Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Performance Analysis of The Speck Cryptography Algorithm

Ahmed Fanfakh[1*], Nihad Abduljalil[2]

[1]University of Babylon, 51002, Babylon, Iraq. E-mail: ahmed.fanfakh@uobabylon.edu.iq
[2]University of Warith Al-Anbiyaa, 56001, Karbala, Iraq. E-mail: nihad.ab@uowa.edu.iq
*Corresponding Author E-mail: ahmed.fanfakh@uobabylon.edu.iq

https://doi.org/10.46649/fjiece.v4.2.3a.20.9.2025

Abstract. *Encryption algorithms such as the block cipher techniques were built in particular to guarantee security on limited systems, whose design clarity is vital. The multiplicity of the planned use cases, however, needs flexibility in execution. In the building of cryptosystems, simplicity, security, and flexibility are continuous but incompatible aims. The requirement for a lightweight block cipher algorithm that supports a broad range of systems, architectures, and block/key sizes efficiently has been developing recently. In 2013, the National Security Agency (NSA) recommended the block ciphers SIMON and SPECK. Different block sizes, including 16, 32, and 64 bits, and key sizes, such as 64, 96, and 128 bits, are supported by SIMON and SPECK. In this research, we evaluate the performance of lightweight SPECK cryptographic block ciphers utilizing multiple criteria such as execution time, throughput, and energy usage. We utilize two distinct Intel CPUs to assess the Speck Cryptography technique. Three keys of sizes 128, 192, and 256 are employed in this article to compare the performance criteria. The obtained results over two different Intel processors show a linear and non-linear relationship to the key size. In other words, the increase in key length increases the execution time and energy consumption, while throughput results decrease. Moreover, the NIST statistical randomness test is implemented for all speck versions to compare them in term of the security level.*

Keywords: *Speck cipher; Cryptography; Throughput; Energy consumption*

1. INTRODUCTION

To provide high security, it is preferable to use cryptographically powerful elements and to advance an algorithm far more frequently than might initially appear appropriate. Effectiveness, a competing goal, requires us to reduce computation as much as possible. The art of cryptography is how to strike a balance between these competing objectives. The reality that performance is not a clearly delineated concept adds to the difficulty. An algorithm may execute efficiently on specialized hardware (such as an ASIC), yet perform poorly on 8-bit microcontrollers. Or it might permit but necessitate a lot of code, high-throughput applications on 64-bit desktop processors. Alternatively, it might be built to maximize efficiency on a certain CPU. More information and communication technology (ICT) devices are now being included into industrial automation systems thanks to advances in technology. The fourth industrial revolution, or Industry 4.0, introduces creative and economic approaches while fundamentally altering established structures and technologies. These systems are basically industrial cyber-physical production systems [1] (furthermore related to the Industrial Internet of Things [2] more recently), where the rapid growth of ICT allows for the development of cutting-edge services and products, technological innovations, and enhanced

tools, as well as improved production rewards while lowering expenses. Supervisory Control and Data Collecting (SCADA) systems' monitoring and data acquisition capabilities have significantly improved

thanks to this technical advancement, but it also raises serious questions about how vulnerable they are to cyberattacks [3],[4],[5]. PLCs, or programmable logic controllers, are essential parts of SCADA systems. Although they are virtually prominent in each sector of business, they are extremely capable of controlling a wide range of industrial systems. The main producers are only now beginning to pay attention to their security aspects, which were virtually missing until recently. The addition of cryptography at the PLC's application level might be the initial phase in supplying these systems with the required security. Additionally, new advancements in the field, as in [6], [7], allow for the execution of cryptography on various devices with limited computational power. The implementation problems of the SIMON and SPECK classes of simple block ciphers in PLC applications are discussed in this work. It demonstrates that, despite the two block ciphers' outstanding performance and potential for imposing application-layer security requirements, designers must carefully examine the underlying hardware architecture, especially the supported data types. The enormous variety of SIMON and SPECK variations ensures that there are enough possibilities, despite the fact that these can have a substantial influence on the efficiency of the program.
In this paper, we study the performance of lightweight SPECK cryptography block ciphers using different factors such as execution time, throughput, and energy consumption.

