^{0}
2	0x0000000000000080	2^{-2}
⋮	⋮	⋮
8	0x0000000000000880	2^{-2}
9	0x0000800080000000	2^{-6}
10	0x0000000000006000	2^{-6}
⋮	⋮	⋮
16	0x0000000000001111	2^{-3}

However, we wanted to understand the impact of the loop on the attack results by making changes to the permutation of ULC. We added a swap operation to exchange the bits that cause potential loops and then conducted differential cryptanalysis using the MILP technique employed by Kaur et al. [12] on the original ULC and the slightly modified ULC permutation. The results of the attack can be seen in Table VI, which shows the number of active S-boxes, probability, and attack time, as well as Table VII, which shows the discovereddifferential trail. Both tables indicate an increase in the number of active S-boxes from 15 to 17, the time required to find the differential trail also increased, and no iterative patterns were observed in the differential trail. However, the improvement is not significant, especially since the number of active S-boxes and the probability are still greater than 2^{-80}.

Based on both analyses, it can be determined that fixing the identified loop does not significantly impact the security of ULC. Therefore, it can be concluded that the primary cause of ULC's vulnerability to differential cryptanalysis is the small number of active S-boxes in each round. The solution to this issue is to modify the permutation component. Modifying the ULC permutation does not alter the implementation cost, as bit permutation can be realized in hardware using only wires (no logic gates required) [15].

B. Modification

The modification to ULC aims to enhance its security against differential cryptanalysis attacks by altering the components that cause vulnerability to such attacks. As previously explained, the permutation in ULC fails to effectively diffuse active bits in subsequent rounds, which makes ULC susceptible to differential cryptanalysis attacks. Therefore, the modification of ULC involves changing its permutation.

Based on the literature study conducted by AA Zakaria [16] on 101 lightweight block ciphers, approximately 65% of lightweight block cipher algorithms proposed after PRESENT use components from previous lightweight block cipher algorithms, with PRESENT being the most frequently referenced. This indicates that one way to modify a block cipher algorithm is by using components that have been previously established. In this paper, we propose four modifications to the ULC permutation by utilizing several permutations from existing block ciphers. The modification is done by replacing the Modified Permutation Layer diagram,

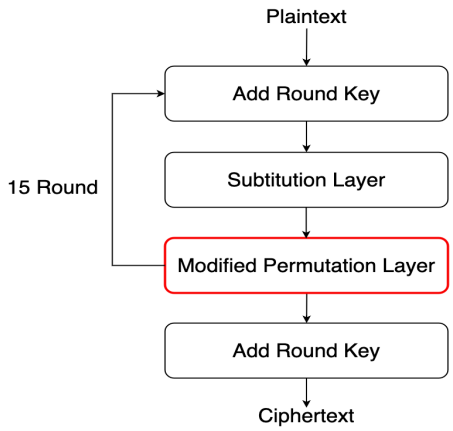

Fig. 1. Modification to ULC Permutation. (replace the Modified Permutation Layer diagram with the permutation from the specified block cipher)

as shown in Fig. 1, with four permutations from the specified block ciphers.

The permutations chosen for the modification of ULC were derived from the block cipher algorithms PRESENT, GIFT, PUFFIN [17], and ICEBERG [18]. The PRESENT permutation was selected because it is designed to spread active S-boxes in the subsequent round, addressing the specific weakness intended to be rectified in the ULC permutation. The GIFT permutation, which is an evolution of the PRESENT permutation, is also designed to spread active S-boxes. This permutation employs the BOGI strategy, ensuring that a bad output in the previous round becomes a good input in the next round. The other two permutation choices were taken from two involutive block cipher algorithms mentioned in the systematic literature review conducted by AA Zakaria [16], namely PUFFIN and ICEBERG. While there are several other candidate involutive algorithms, only these two have their involutive components located in their permutations.

The selection of these permutations is expected to demonstrate the influence of permutations specifically designed with security aspects on ULC, as well as their comparison with permutations focused on performance using involutive permutations like ULC. The illustration of the modifications on ULC can be seen in Fig. 2, which shows the diagram of the Modified Permutation Layer replaced with the four predetermined block cipher permutations. Subsequently, the four modifications will be evaluated for security, memory consumption, and performance.

