International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Novel Lightweight Approach to Perform Cryptography for Data Security & Privacy in IoT Mobile Devices

[1]Rahul Neve, [2]Dr. Rajesh Bansode, [3]Vikas Kaul

Submitted: 21/04/2023 **Revised**: 28/06/2023 **Accepted**: 16/07/2023

Abstract— The lightweight cryptographic (LWC) algorithm is used for resource constraint devices. The performance analysis and development of LWC is for achieving better data security in resource constrained mobile devices for effective implementation. Literature survey on LWC was carried out where it was observed that the implementation of two well-known algorithms "SIMON" and "SPECK" are in latest research as per future technological requirements. On comparing SIMON & SPECK algorithms with conventional blocks, lightweight block ciphers the following challenges that are required to be mitigated by including usage of minimal hardware overhead in proposed design (e.g., time, memory consumption), viz. use of low-cost smart mobile devices, with minimal power, low energy consumption and improved security performance.

Algorithms were implemented on the Raspberry Pi 3 with 1GB RAM, Quad Core 1.2GHz Broadcom BCM2837, with 32-bit Raspbian Operating System of 5V and current of 2 mA. Input to the algorithm is fed as text with varying size viz. 100kB, 200kB, 300kB, 400kB, 500kB.An attempt is made to developed hybrid LWC algorithm by using key scheduling logic of SPECK and Round function logic of SIMON. Experiment was performed using text file as inputs with varying sizes. On comparing the actual SIMON algorithm with Hybrid-SIMON_SPECKey algorithm it is observed that encryption and decryption time comsuption is 50% less. Thereby an improvement is observed in time and energy efficiency. Similarly in case of memory consumption of SPECK Algo with Hybrid SIMON-SPECKey algorithm it consumes an average 19% less memory during encryption as well as decryption.

Index Terms—*Block Cipher Cryptography, Constrained Devices, Decryption , Encryption , Feistel structure , IoT devices , key scheduling, Lightweight , resource constrained , symmetric-key.*

Introduction

LIGHTWEIGHT CRYPTOGRAPHY refers to the development and implementation of cryptographic techniques that are specifically designed to be efficient and effective in embedded systems, sensor networks, and other low-power IoT devices. These devices often have limited computational power, storage capacity, and energy resources, making them difficult to secure using traditional cryptographic methods.[3] The traditional cryptographic algorithms are computationally heavy in terms of energy and memory hence are not meant for constrained power devices as well as the same are very time consuming. The Internet has rapidly evolved to a complex worldwide network of thousands of heterogeneous devices with a wide range of capabilities, features, and sizes that offer a variety of services to its users. The Internet has been expanded to include smart physical things with sensing, computing, storing, and communications capabilities, known as the "Internet of Things" (IoT), in addition to mobile devices like smart watches, phones, play stations in crowd sourcing environment [1].

[1]*PhD Scholar, TCET,Kandivali, India*
[2]*Professor, Department of Information Technology , TCET , Kandivali , India ,*
[3]*Associate Professor LRTCE, Thane , India*

IoT has so far gained popularity in sectors including logistics, manufacturing, retail, and pharmaceuticals. As wireless communication, smartphone, and sensor network technologies progress, the Internet of Things (IoT) is being used by an increasing number of networked or smart items. As a result, new information and communications technology (ICT) and enterprise systems technologies have been significantly impacted by these IoT-related technologies. IoT's technological standards must be created for information exchange, storing and processing that information, and communications amongst objects if high-quality services are to be offered to end users. The success of the Internet of Things (IoT) hinges on standardization, which offers worldwide interoperability, compatibility, dependability, and efficient operations.[2]

The multiple lightweight methods developed so far have their own merits and demerits. Due to the variability of requirements, it is hard to select a one size – fits – all algorithm. Existing lightweight cryptography algorithms and techniques are facing problems with key size, encryption decryption time.

There is the need for the new algorithms which will help to optimize for low power devices, and this can be achieved through lightweight cryptographic algorithms. New logics

and techniques need to be implemented to design computationally efficient algorithm with minimal storage space , power and time consumption. Also it must ensure that these lightweight algorithm should not compromise with the security despite of minimal resources available. It can be achieved by striking a balance between the level of security provided and the number of resources required to implement the cryptographic algorithms. In addition, lightweight cryptography also places emphasis on usability and compatibility with existing protocols and systems.

