Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.3, June 2025, pp. 642~650
e-ISSN: 2548-6861

Evaluation of the Effectiveness of Lightweight Encryption Algorithms on Data Performance and Security on IoT Devices

Damar Indrajati [1]*, Wahid Miftahul Ashari [2]*
* Teknik Komputer, Universitas Amikom Yogyakarta
damarindr@students.amikom.ac.id [1], wahidashari@amikom.ac.id [2]

Article Info

Article history:

Received 2025-03-07
Revised 2025-03-20
Accepted 2025-03-26

Keyword:

Internet of Things (IoT),
Lightweight Encryption,
Encryption Performance,
Data Security,
Power Efficiency.

ABSTRACT

Data security remains a major concern in the Internet of Things (IoT) landscape due to the inherent limitations in computational power, memory capacity, and energy availability of IoT devices. To address these challenges, lightweight encryption algorithms have emerged as alternatives to conventional cryptographic methods, aiming to balance performance and security. This study evaluates the effectiveness of five encryption algorithms—SIMON64/128, SPECK64/128, XTEA64/128, PRESENT64/128, and AES128—on IoT devices through experimental analysis of their security strength, execution time, CPU utilization, memory usage, and power efficiency. The experiments were conducted on a Raspberry Pi 3B+ using C-based implementations to emulate realistic IoT scenarios. The findings reveal that AES128 offers the strongest security characteristics, including the highest Avalanche Effect (39.29%) and Differential Resistance Score (6.76/10), but at the expense of significant resource consumption. In contrast, SIMON64/128 and SPECK64/128 deliver superior performance in terms of speed and resource efficiency, making them ideal for low-power environments, albeit with concerns about potential cryptographic backdoors. XTEA64/128 emerges as a practical compromise, delivering moderate security and low power consumption without known vulnerabilities. Based on these results, AES128 is suitable for high-capacity IoT platforms prioritizing strong encryption, while SIMON and SPECK are preferable for resource-constrained devices, with XTEA serving as a balanced alternative. This research contributes a comparative framework to guide the selection of encryption algorithms for IoT systems, ensuring an optimal trade-off between security and operational efficiency.

I. PENDAHULUAN

Dalam era digital yang semakin berkembang, keamanan data menjadi aspek krusial dalam implementasi *Internet of Things (IoT)*. IoT memungkinkan perangkat saling berkomunikasi dan bertukar data melalui jaringan, sehingga meningkatkan efisiensi dan otomatisasi di berbagai sektor[1]. Namun, perangkat IoT seringkali memiliki keterbatasan sumber daya, seperti daya pemrosesan rendah, kapasitas memori terbatas, serta konsumsi daya yang harus dijaga agar efisien. Oleh karena itu, penggunaan algoritma enkripsi ringan menjadi solusi utama untuk menjaga keamanan data tanpa membebani sistem secara signifikan[2].

Sejumlah penelitian sebelumnya telah mengevaluasi berbagai algoritma enkripsi ringan dalam konteks IoT, seperti Penelitian yang berjudul *"IoT Security: Implementation of Xtea, Simon/Speck Lightweight Block Ciphers"* yang ditulis oleh Ertaul & Chauhan (2023) membahas tentang perbandingan algoritma XTEA dan Algoritma SIMON/SPECK. Fokus dari penelitian ini adalah mengetahui algoritma manakah yang memiliki efisiensi paling tinggi berdasarkan beberapa indikator, seperti *execution time*, *power consumption*, dan *memory requirement*. Hasil dari penelitian ini adalah algoritma XTEA lebih baik dibandingkan dengan algoritma SIMON dan SPECK dalam hal efisiensi[3].

Penenlitian yang berjudul *"Performance Evaluation of IoT Encryption Algorithms: Memory, Timing, and Energy"* yang

ditulis oleh Maitra et all. (2019) membahas tentang kinerja algoritma enkripsi ringan, khususnya AES (dengan dan tanpa akselerasi perangkat keras) dan XTEA, untuk digunakan dalam perangkat IoT yang terbatas sumber daya. Fokus penelitiannya adalah pada penggunaan memori, waktu eksekusi, dan konsumsi energi. Hasil penelitiannya menunjukkan bahwa meskipun AES yang dipercepat perangkat keras adalah yang tercepat dan paling efisien dalam hal energi, XTEA menawarkan alternatif yang layak untuk mikrokontroler dengan sumber daya rendah, dengan sedikit lebih banyak waktu dan energi yang dikonsumsi tetapi memerlukan memori yang jauh lebih sedikit[4].