IV. Evaluation

In this chapter, we will evaluate the proposed modifications and compare them with several reference algorithms used as modification benchmarks, such as PRESENT and GIFT, as well as the original algorithm, ULC

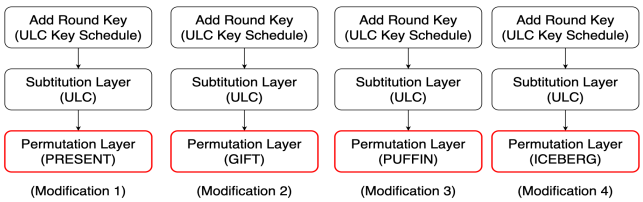

Fig. 2. Illustration of Modification on ULC.

TABLE VIII. Security Comparison

Block Cipher (Attack Round/Full Round)	Active S-box	Probability	Time
ULC (15/15) [12]	15	2^{-45}	29s
PRESENT (31/31) [13]	62	2^{-124}	872s
GIFT (12/28) [19]	24	2^{-59}	93s
Modification 1 (15/15) [this paper]	21	2^{-55}	340s
Modification 2 (15/15) [this paper]	38	2^{-101}	819s
Modification 3 (15/15) [this paper]	18	2^{-48}	148s
Modification 4 (15/15) [this paper]	20	2^{-51}	148s

algorithms used as modification benchmarks, such as PRESENT and GIFT, as well as the original algorithm, ULC. The evaluation will be conducted using a device with the specifications of an Intel i7 2.7 GHz Quad-Core processor and 16 GB RAM, focusing on aspects of security, memory consumption, and performance. This will determine whether any modifications provide better security without significantly reducing memory consumption and performance.

A. Security Evaluation

Security evaluation is conducted by performing differential cryptanalysis attacks on the proposed modifications. The differential cryptanalysis attacks are carried out using the MILP technique by Kaur et al. [12]. The results of these attacks are then compared with previous attacks performed on PRESENT [13], GIFT [19], and ULC [12], and presented in Table VIII and the chart in Fig. 3. The comparison indicators used are attack probability, the number of active S-boxes, and the time required to find the differential trail.

In Table VIII, it can be seen that Modification 2, which replaces the ULC permutation with the GIFT permutation, shows a significant increase in security. The attack probability becomes smaller at 2^{-101} compared to the attack on ULC at 2^{-45}. There is an increase in the number of active S-boxes in the 15 rounds of the ULC modification, with 38 active S-boxes compared to the 15 active S-boxes in ULC. Other modifications show an increase in security, but not as significantly. The two involutive permutations designed to enhance performance and reduce implementation costs do not provide sufficient security, similar to the permutation in ULC. This is indicated by the attack probabilities being almost the same as the attack on ULC, at 2^{-48} and 2^{-51}. In Fig. 3, we can also see a comparison chart of the security across all tested algorithms, highlighting the significant security difference

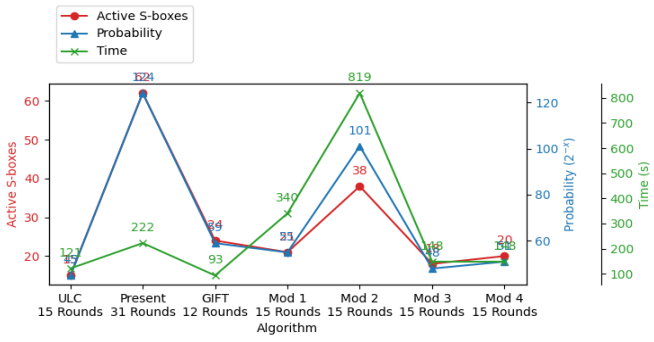

Fig. 3. Security comparison chart.

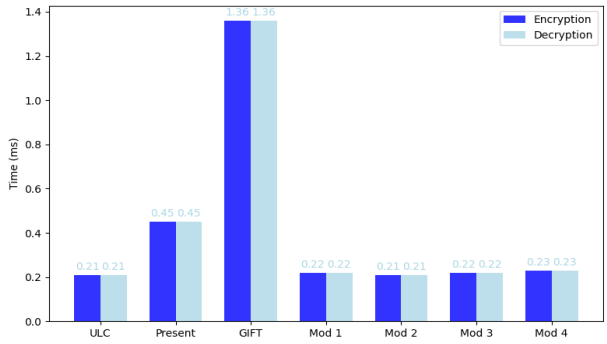

Fig. 4. Security comparison chart.

between Modification 2 and the other modifications. However, the security of Modification 2 cannot yet be compared to the security of PRESENT.