In general, lightweight cryptography is essential for protecting devices with limited resources, which are more and more prevalent in the Internet of Things (IoT) age. Lightweight cryptography makes guarantee that these devices may be safely incorporated into a bigger network without jeopardizing the system's overall security by using optimized cryptographic methods and algorithms. The National Institute of Standards and Technology (NIST) has taken a leading role in the creation and standardized use of light cryptographic algorithms. Here are a few NIST-recently suggested or standardized lightweight cryptographic algorithms:

- Simon and Speck: Simon and The US National Security Agency has released the lightweight cypher block families SIMON and SPECK. The objective was to give the flexibility and performance qualities that developers need while still giving the security that the cryptography community demands. These Algorithm was submitted was selected as finalist in the NIST lightweight cryptography competition and standardized in 2018. Simon was showing good results for hardware implementation and Speck were designed to provide efficient software implementation. Simon and Speck algorithms are used to achieve confidentiality parameter of the security in the recourse constrained system.
- Another parameter of security triad is integrity, which is need to be achieved through hashing algorithm. One of the well-known lightweight hashing algorithms is ASCON-HASH [6][16]. It is based on sponge – structure. The ASCON hash function takes an input message of any length and produces a fixed-size hash value as output. The algorithm uses a permutation-based design, where the input message is first padded to a multiple of the block size and then processed in blocks using a round function that mixes the input with a secret key and a nonce. The final output is computed by applying a finalization step to the last block and returning a portion of the internal state[7][15].

This research is focus on the confidentiality part of cryptography. It is usefully to achieve privacy and secrecy in mobile crowd sourcing. After detailed study of two popular algorithm, a new hybrid algorithm was designed as well as implemented on IoT device. Following sections of paper is shows detailed study and implementation of these algorithms.

II. Simon & Speck Lightweight Algorithms

A. SIMON LWC Algorithm

Step 1: Key Expansion- The algorithm starts with expansion of the given key K into a set of round keys K_i using a key scheduling function, which depends on the size of the block and the key in terms of bits. The key scheduling function uses bitwise XOR, circular shift operations, and S-box substitutions to generate the round keys. The number of rounds in the algorithm depends on the block size and the key size.[13][14]

Let K be the secret key of length n, and let w be a word of length m such that m = n/2. The round keys k_i are generated as follows:

Key Expansion / Scheduling in SIMON

k_0 = K

for i = 0, 1, 2, ..., m-1:

w = ROTATE_RIGHT (w, 3) ⊕ k_i ⊕ (w ⊕ ROTATE_RIGHT(w, 1)) ⊕ T(w)

k_i+1 = ROTATE_LEFT(k_i, 1) ⊕ w ⊕ C_i

where ROTATE_RIGHT(w, r) denotes a right rotation of the word w by r bits, ROTATE_LEFT(k, r) denotes a left rotation of the key k by r bits, T(w) is a non-linear function that maps a word to another word, and C_i is a round constant that is specific to each round i.

Step 2: Encryption- The plaintext block P is divided into two parts PT_{i-1} and PT_i of size n/2 bits each, where n is the block size. The encryption process consists of n rounds of operations that alternate between linear transformations and non-linear substitutions. The ith round operations are given

by:

(a) Substitution: PT_{i-1} is passed through an S-box function S() to obtain S(PT_{i-1}).

(b) Linear Transformation: PT_{i-1} is bitwise XORed with the left rotated version of PT_{i-2} by a fixed number of bits R_1 and the bitwise AND of PT_{i-2} and the left rotated version of PT_{i-2} by a fixed number of bits R_2, and then bitwise XORed with S(PT_{i-1}) to obtain PT_i.

The final ciphertext C is obtained by concatenating PT_n and PT_{n-1}.

The encryption process can be expressed mathematically as follows:

Encryption in SIMON Algorithm

$PT_{i,1} = PT_{i-1,2}$

$PT_{i,2} = PT_{i-1,1} \oplus (PT_{i-1,2} <<<R_1) \oplus (PT_{i-1,2} <<<R_8) AND (PT_{i-1,2} <<<R_2))$

$PT_{i,1}' = S(PT_{i,2})$

$PT_{i,2}' = PT_{i-1,1}$

$C = PT_{n,1}' || PT_{n,2}'$

where "||" denotes concatenation, "<<<" denotes left rotation, "AND" denotes bitwise AND operation, and S() is the S-box function.