2. RELATED WORKS

The Advanced Encryption Standard (AES) is widely used as a symmetric cipher across various applications, yet it lacks the flexibility required for low-power and resource-constrained devices [8]. To address these limitations, the NSA developed the lightweight block cipher families SIMON and SPECK in 2013 [6], [7], [9], which have been extensively evaluated for performance in diverse hardware and software environments. For instance, Bethrow et al. [10] studied SPECK on an MSP430 processor, and Manifas et al. [11] demonstrated its applicability in embedded systems, while ARM-NEON acceleration further enhances efficiency [12]. Recent cryptanalysis studies [13], [14] focus on key-recovery attacks and the structural properties of these ciphers, confirming their resilience under reduced-round scenarios. A key concern in lightweight cipher implementations is the generation of unpredictable and secure keys. Predefined or public key material can compromise security by increasing predictability [15]. Key Derivation Functions (KDFs) are therefore essential in generating high-entropy round keys and cryptographic material from a single master key, mitigating related-key and key-recovery attacks [16], [17]. In the context of SPECK, an efficient KDF is critical to maintain the cipher’s lightweight performance while ensuring statistical independence of derived keys, strengthening resistance against cryptanalytic attacks. Notably, the Quasigroup-based KDF proposed in [18] demonstrates a fully key-dependent expansion that significantly enhances security by making every change in the input affect the derived key output. By integrating a robust KDF with SPECK, our approach addresses both efficiency and security requirements, ensuring suitability for constrained devices while preserving the advantages of lightweight encryption. Recent cryptanalysis studies have focused on the security aspects of SIMON and SPECK. Notably, Gohr's work introduced a neural differential distinguisher for reduced-round SPECK32/64, achieving significant accuracy. Furthermore, Zhang et al. improved upon this model by incorporating advanced neural network architectures, enhancing the distinguisher's effectiveness [19].

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

3. PERFORMANCE MEASUREMENTS

The system's performance is measured by its throughput capacity and energy consumption. Both these metrics depend on the execution time of the application. The Speck cryptography algorithm uses different key sizes, each of which needs more rounds to encrypt or decrypt the plain message. However, the number of rounds in the speck algorithm increases the required execution time to encrypt or decrypt the message. The main goal of this paper is to study the performance cost of the speck algorithm while using different key lengths. The throughput of encrypting or decrypting a message of size *N* during the execution time *T* is calculated as in Eq. 1.

$$\text{Thoughpot}\ (GigabitPerSecond) = \frac{Message\ size\ (GigaBit)}{\text{Execution time (Sec)}} \qquad (1)$$

The above equation is used by many researchers in the literature, such as in [20].
The energy consumption of a processor executing an application in the execution time unit T can be calculated by multiplying the power consumption of a processor by the execution time [21], [22], [23]. Thus, the energy consumption is computed as in Eq. 2.

$$Energy\ (Joule) = Prcessor_{power\ consuption}(W) * Execution\ time\ (Sec) \qquad (2)$$

4. SPECK CRYPTOGRAPHY ALGORITHM

Speck operates on data blocks of a predetermined size and uses a key for both encryption and decryption, see Fig. 1. The dimensions of the block and key directly impact the level of security and the amount of data that can be processed in one operation. The rounds number in Speck's round algorithm is determined by the key and block size [15], as shown in Table 1.

Table 1. The number of rounds in Speck cipher

Key Length (Bits)	Block Size (Bits)	Number of Rounds
64	32	22
72	48	22
96	48	23
96	64	26
128	64	27
96	96	28
144	96	29
128	128	32
192	128	33
256	128	34

The Speck round function may be decomposed into three distinct operations: XOR, modulo addition, and rotation, as seen in Eq 3.

$$Li{+}1 = ((Li \gg \alpha) \boxplus R) \oplus ki \qquad (3)$$
$$Ri{+}1 = (Ri \ll \beta) \oplus Li{+}1$$

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

The parameters Li and Ri denote the left and right portions of the data during round i, respectively. The variable ki represents the round key during round i, while α and β are constants used for rotation.