There are 62 active S-boxes with a probability of 2^{-124} in the differential cryptanalysis attack performed by S. Sun et al. [13]. This is achievable because PRESENT has 31 rounds, which significantly influences the number of active S-boxes. Nonetheless, Modification 2 achieves a similar level of security with only 15 rounds. This is advantageous because the number of rounds impacts the performance of the block cipher algorithm. Therefore, in the next chapter, performance evaluation will be conducted. It is expected that Modification 2 can maintain the proven good performance of ULC while still providing sufficient security.

B. Performance Evaluation

We evaluated the performance of the modifications by implementing the software and running it on a resource-limited device such as the Arduino UNO R3. This device was chosen to ensure that the testing method was consistent with the one used for ULC. However, there are differences in the implementation process, so this paper will repeat all performance tests on the compared algorithms.

Performance tests were conducted by performing 100 encryptions and decryptions, then calculating the average time required. In Fig. 4, it can be seen that the performance of all ULC modifications has almost the same time. This is because the modifications made were minimal, without changing the structure and components that significantly affect speed. As previously discussed, the high number of rounds in PRESENT impacts its encryption and decryption performance. With sufficient security and better performance, the second modification can be a good choice for a block cipher.

C. Memory Consumption Comparison

Evaluation of memory consumption was conducted in the same manner as the performance evaluation. Software implementations for all modifications were run on the Arduino UNO R3. During encryption and decryption, memory consumption was recorded. The memory consumption results can be seen in Fig. 5 and Fig. 6. It is evident that the RAM and ROM consumption of the ULC modifications is nearly identical to the original ULC.

V. Conclusion

The primary objective of this study is to enhance the security of ULC against differential cryptanalysis by modifying its permutation component. This adjustment aims to reduce the vulnerability caused by the original permutation's failure to effectively diffuse active bits across

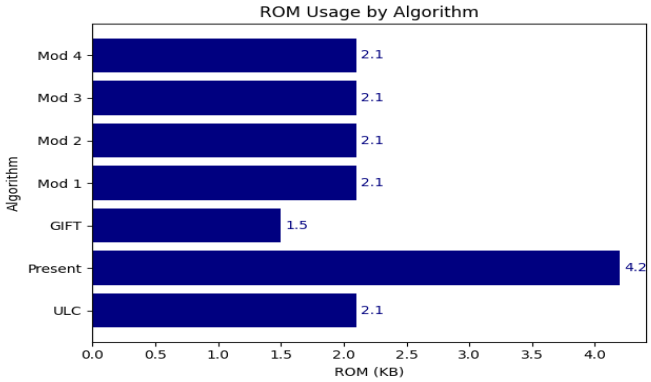

Fig. 5. ROM conumption comparison of modified ULC to other block cipher.

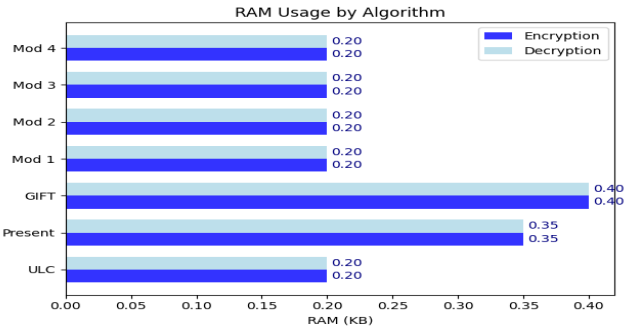

Fig. 6. RAM conumption comparison of modified ULC to other block cipher.

subsequent rounds. By incorporating permutations from well-established block ciphers, our results demonstrate that Modification 2, which replaces the ULC permutation with the GIFT permutation, successfully increases the number of active S-boxes, lowers the attack probability, and extends the time required to identify differential trails, thereby improving its security. Additionally, the performance and memory consumption evaluation results for the proposed modification show results similar to the original ULC. The implementation cost of the modified permutation remains comparable to the original ULC, as bit permutations can be realized in hardware using only wire shuffling without logic gates. Furthermore, the encryption performance and memory consumption of the modified ULC do not experience significant changes. Therefore, it can be concluded that replacing the ULC permutation with the GIFT permutation can be proposed as a modification to make ULC resistant to differential cryptanalysis attacks. Future research should focus on evaluating the security of the modified ULC against other types of attacks and on developing an involutive permutation that can spread active S-boxes in each round, ensuring resistance to cryptanalysis attacks and providing good performance.