Step 3: The decryption process is similar to the encryption process, but with the round keys applied in reverse order.[4] The decryption process can be expressed mathematically as follows:

Decryption Process in SIMO Algorithm

$PT_{i,1}' = PT_{i-1,2}$

$PT_{i,2}' = PT_{i-1,1}$

$PT_{i,2} = S^{-1}(PT_{i,1}') \oplus (PT_{i-1,2}<<<R_1) \oplus (PT_{i- 1,2} AND (PT_{i-1,2}<<<R_2))$

$PT_{i,1} = PT_{i-1,1}$

$PT_{1,1} = PT_{2,2} \oplus (PT_{1,2}<<<R_1) \oplus (PT_{1,2}<<<R_8) AND (PT_{2,2}<<<R_2))$

$PT_{1,2} = PT_{2,1}$

where S^{-1}() denotes the inverse of the S-box function, and the initial values of PT_{1,1} and PT_{1,2} are obtained by concatenating the ciphertext block '*C*' in reverse order.

Above LWC SIMON algorithm is implemented on raspberry Pi 3 model B by providing text file of various sizes as input and results are noted down. Table I shows the

TABLE I SIMON ALGORITHM PERFORMANCE EVALUATION

File Size in kB	Encryption time in seconds	Decryption time in seconds	Encryption Memory consumption	Decryption Memory consumption	Energy Consumption during Encryption in milli joules	Energy Consumption during Decryption in milli joules

100	8.37	8.5	565.2 kB	647.2 kB	83.7	85
200	16.64	17.61	1.5 Mb	1.2 Mb	166.4	176.1
300	25.4	28.6	2.4 Mb	1.4 Mb	254.2	286.9
400	33.69	40.2	2.9 Mb	1.6 Mb	336.9	402
500	43.2	53.5	3.6 Mb	1.8 Mb	432	535

B. SPECK LWC Algorithm

Key Schedule:

For each round, the key schedule generates a new subkey Ki and rotates the previous key by a fixed number of bits:

Key Scheduling in SPECK

K [0], K[1], ..., K[N] = Key

for i in range(Rounds):

K[i+1] = (K[i] >> (W - A)) + S(K[i], i) + i

Key_schedule.append(K[i+1])

K[i+1] = K[i+1] & ((2 ** W) - 1)

where:

- W is the word size in bits (16 or 32)
- A is the block size in bits (64 or 128)
- Rounds is the number of encryption rounds (32 or 48)
- S is the round-dependent function that takes a key and a round index and outputs a word of the same size as the key.[8]

Encryption:

The encryption process involves a series of substitution and permutation operations on the plaintext using the generated subkeys:

Encryption Algorithm in SPECK LWC

plaintext = (P1, P2)

for i in range(Rounds):

plaintext = ((plaintext[1] ^ ((plaintext[0] << A - 8) + Key_schedule[i])) & ((2 ** A) - 1),

(plaintext[0] + ((plaintext[1] >> A - 8) ^ Key_schedule[i])) & ((2 ** A) - 1))

ciphertext = plaintext

Decryption:

The decryption process involves a series of substitution and permutation operations on the ciphertext using the generated subkeys in reverse order:

Decryption in SPECK LWC

ciphertext = (C1, C2)

for i in range(Rounds - 1, -1, -1):

ciphertext = ((ciphertext[1] - ((ciphertext[0] << A - 8) + Key_schedule[i])) & ((2 ** A) - 1),

(ciphertext[0] ^ ((ciphertext[1] >> A - 8) ^ Key_schedule[i])) & ((2 ** A) - 1))

plaintext = ciphertext

where:

-P1 and P2 are the two blocks of the plaintext (each of size A/2),

- C1 and C2 are the two blocks of the ciphertext (each of size A/2)

- - represents subtraction

- ^ represents bitwise XOR

- << represents left shift

- >> represents right shift

- & represents bitwise AND

LWC SPECK Algorithms was implemented, and observation are noted in Table II.