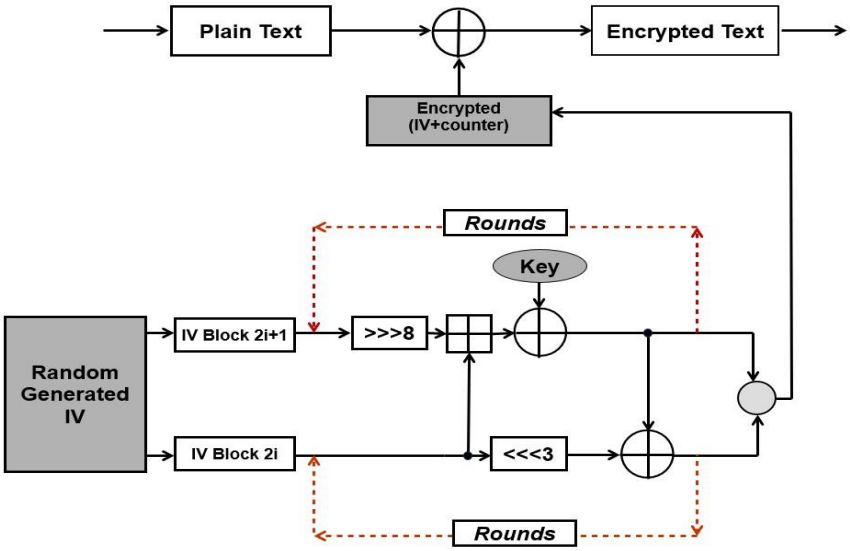

Fig.1. SPECK round function.

The last three instances presented in Table 1 were taken into consideration since, according to the Federal government, cryptographic protection must have a minimum-security level of 112 bits [24]. The length requirements for the plaintext and user-supplied key inputs are necessary for the SPECK CTR strategy to produce a ciphertext. The algorithm of the key derivation function acts as a key generator for the algorithm and is used as key input during the key scheduling and encryption process. See [25] for more details. The algorithms 1 and 2 present the Speck cryptography of CTR mode while encrypting/decrypting a block size of 128 bits while using different key sizes.

Algorithm 1. Speck Round Function

Input:
- Two 64-bit words: x, y
- Round key k

Output:
- Updated words x, y

Procedure:
1. Rotate xxx right by 8 bits.
2. Add y to x modulo 2^{64}
3. XOR x with the round key k.
4. Rotate y left by 3 bits.
5. XOR y with the updated value of x.

Return the updated pair (*x*, *y*).

Algorithm 2. Speck Encryption Function

Input:
- Key schedule of size 128 or 192 or 256 bits
- No_rounds: the number of rounds 32, 33, 34 depends on key size
- iv: the initial vector, two blocks each of size 64 bits
- in: an array of plain message composed of n blocks each of size 64 bits

Output:

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

- out: cipher message of size n blocks

Procedure:

```
1: for i = 0 → n do
2:   iv[0] ← iv[0] + i
3:   crypted_iv[0] ← iv[0]
4:   crypted_iv[1] ← iv[1]
5:   for j = 0 → No_rounds do
6:        SpeckRound (crypted_iv[1], crypted_iv[0], key_schedule[j])
7:   End of For j
7:   out[i] ← crypted iv[0] ⊕ in[i]
8:   out[i + 1] ← crypted iv[1] ⊕ in[i + 1]
9: End of for i
```

5. ANALYTICAL RESULTS AND EXPERIMENTS

5.1 EXPERIMENTS SETTING

In this section, we present the Speck cryptography results, which used three different lengths of the key as well as two types of processors. The three commonly used lengths of keys are (128/128, 128/192, and 128/256). For each key size, there is a different number of rounds for the speck cryptography. The number of rounds is 32, 33, and 34 for the keys 128, 192, and 256, respectively. The two types of processors used in this work are Xeon processors operating and Intel Core-i7 processors operating at 2.2 and 2.8 GHz frequencies, respectively. Table 2 shows the detailed characteristics of the two processors used in the experiments. Moreover, ten message sizes were used, ranging from 10 to 100 megabytes.