Penelitian yang berjudul *"High-Speed Implementation of PRESENT on AVR Microcontroller"* yang ditulis oleh Kwon et all. (2021) membahas mengenai implementasi dan optimasi dari cipher blok PRESENT, khususnya pada mikrokontroler AVR yang digunakan dalam lingkungan dengan sumber daya terbatas seperti aplikasi IoT. Fokus penelitian ini adalah pada teknik optimasi untuk mode operasi *Electronic Code Book (ECB)* dan *Counter (CTR),* dengan penekanan pada penggunaan tabel pencarian yang telah dihitung sebelumnya untuk meningkatkan kinerja. Hasil dari penelitian ini menunjukkan bahwa implementasi yang diusulkan mencapai perbaikan signifikan dalam siklus jam per byte (CPB) dibandingkan dengan karya sebelumnya, dengan mode PRESENT-CTR menunjukkan efisiensi yang lebih baik dibandingkan mode ECB. Penelitian ini menekankan kelayakan penggunaan optimasi ini dalam aplikasi dunia nyata pada perangkat dengan sumber daya rendah dan menyimpulkan bahwa pendekatan mereka secara efektif mengurangi overhead yang terkait dengan algoritma PRESENT, menjadikannya cocok untuk berbagai aplikasi kriptografi[5].

Penelitian yang berjudul *"Light-Weight Present Block Cipher Model for IoT Security on FPGA"* yang ditulis oleh Bharathi & Parvatham (2022), membahas tentang desain dan implementasi modul enkripsi dan dekripsi ringan menggunakan cipher PRESENT untuk keamanan *Internet of Things (IoT).* Fokus penelitian ini adalah pada efisiensi, konsumsi daya, dan kesesuaian modul tersebut untuk perangkat IoT. Hasil dari penelitian menunjukkan bahwa modul PRESENT-80/128/256 yang diimplementasikan pada FPGA Artix-7 memiliki penggunaan daya yang rendah (0.186 – 0.192 W) dan *throughput* yang tinggi (hingga 1644 Mbps), menjadikannya ideal untuk aplikasi IoT yang aman. Penelitian ini juga membandingkan kinerja modul dengan cipher blok ringan yang ada, menyoroti perbaikan dalam *throughput* dan efisiensi[6].

Penelitian yang berjudul *"Simon and Speck: Block Ciphers for the Internet of Things"* yang ditulis oleh Beaulieu et al. (2015) membahas tentang dua cipher blok ringan, yaitu SIMON dan SPECK, yang dirancang untuk aplikasi di lingkungan terbatas, khususnya di *Internet of Things (IoT).* Fokus penelitian ini adalah pada kinerja, implementasi, dan keamanan dari cipher tersebut. Hasil penelitian menunjukkan bahwa SIMON dan SPECK memiliki efisiensi yang baik dalam implementasi ASIC dan perangkat lunak, dengan SIMON mencapai *throughput* 17.1 Gbit/s dan SPECK mencapai 10.6 Gbit/s dalam implementasi ASIC. Selain itu, kedua algoritma ini menunjukkan kinerja yang lebih baik dibandingkan dengan AES dalam konteks aplikasi kriptografi ringan, serta memiliki ketahanan yang lebih baik terhadap serangan saluran samping. SIMON dan SPECK juga dirancang untuk fleksibilitas dan kesederhanaan, menjadikannya alternatif yang efektif untuk cipher blok tradisional dalam lingkungan yang terbatas[7].

Penelitian yang berjudul "Perbandingan Algoritma SIMON dan SPECK Dalam Pengamanan Citra Digital" yang ditulis oleh Fatma et all. (2024) membahas mengenai algoritma kriptografi SIMON dan SPECK yang dirancang untuk enkripsi dan dekripsi citra digital. Fokus penelitiannya adalah pada kinerja kedua algoritma dalam hal waktu enkripsi dan dekripsi, perubahan ukuran file, serta tingkat keacakan menggunakan metrik seperti *UACI (Unified Average Changing Intensity)* dan *NPCR (Number of Pixels Change Rate).* Hasil penelitiannya menunjukkan bahwa algoritma SPECK lebih cepat dalam waktu enkripsi dan dekripsi dibandingkan SIMON. Kedua algoritma menyebabkan peningkatan ukuran file setelah enkripsi sekitar 24%. Nilai UACI menunjukkan bahwa SPECK mencapai tingkat keacakan yang lebih tinggi (20,94%) dibandingkan SIMON (19,65%), sementara nilai NPCR keduanya serupa. Namun, kedua algoritma tidak memenuhi ambang batas teoritis minimum untuk keamanan yang efektif[8].