TABLE II SPECK ALGORITHM PERFORMANCE EVALUATION

File Size in kB	Encryption time in seconds	Decryption time in seconds	Encryption Memory consumption	Decryption Memory consumption	Energy Consumption during Encryption in milli joules	Energy Consumption during Decryption in milli joules
100	3.6	4.3	720.9 KB	647.2 KB	360	430
200	7.38	8.89	1.7 MB	720 KB	738	889
300	11.18	15.9	2.6 MB	876.5 KB	111.8	159
400	14.7	22.77	3.1 MB	1.5 MB	147	227.7
500	18.46	31.64	3.8 MB	1.3 MB	184.6	316.4

III. Hybrid Simon-Spekey Lwc Algorithm

The new hybrid algorithm is developed by combination of confusion and diffusion rounds of SIMON and key scheduling of SPECK algorithm after following analysis:

The confusion and diffusion layers are essential components of block ciphers that help to ensure the security of the algorithm. Both LWC SIMON and SPECK have been designed to achieve strong confusion and diffusion properties, although they use different methods to achieve these goals. In LWC SIMON, the confusion and diffusion properties are achieved using a Feistel network structure. Each round of the Feistel network involves applying a nonlinear function to one half of the block and combining the result with the other half using a linear function. The confusion property is achieved with the implementation of nonlinear function whereas linear function is to achieve diffusion Specifically, the nonlinear function used in LWC SIMON is an S-box that maps a 16-bit input to a 16-bit output. The S-box is designed to be highly nonlinear and to provide good diffusion properties. The linear function used

in LWC SIMON involves bitwise XOR and rotations operations, which help to spread the effect of the nonlinear function across the entire block.

In LWC SPECK, substitution-permutation network (SPN) structure is used to achieved the confusion and diffusion properties. The SPN structure involves applying a sequence of S-boxes and permutation layers to the input block. The S-boxes provide confusion, while the permutation layers provide diffusion[12].The S-boxes used in LWC SPECK are 8-bit to 8-bit affine transformations that are designed to be highly nonlinear and to provide good confusion properties. The permutation layers involve shuffling the bits of the block to provide diffusion. SIMON uses a Feistel network structure and an S-box to provide confusion and diffusion, while SPECK uses an SPN structure and a sequence of S-boxes and permutation layers to achieve the same goals.

The key scheduling function of LWC Speck algorithm is more efficient than that of LWC Simon algorithm due to a few reasons:

Computational Complexity: The key scheduling function of Speck requires fewer operations and fewer rounds of computation compared to the key scheduling function of Simon. Speck key schedule consists of three phases: key expansion, whitening key generation, and round key generation, which are relatively simple and efficient operations involving bitwise rotations, XOR operations, and modular addition. In contrast, the Simon key scheduling function involves a more complex Feistel network structure and requires more rounds of computation.

Key Size: Speck uses smaller key sizes than Simon, which results in a simpler and more efficient key scheduling function. Speck supports key sizes of 64, 96, and 128 bits, while Simon supports key sizes of 64, 96, 128, 160, and 256 bits. The larger key sizes of Simon require more rounds of computation and a more complex key scheduling function.[11][12]

Implementation: The simpler and more efficient key scheduling function of Speck makes it easier to implement in both hardware and software environments. This is particularly significant for resource-constrained environments such as mobile IoT devices, where efficient implementation is critical.[10]

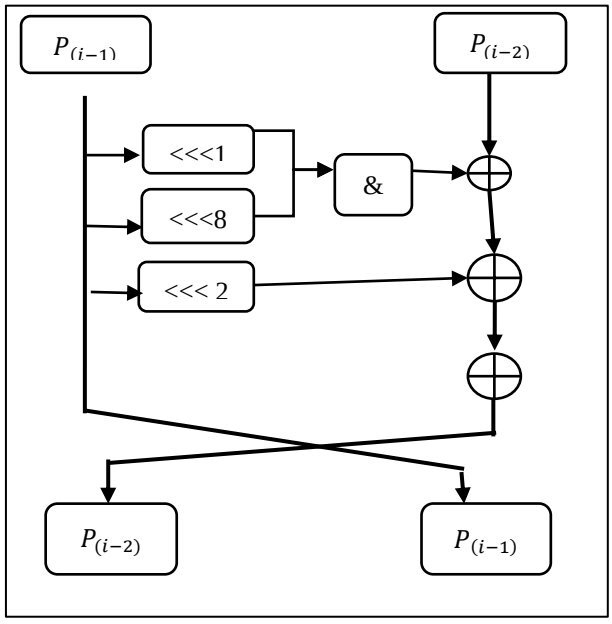

Fig 1: Round Function SIMON Algorithm

Fig 1 depict the SIMON round function; it is based on feistel structure. Detailed description is already mentioned in step 2 of SIMON Encryption

TABLE III NEW HYBRID SIMON-SPECKEY ALGORITHM PERFORMANCE EVALUATION.