Table 2. The computing processors' technical details

Processor Name	Frequency speed (GHz)	Cache memory Size (MB)	Power consumption (watt)	Operational use
Intel Xeon	2.2	2	65	Desktop processor
Intel Core i7-700HQ	2.8	6	45	Laptop processor

5.2 SECURITY RESULTS OF NIST STATISTICAL TEST

The NIST developed a comprehensive group of examinations to assess the statistical properties of binary sequences, ensuring they exhibit true randomness as required for cryptographic applications, is a crucial tool used to assess the reliability of RandomNumberGenerators (RNGs) and cryptographic algorithms [26]. Amidst the current reliance on randomization for the security of many applications, it is crucial to understand the purpose and importance of the NIST randomization test suite. The primary objective of the NIST Randomness Test Group is to assess the statistical characteristics of sequences produced by random number generators (RNGs). The purpose of these statistical tests is to identify any possible biases, trends, or anomalies in the sequence that may compromise its randomization and unpredictability. Considering the crucial role of random numbers in cryptographic protocols, the generator's

capability to successfully pass the rigorous NIST suite tests demonstrates its suitability for cryptographic applications. Through all NIST tests, the P-values falling within an acceptable range (typically 0.01 to 0.99) show the occurrence of arbitrariness attributes in the ciphertext. The absence of p-values in this range could potentially suggest the presence of non-randomness. Table 3. illustrates the results of the P-values from all NIST tests, with both versions of the Speck cipher undergoing testing and yielding acceptable P-values. However, based on the mean p-values

Table 3. The Comparison of the Statistical NIST Test P-values Results

Test name	Speck 128/128	Speck 128/192	Speck 128/256	Best Performer
Frequency	0.0179 (weak)	0.350	0.740	128/256
BlockFrequency	0.213	0.534	0.122	128/192
CumulativeSums	0.350	0.740	0.740	128/192 & 256
Runs	0.350	0.740	0.740	128/192 & 256
Rank	0.911	0.740	0.213	128/128
FFT	0.740	0.534	0.534	128/128
Avg. NonOverlappingTemplate	0.497	0.436	0.516	128/256
OverlappingTemplate	0.871	0.911	0.740	128/192
ApproxEntropy	0.213	0.534	0.122	128/192
Serial	0.350	0.911	0.740	128/192
LinearComplexity	0.911	0.534	0.350	128/128
Average P-Value	**0.493**	**0.633**	**0.505**	**128/192**

Among the three Speck variants showed in Table 3, Speck 128/192 exhibits the most balanced randomness, with consistently high p-values across most tests. Speck 128/256 performs reliably but shows slightly lower uniformity in some tests, while Speck 128/128, despite strong structural metrics, displays noticeable bias in basic randomness indicators. Overall, Speck 128/192 is the statistically most robust variant. The increase in the key size in the case of 192 and 256 gives better p-values compared to 128. This indicates that the key size absolutely increases the security level of the cipher algorithm.

5.3 THE PERFORMANCE RESULTS

The Speck cryptography algorithm is one of the lightweight cryptographies that are used in small devices. However, this section presents and analyzes its performance results in terms of execution time, energy consumption, and throughput. Two different processors were used to conduct these experiments. The primary goal of this research is to demonstrate how the performance of the Speck algorithm changes when it is implemented on various hardware devices. The execution time results are presented in Fig.2 for three different key lengths. As mentioned before, the key length in the Speck cipher produces more rounds when increased. However, the results of the execution time over the two processors revealed that increasing the key length increases the algorithm's execution time.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