Penelitian lain yang menjadi pertimbangan adalah Studi oleh Yustiarini et al. dengan judul *"A Comparative Method for Securing Internet of Things (IoT) Devices: AES vs Simon-Speck Encryptions"* membahas perbandingan kinerja algoritma enkripsi AES, SIMON, dan SPECK dalam mengamankan komunikasi IoT menggunakan protokol MQTT. Fokus penelitian ini adalah mengevaluasi performansi jaringan, termasuk delay, throughput, efisiensi penggunaan memori, dan *avalanche effect* dari masing-masing algoritma. Hasil penelitian menunjukkan bahwa SPECK memiliki kinerja terbaik dalam mengurangi delay komunikasi dan efisiensi penggunaan memori, sementara SIMON memiliki nilai avalanche effect tertinggi, yang menunjukkan sensitivitas lebih tinggi terhadap perubahan input. AES, meskipun lebih aman, memiliki performa yang lebih rendah dalam aspek efisiensi komunikasi dan memori dibandingkan SIMON dan SPECK. Studi ini menyoroti pentingnya pemilihan algoritma enkripsi yang optimal sesuai dengan keterbatasan sumber daya perangkat IoT untuk memastikan keseimbangan antara keamanan dan efisiensi[9].

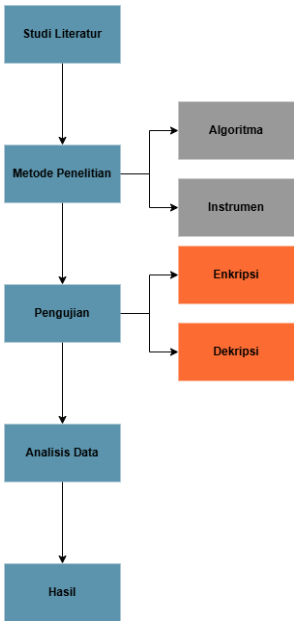
Perbedaan penelitian ini dengan penelitian sebelumnya adalah Penelitian ini membandingkan lima algoritma secara langsung: SIMON, SPECK, XTEA, PRESENT, dan AES. selain itu penelitian ini berfokus pada evaluasi efektivitas protokol enkripsi ringan terhadap kinerja dan keamanan data pada Raspberry Pi 3B+, sebuah perangkat yang sering digunakan dalam implementasi IoT kelas menengah dan edge computing. Meskipun Raspberry Pi bukanlah perangkat ultra-

low power seperti mikrokontroler STM32 atau ESP32, hasil evaluasi ini tetap relevan karena perangkat dengan spesifikasi lebih rendah cenderung menunjukkan tren performa yang serupa dalam eksekusi algoritma enkripsi ringan, meskipun dengan skala yang berbeda[10].

Melalui penelitian ini, diharapkan dapat diperoleh pemahaman yang lebih mendalam mengenai bagaimana setiap algoritma enkripsi ringan beroperasi pada perangkat IoT khususnya Raspberry Pi. Selain itu, hasil penelitian ini akan memberikan referensi dalam pemilihan algoritma enkripsi yang optimal untuk implementasi di lingkungan IoT yang membutuhkan keseimbangan antara efisiensi dan keamanan data.

II. METODE

Pada bagian ini dijelaskan langkah-langkah yang dilakukan dalam penelitian untuk mengevaluasi efektivitas protokol enkripsi ringan terhadap kinerja dan keamanan data pada perangkat IoT.



Gambar 1. Diagram Alur

A. *Diagram Alur*

Diagram alur penelitian seperti pada Gambar 1 menunjukkan tahapan utama dalam studi ini, dimulai dari studi literatur, metode penelitian, pengujian, analisis data, hingga memperoleh hasil akhir. Diagram ini disusun berdasarkan kajian literatur dan referensi dari berbagai penelitian sebelumnya yang membahas perbandingan algoritma kriptografi ringan dalam keamanan perangkat IoT[8][11]. Dengan merujuk pada penelitian-penelitian tersebut, diagram alur penelitian disusun untuk merepresentasikan proses evaluasi performa algoritma kriptografi ringan pada perangkat IoT.

B. *Algoritma*

Penelitian ini menggunakan lima algoritma enkripsi, yaitu SIMON, SPECK, XTEA, PRESENT, dan AES, yang dipilih berdasarkan relevansinya dalam perangkat IoT dan berbagai penelitian sebelumnya.