References

[1] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. "DDoS in the IoT: Mirai and other botnets." Computer 50, no. 7, 2017, pp 80-84.

[2] B. Aboushosha, R. A. Ramadan, A. D. Dwivedi, A.El-Sayed, and M. M. Dessouky. "SLIM: A lightweight block cipher for internet of health things." IEEE Access 8, 2020, 203747-203757.

[3] Y. Guo, L. Li, and B. Liu. "Shadow: A lightweight block cipher for IoT nodes." IEEE Internet of Things Journal 8, no. 16, 2021, 13014-13023.

[4] A. Biswas, A. Majumdar, S. Nath, A. Dutta, and K. L. Baishnab. "LRBC: a lightweight block cipher design for resource constrained IoT

devices." Journal of Ambient Intelligence and Humanized Computing, 2023, 1-15.

[5] K. C. Gupta, S. K. Pandey, and S. Samanta. "FUTURE: A lightweight block cipher using an optimal diffusion matrix." In International Conference on Cryptology in Africa, pp. 28-52, Cham: Springer Nature Switzerland, 2022.

[6] X. Huang, L. Li, and J. Yang. "IVLBC: An involutive lightweight block cipher for Internet of Things." IEEE Systems Journal 17, no. 2, 2022, 3192-3203.

[7] H. Kim, et al. "PIPO: A lightweight block cipher with efficient higher-order masking software implementations." Information Security and Cryptology–ICISC 2020: 23rd International Conference, Seoul, South Korea, December 2–4, 2020, Proceedings 23. Springer International Publishing, 2021.

[8] D. Singh, M. Kumar, and T. Yadav. "RAZOR: A Lightweight Block Cipher for Security in IoT." Defence Science Journal 74, no. 1, 2024.

[9] Q. Song, L. Li, and X. Huang. "LELBC: A low energy lightweight block cipher for smart agriculture." Internet of Things 25, 2024.

[10] S. Banik, et al. "WARP: Revisiting GFN for lightweight 128-bit block cipher." Selected Areas in Cryptography: 27th International Conference, Halifax, NS, Canada, Springer International Publishing, 2021.

[11] L. Sliman, et al. "Towards an ultra lightweight block ciphers for Internet of Things." Journal of information security and applications 61, 2021.

[12] M. Kaur, T. Yadav, M. Kumar, and D. Dey. "Full-round differential attack on ULC and LICID block ciphers designed for IoT." Cryptology ePrint Archive, 2023.

[13] S. Sun, L. Hu, L. Song, Y. Xie, and P. Wang. "Automatic security evaluation of block ciphers with S-bP structures against related-key differential attacks." In International Conference on Information Security and Cryptology, pp. 39-51. Cham: Springer International Publishing, 2013.

[14] A. Bogdanov, et al. "PRESENT: An ultra-lightweight block cipher." Cryptographic Hardware and Embedded Systems-CHES 2007: 9th International Workshop, Vienna, Austria, September 10-13, 2007. Proceedings 9. Springer Berlin Heidelberg, 2007.

[15] S. Banik, et al. "GIFT: A small present: Towards reaching the limit of lightweight encryption." Cryptographic Hardware and Embedded Systems–CHES 2017: 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. Springer International Publishing, 2017.

[16] A. A. Zakaria, A. H. Azni, F. Ridzuan, N. H. Zakaria, and M. Daud. "Systematic literature review: trend analysis on the design of lightweight block cipher." Journal of King Saud University-Computer and Information Sciences 35, no. 5, 2023, 101550.

[17] H. Cheng, H. M. Heys, and C. Wang. "Puffin: A novel compact block cipher targeted to embedded digital systems." In 2008 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools, pp. 383-390. IEEE, 2008.

[18] FX. Standaert, G. Piret, G. Rouvroy, JJ. Quisquater, and JD. Legat. "ICEBERG: An involutional cipher efficient for block encryption in reconfigurable hardware." In Fast Software Encryption: 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004. Revised Papers 11, pp. 279-298. Springer Berlin Heidelberg, 2004.

[19] B. Zhu, X. Dong, and H. Yu. "MILP-based differential attack on round-reduced GIFT." In Topics in Cryptology–CT-RSA 2019: The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4–8, 2019, Proceedings, pp. 372-390. Springer International Publishing, 2019.