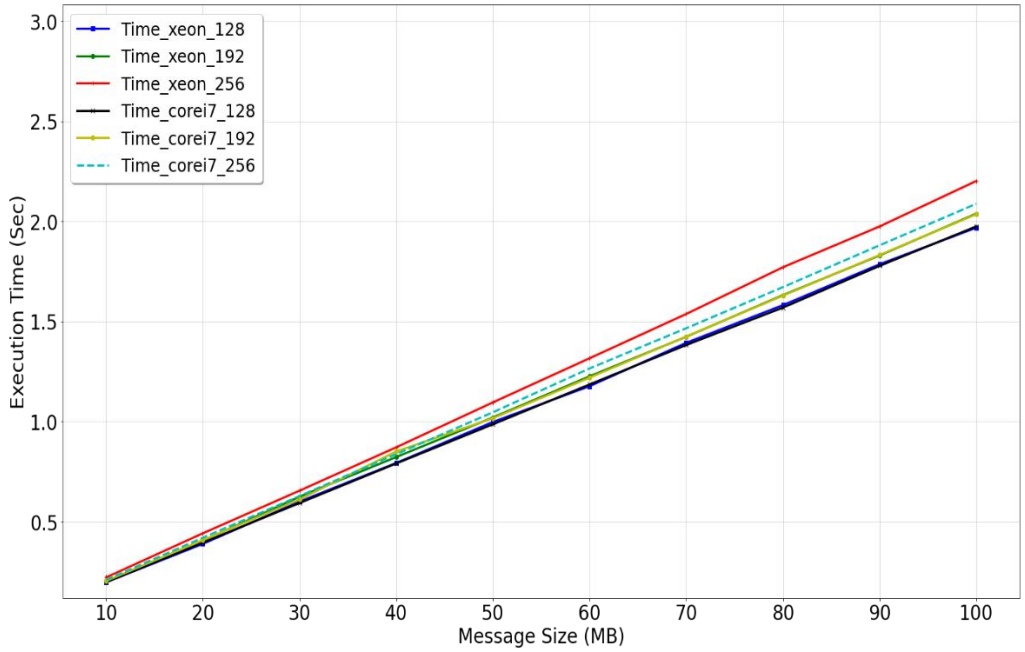

Fig.2. The execution time results of the Speck algorithm

Noticeably, there is a slight fluctuation that occurs in the execution time between the two processors' results. Whereas, this fact is not valid for energy consumption results when the power consumption of a processor is higher than the other one. Fig. 3. presents the energy consumption results. It shows that the Intel Xeon processor consumed more energy with an average increase of up to 45% compared to the Intel core i7.

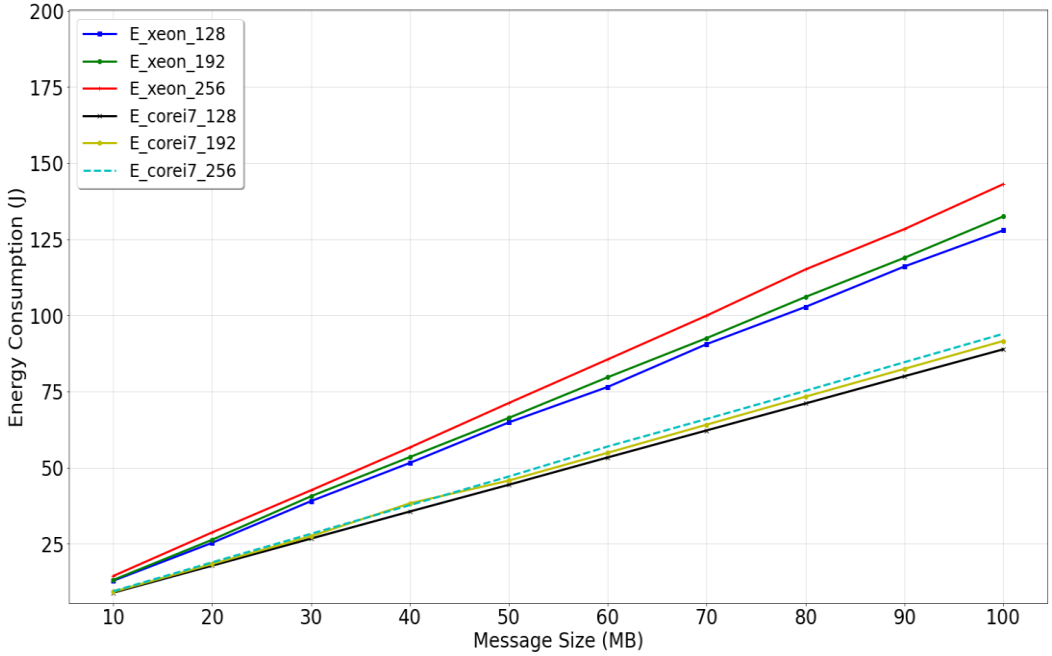

Fig. 3. The energy consumption results of the Speck algorithm

Indeed, the energy consumption depends mainly on two factors: the execution time and the power consumption of the processor. However, the Intel Xeon processor consumes more energy due to its low computing speed, high-power consumption, and emphasizing cache size and Thermal Design Power (TDP)

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

impact compared to the Intel Core i7 processor. Cache memory plays an important role in the increase of execution time, which can also affect all performance metrics.