File Size in kB	Encryption time in seconds	Decryption time in seconds	Encryption Memory consumption	Decryption Memory consumption	Energy Consumption during Encryption in milli joules	Energy Consumption during Decryption in milli joules
100	4.18	4.28	520 kB	483.3	41.8	42.8
200	8.43	9.11	1.8 MB	583 kB	84.3	91.1
300	12.58	15.7	2.2 MB	720kB	125.8	157
400	16.8	22.6	3.2 MB	1.5 MB	168	226
500	20.9	30.9	3.7 MB	1.2 MB	209	309.7

IV. Implementation Of Hybrid-Simonspeckey In The Iot System.

To achieve data security for IoT kind of devices, newly developed Hybrid-SIMONSPECKey algorithm is used in the system. Figure 2 illustrates a proposed system for data security for resource constrained devices (IoT kind of devices). System will accept data in form of text, audio, video from various sensor nodes, these sensors will be attached with controller. Controller such as raspberry pi accept the data from sensors. LWC encryption algorithm process the data and cipher block is created. This cipher / encrypted data is transmitted to another node or on the cloud through communication channels. At the receiving end cipher text is decrypted using Hybrid-SIMONSPECKey decryption module.

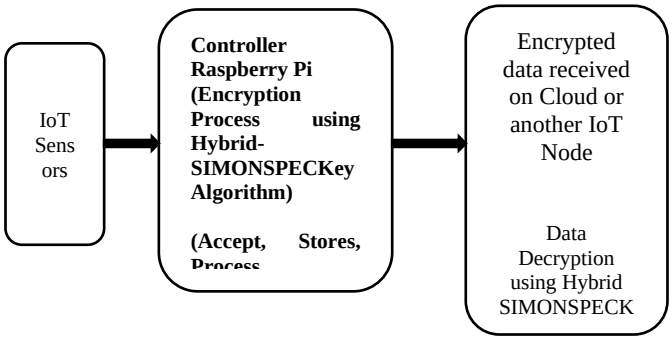

Fig 2: Use of Hybrid-SIMONSPECKey Encryption & Decryption in IoT System for Data Security

In this system, a sensor device is used to gather sensing data, which is then processed by a controller using a lightweight cryptographic algorithm. Before transmission of data to another node or to the cloud, it is encrypted to make it secure and maintain integrity. At the receiving end, the encrypted data is decrypted using a newly developed lightweight cryptographic algorithm, which is designed to be efficient in terms of processing power, energy and memory consumption.

The lightweight cryptographic algorithm used in this system is specifically designed for mobile devices, such as sensors and controllers, that have limited processing power and memory capacity. The algorithm must be able to encrypt and decrypt data quickly, while also minimizing the energy consumption required for these operations. Newly developed Hybrid-SIMONSPECKey Algorithm ensures data protection against unauthorized access or data. At the receiving end, the encrypted data is decrypted using the same lightweight cryptographic algorithm to ensure the integrity and confidentiality of the data.

V. Comparative Analysis Of Experimental Results

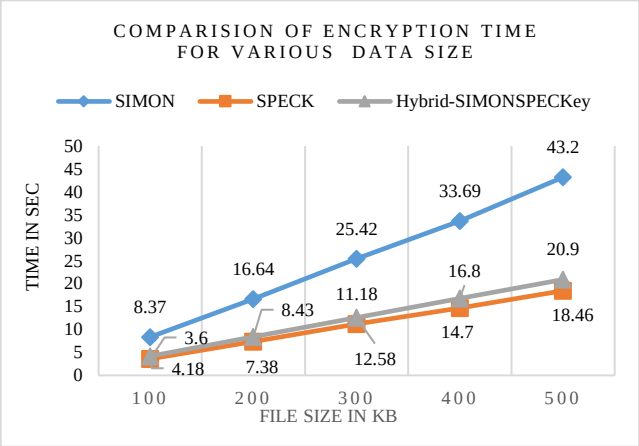

Fig 3: Encryption time consumed in seconds.

In figure 3 It is clearly observed that the Hybrid-SIMONSPECKey algorithm consumes less time for encryption process when compared with the SIMON algorithm where as it is very nearer to the results of SPECK algorithm.

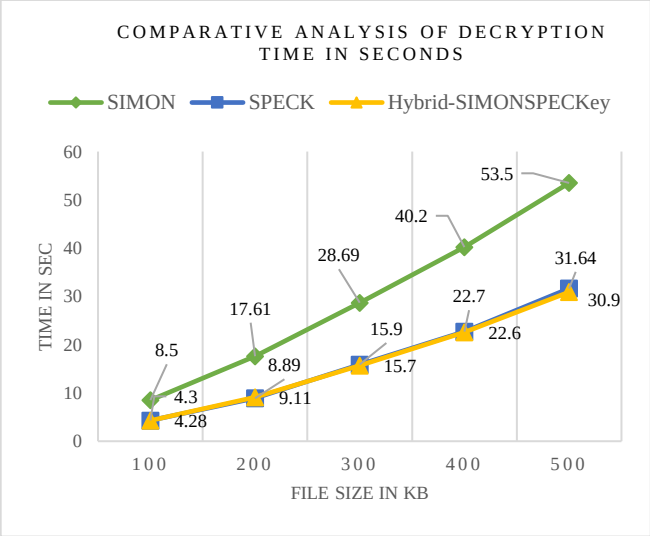

Fig 4: Decryption time in seconds

The hybrid algorithm which is newly developed using the rounds of SIMON and the key scheduling of SPECK demonstrates a significant improvement in decryption time over the SIMON algorithm. With a file size of 100 kb, the hybrid algorithm takes only 4.28 seconds to decrypt the data, which is almost half the time taken by the SIMON algorithm alone. This suggests that the hybrid algorithm offers a good balance between security and efficiency, making it an attractive option for lightweight cryptography applications.

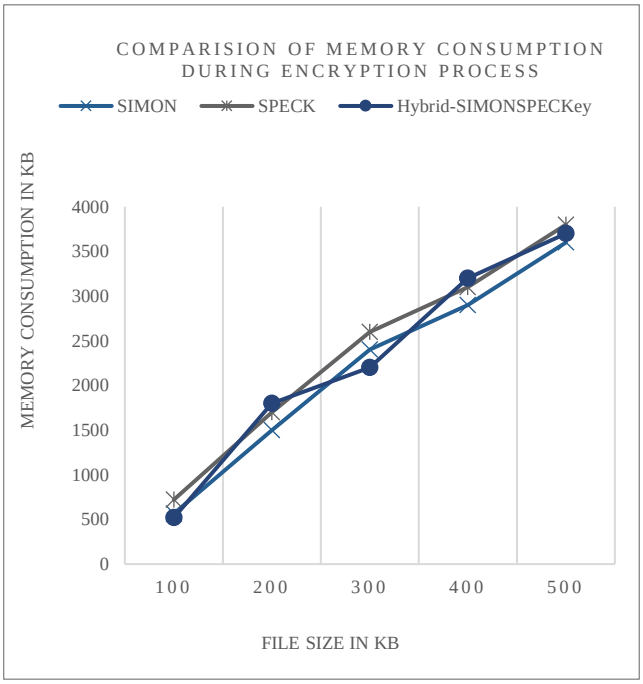

Fig 5: Memory consumption in KB during Encryption process

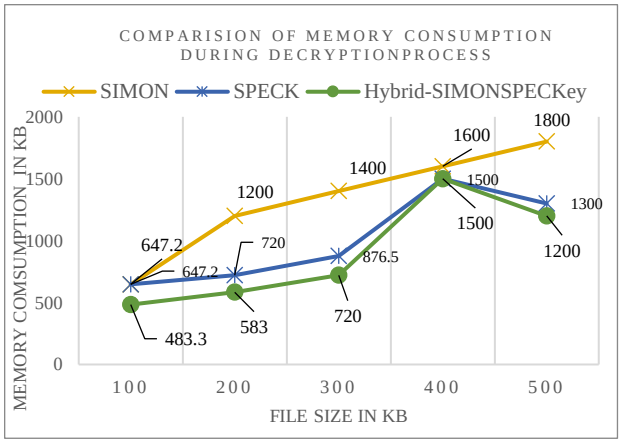

Fig 6: Analysis of memory in KB during decryption process

After performing the analysis on the parameter like time, memory and energy consumption , table IV shows comparison of SIMON algorithm with new developed hybrid algorithm.