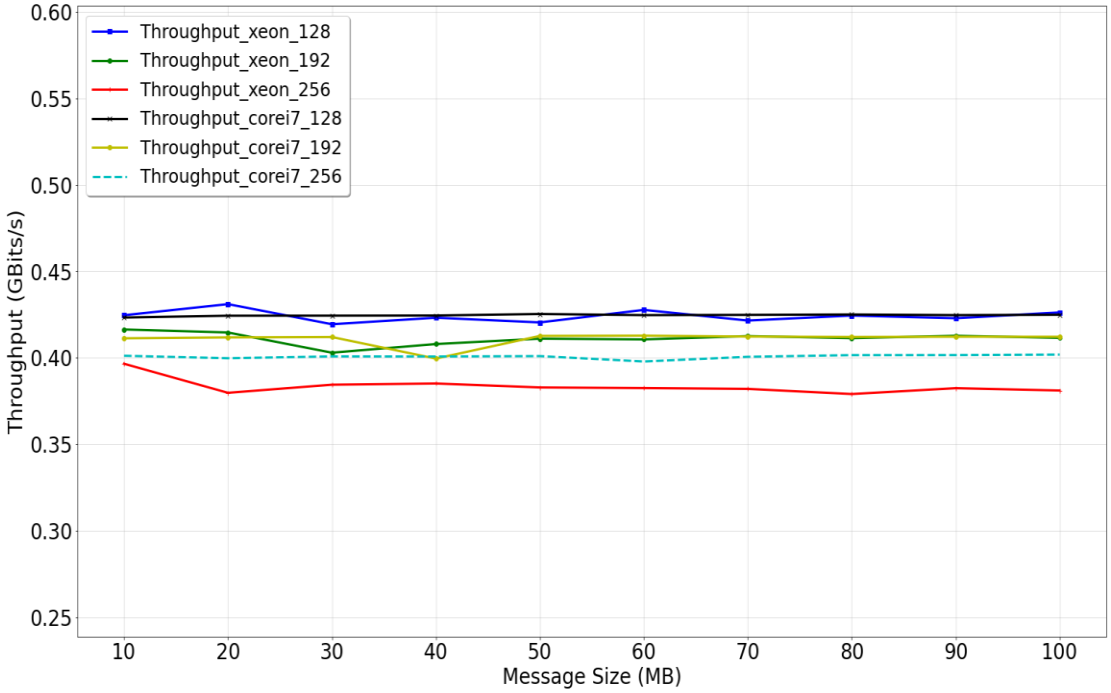

Fig. 4. The throughput results of the Speck algorithm

The throughput results mainly depend on the execution time, which depends on the speed of the processor. The throughput results are presented in Fig. 4. It demonstrates that the Speck with a 128-key size gives more throughput. This is because the algorithm needs fewer rounds and thus consumes fewer execution times.

6. CONCLUSION AND FUTURE WORK

This paper presents a detailed performance study for the Speck cryptography algorithm. The algorithm used three different keys to prove how the performance metrics used change accordingly to their lengths. The performance metrics used are execution time, energy consumption, and throughput. Moreover, two Intel processors are utilized in the experiments. These processors are different in their computing speed and power consumption. The obtained results show that the execution time is increased while the key length is also increased. Whereas energy results are affected by the increment of execution time and power usage. The throughput results explain the non-linear relation to the execution time.

In the future, other processor types such as ARM-based and embedded processors will used in the experiment to evaluate its capabilities compared to Intel family processors. Consequently, speck cryptography will be implemented on parallel multicore processors, and the three-performance metrics are interesting for study and analysis.

REFERENCES

[1] Drath, R., & Horch, A., Industrie 4.0: Hit or hype? [industry forum], IEEE Industrial Electronics Magazine, 8(2), 56, 2014.
[2] Da Xu, L., He, W., & Li, S., Internet of things in industries: A survey, IEEE Transactions on Industrial Informatics, 10(4), 2233, 2014.
[3] Hagerott, M., Stuxnet and the vital role of critical infrastructure operators and engineers, Int. J. Crit. Infrastructure Prot., 7(4), 244, 2014.
[4] Goodin, D., First known hacker-caused power outage signals troubling escalation, Ars Technica, 4, 2016.
[5] Liang, G., Weller, S. R., Zhao, J., Luo, F., & Dong, Z. Y., The 2015 Ukraine blackout: Implications for false data injection attacks, IEEE Transactions on Power Systems, 32(4), 3317, 2016.
[6] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L., The SIMON and SPECK lightweight block ciphers, Proceedings of the 52nd Annual Design Automation Conference, 1, 2015.
[7] Nithya, R., & Kumar, D. S., Where AES is for Internet, SIMON could be for IoT, Procedia Technology, 25, 302, 2016.
[8] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L., SIMON and SPECK: Block Ciphers for the Internet of Things, Cryptology ePrint Archive, 2015.
[9] Buhrow, B., Riemer, P., Shea, M., Gilbert, B., & Daniel, E., Block cipher speed and energy efficiency records on the MSP430: System design trade-offs for 16-bit embedded applications, International Conference on Cryptology and Information Security in Latin America, Springer, Cham, 104, 2014.
[10] Manifavas, C., Hatzivasilis, G., Fysarakis, K., & Rantos, K., Lightweight cryptography for embedded systems – a comparative analysis, In Data Privacy Management and Autonomous Spontaneous Security, Springer, Berlin, Heidelberg, 333, 2013.
[11] Park, T., Seo, H., & Kim, H., Parallel implementations of SIMON and SPECK, 2016 International Conference on Platform Technology and Service (PlatCon), IEEE, 1, 2016.
[12] Abed, F., List, E., Lucks, S., & Wenzel, J., Cryptanalysis of the SPECK family of block ciphers, Cryptology ePrint Archive, 2013.
[13] Abed, F., List, E., Lucks, S., & Wenzel, J., Differential and linear cryptanalysis of reduced-round SIMON, Cryptology ePrint Archive, 2013.
[14] Alkhzaimi, H. A., & Lauridsen, M. M., Cryptanalysis of the SIMON family of block ciphers, Cryptology ePrint Archive, 2013.
[15] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L., The SIMON and SPECK families of lightweight block ciphers, Cryptology ePrint Archive, 2013.
[16] Disina, A. H., Jamel, S., Pindar, Z. A., & Deris, M. M., All-or-Nothing Key Derivation Function Based on Quasigroup String Transformation, 2016 International Conference on Information Science and Security (ICISS), IEEE, 1, 2016.
[17] Chuah, C. W., Dawson, E., & Simpson, L., Key derivation function: the SCKDF scheme, IFIP International Information Security Conference, Springer, Berlin, Heidelberg, 125, 2013.
[18] Krawczyk, H., Cryptographic extraction and key derivation: The HKDF scheme, Annual Cryptology Conference, Springer, Berlin, Heidelberg, 631, 2010.
[19] Yue, X., & Wu, W., Improved Neural Differential Distinguisher Model for Lightweight Cipher Speck, Applied Sciences, 13(12), 6994, 2023.
[20] Fanfakh, A., Noura, H., & Couturier, R., ORSCA-GPU: One round stream cipher algorithm for GPU implementation, J. Supercomput., 78, 11744, 2022.
[21] Fanfakh, A. B. M., Predicting the performance of MPI applications over different grid architectures, JUBPAS, 27(1), 468, 2019.

[22] Idrees, S. K., & Fanfakh, A. B. M., Performance and energy consumption prediction of randomly selected nodes in heterogeneous cluster, In New Trends in Information and Communications Technology Applications (NTICT 2018), Springer, Cham, 938, 2018.
[23] Fanfakh, A., Charr, J.-C., Couturier, R., & Giersch, A., CPUs energy consumption reduction for asynchronous parallel methods running over grids, IEEE Intl. Conf. on Computational Science and Engineering (CSE), Embedded and Ubiquitous Computing (EUC), and Distributed Computing and Applications for Business Engineering (DCABES), 205, 2016.
[24] Barker, E., & Roginsky, A., Transitioning the use of cryptographic algorithms and key lengths, NIST Special Publication 800-131A Rev. 2 (Draft), National Institute of Standards and Technology, 2018.
[25] Lustro, R. A. F., Modified Key Derivation Function for Enhanced Security of Speck in Resource-Constrained Internet of Things, Int. J. Comput. Netw. Inf. Secur., 13(4), 2021.
[26] Rukhin, A., et al., Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST, Revised April 2010.