TABLE IV COMPARATIVE ANALYSIS OF NEW HYBRID ALGORITHM WITH SIMON ALGORITHM .

Technical Parameter Measured (for 100kB of text file)	SIMON	Hybrid-SIMON SPECKey
Time consumption for Encryption & Decryption	SIMON algorithm consumes 8.37 secs for encryption and 8.5 secs for decryption.	Hybrid-SIMONSPECKey algorithm consumes 4.18 secs for encryption and

		4.28 secs for decryption which is nearer to the SPECK results.
Memory Consumption during Encryption & Decryption	In comparison of SPECK algorithm, SIMON algorithm consumes less memory and consumes 8% more memory for encryption and almost 30% more memory for decryption in comparison with Hybrid SIMON-SPECKey algorithm.	Efficiency in terms of memory utilization is improved as Hybrid SIMON SPECKey algorithm consumed less memory during encryption as compared to SIMON algorithm.
Energy Consumption in milli joules	SIMON algorithm consumes 50% more energy as compared to SPECK and Hybrid SIMON-SPECKey algorithm during encryption & decryption process	As Hybrid SIMON SPECKey consumes less time , thereby requires comparatively less energy for execution

V. Conclusion

Our study proposes a novel lightweight approach for performing cryptography in IoT devices that offers improved processing speed, memory usage, and energy consumption. In this research, we proposed a hybrid algorithm that combines the Simon and Speck lightweight cryptography algorithms to improve encryption performance. The rounds of Simon and the key scheduling of Speck is used to developed a new lightweight algorithm with enhanced performance in terms of time, memory, and energy consumption.

The experimental results prove that the hybrid algorithm outperforms Simon in terms of time & energy consumption, whereas it shows good results for memory consumption when compared to SPECK algorithm. This improvement in performance is achieved while maintaining the same level of security offered by the original Simon and Speck algorithms.[9]

This new Hybrid LWC algorithm shows good results in terms of time, energy, and memory consumption. It consumes 50% less time as well as energy as compared to the SPECK Algo, Hybrid SIMON-SPECKey consumes average 19% less memory during encryption as well as decryption.

The proposed approach addresses the security concerns associated with the transmission of sensitive data in IoT and mobile devices. The use of lightweight cryptographic algorithms ensures that the data is protected from unauthorized access or tampering during transmission, thereby maintaining the integrity and confidentiality of the data.

References

[1] M. N. Khan, A. Rao and S. Camtepe, "Lightweight Cryptographic Protocols for IoT-Constrained Devices: A Survey," in IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4132-4156, 15 March15, 2021, doi: 10.1109/JIOT.2020.3026493.

[2] L. D. Xu, W. He and S. Li, "Internet of Things in Industries: A Survey," in IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233-2243, Nov. 2014, doi: 10.1109/TII.2014.2300753.

[3] Biryukov, A., Leurent, G., Perrin, L. (2016). Cryptanalysis of Feistel Networks with Secret Round Functions. In: Dunkelman, O., Keliher, L. (eds) Selected

Areas in Cryptography – SAC 2015. SAC 2015. Lecture Notes in Computer Science, vol 9566. Springer, Cham. https://doi.org/10.1007/978-3-319-31301-6_6.

[4] Zhang X, Liu B, Zhao Y, Hu X, Shen Z, Zheng Z, Liu Z, Chong KS, Yu G, Wang C, Zou X. Design and Analysis of Area and Energy Efficient Reconfigurable Cryptographic Accelerator for Securing IoT Devices. Sensors (Basel). 2022 Nov 25;22(23):9160. doi: 10.3390/s22239160. PMID: 36501862; PMCID: PMC9739433.

[5] S. A. Hamad, Q. Z. Sheng, W. E. Zhang, and S. Nepal, "Realizingan Internet of Secure Things: A Survey on Issues and EnablingTechnologies," IEEE Communications Surveys Tutorials, vol. 22, no. 2,pp. 1372–1391, Secondquarter 2020.

[6] X. Wei, M. El-Hadedy, S. Mosanu, Z. Zhu, W. -M. Hwu and X. Guo, "RECO-HCON: A High-Throughput Reconfigurable Compact ASCON Processor for Trusted IoT," 2022 IEEE 35th International System-on-Chip Conference (SOCC), Belfast, United Kingdom, 2022, pp. 1-6, doi: 10.1109/SOCC56010.2022.9908100.

[7] Dobraunig, C., Eichlseder, M., Mendel, F. et al. ASCON v1.2: Lightweight Authenticated Encryption and Hashing. J Cryptol 34, 33 (2021). https://doi.org/10.1007/s00145-021-09398-9

[8] K. Aggarwal, "Comparison of RC6, modified RC6 & enhancement of RC6," 2015 International Conference on Advances in Computer Engineering and Applications, Ghaziabad, India, 2015, pp. 444-449, doi: 10.1109/ICACEA.2015.7164746.

[9] Okeya, K., Sakurai, K. (2001). Efficient Elliptic Curve Cryptosystems from a Scalar Multiplication Algorithm with Recovery of the y-Coordinate on a Montgomery-Form Elliptic Curve. In: Koç, Ç.K., Naccache, D., Paar, C. (eds) Cryptographic Hardware and Embedded Systems — CHES 2001. CHES 2001. Lecture Notes in Computer Science, vol 2162. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-44709-1_12

[10] Epishkina, A.V., Kanner, A.M., Kanner, T.M. (2020). Comprehensive Testing of Software and Hardware Data Security Tools Using Virtualization. In: Misyurin, S., Arakelian, V., Avetisyan, A. (eds) Advanced Technologies in Robotics and Intelligent Systems. Mechanisms and Machine Science, vol 80. Springer, Cham. https://doi.org/10.1007/978-3-030-33491-8_9

[11] H. Hasan, G. Ali, W. Elmedany and C. Balakrishna, "Lightweight Encryption Algorithms for Internet of Things: A Review on Security and Performance Aspects," 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakheer, Bahrain, 2022, pp. 239-244, doi: 10.1109/3ICT56508.2022.9990859.

[12] A. S. Omar and O. Basir, "SIMON 32/64 and 64/128 block cipher: Study of cross correlation and linear span attack immunity," 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 2017, pp. 1-6, doi: 10.1109/PIMRC.2017.8292209.

[13] Sun, B., Liu, M., Guo, J., Qu, L., Rijmen, V. (2016). New Insights on AES-Like SPN Ciphers. In: Robshaw, M., Katz, J. (eds) Advances in Cryptology – CRYPTO 2016. CRYPTO 2016. Lecture Notes in Computer Science(), vol 9814. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-53018-4_22

[14] Roy, S., Shrivastava, M., Pandey, C.V. et al. IEVCA: An efficient image encryption technique for IoT applications using 2-D Von-Neumann cellular automata. Multimed Tools Appl 80, 31529–31567 (2021). https://doi.org/10.1007/s11042-020-09880-9

[15] K. Ramezanpour, P. Ampadu and W. Diehl, "A Statistical Fault Analysis Methodology for the Ascon Authenticated Cipher," 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), McLean, VA, USA, 2019, pp. 41-50, doi: 10.1109/HST.2019.8741029.

[16] J. Kaur, M. Mozaffari Kermani and R. Azarderakhsh, "Hardware Constructions for Error Detection in Lightweight Authenticated Cipher ASCON Benchmarked on FPGA," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 4, pp. 2276-2280, April 2022, doi: 10.1109/TCSII.2021.3136463.

[17] Johansson Anna, Maria Jansen, Anna Wagner, Anna Fischer, Maria Esposito. Machine Learning Techniques to Improve Learning Analytics. Kuwait Journal of Machine Learning, 2(2). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/189

[18] Johansson Anna, Maria Jansen, Anna Wagner, Anna Fischer, Maria Esposito. Machine Learning Techniques to Improve Learning Analytics. Kuwait Journal of Machine Learning, 2(2). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/189

[19] Dhabliya, D., Soundararajan, R., Selvarasu, P., Balasubramaniam, M. S., Rajawat, A. S., Goyal, S. B., . . . Suciu, G. (2022). Energy-efficient network protocols and resilient data transmission schemes for wireless sensor Networks—An experimental survey. Energies, 15(23) doi:10.3390/en15238883