SIMON dan SPECK dipilih karena banyak digunakan dalam penelitian kriptografi ringan serta telah diterapkan pada berbagai perangkat IoT yang memiliki keterbatasan sumber daya. Kedua algoritma ini dikembangkan oleh NSA dengan tujuan menyediakan enkripsi yang efisien dalam perangkat dengan daya dan komputasi terbatas. Meskipun terdapat isu keamanan terkait potensi backdoor dalam desainnya, SIMON dan SPECK tetap banyak digunakan dan layak untuk dibandingkan dalam penelitian ini guna mengevaluasi efektivitasnya dan sebagai pembanding dalam konteks IoT[12][13].

XTEA dipilih karena telah banyak digunakan dalam penelitian mengenai enkripsi ringan dan memiliki efisiensi tinggi dalam implementasi perangkat dengan sumber daya terbatas. Algoritma ini merupakan pengembangan dari TEA dengan peningkatan pada keamanan dan efisiensi dalam hal konsumsi daya dan komputasi. Penggunaannya dalam perangkat IoT telah diteliti secara luas, menjadikannya salah satu kandidat yang relevan dalam perbandingan ini[14].

PRESENT dipilih karena merupakan bagian dari standar ISO/IEC 29192, yang mengklasifikasikan algoritma ini sebagai salah satu skema enkripsi ringan yang sesuai untuk perangkat dengan keterbatasan sumber daya. Standar tersebut juga mencakup algoritma lain seperti CLEFIA dan LEA, namun dalam penelitian ini, PRESENT dipilih karena memiliki ukuran blok dan panjang kunci yang sama dengan algoritma enkripsi ringan lainnya yang diuji, yaitu 64 bit untuk ukuran blok dan 128 bit untuk panjang kunci. Hal ini memungkinkan perbandingan yang lebih adil terhadap algoritma lain dalam aspek efisiensi dan keamanan[15][16].

AES128 dipilih karena merupakan varian terkecil dari AES yang masih cukup ringan untuk digunakan pada perangkat IoT. Meskipun AES bukan termasuk kategori algoritma enkripsi ringan, algoritma ini telah menjadi standar keamanan global dengan tingkat perlindungan yang tinggi dan banyak diterapkan pada perangkat IoT yang memerlukan enkripsi kuat. AES digunakan sebagai pembanding utama untuk mengetahui sejauh mana algoritma enkripsi ringan dapat memberikan keseimbangan antara keamanan dan performa dibandingkan dengan algoritma yang lebih kuat tetapi memiliki kebutuhan komputasi yang lebih tinggi [9].

TABEL I
ALGORITMA YANG DIGUNAKAN

No	Algoritma	Ukuran Blok	Panjang Kunci
1	PRESENT	64 bit	128 bit
2	SIMON	64 bit	128 bit
3	SPECK	64 bit	128 bit
4	XTEA	64 bit	128 bit
5	AES	128 bit	128 bit

C. Instrumen

Instrumen perbandingan dalam penelitian ini dirancang untuk mengevaluasi performa algoritma kriptografi ringan pada perangkat IoT, khususnya Raspberry Pi 3B+. Implementasi algoritma enkripsi dilakukan menggunakan bahasa C, sesuai dengan spesifikasi algoritma yang diuji. Parameter evaluasi mencakup keamanan, kecepatan eksekusi, penggunaan CPU, penggunaan memori, dan konsumsi daya, dengan masing-masing berbobot 25%, untuk memastikan keseimbangan dalam aspek keamanan serta efisiensi, sebagaimana tercantum dalam Tabel 2.

Hal ini didasarkan pada penelitian sebelumnya yang menyoroti faktor utama dalam evaluasi efisiensi algoritma enkripsi pada perangkat dengan keterbatasan sumber daya. Salah satu referensi utama adalah penelitian oleh Makaraneko dalam *"A Comparative Analysis of Cryptographic Algorithms in the Internet of Things"*, yang menganalisis perbandingan algoritma kriptografi dalam komunikasi IoT. Penelitian ini menekankan pentingnya throughput, konsumsi energi, konsumsi daya, dan penggunaan memori sebagai parameter utama dalam menilai kinerja algoritma enkripsi pada perangkat IoT [17].

Selain itu, penelitian oleh Ertaul dalam *"IoT Security: Implementation of XTEA, SIMON/SPECK Lightweight Block Ciphers"* mengevaluasi efisiensi XTEA, SIMON, dan SPECK dalam konteks keamanan IoT. Parameter evaluasi yang digunakan meliputi waktu eksekusi, kebutuhan memori, dan konsumsi daya. Studi ini memberikan wawasan penting terkait efisiensi dan kesesuaian algoritma tersebut untuk perangkat IoT dengan keterbatasan sumber daya [3].

Lebih lanjut, penelitian oleh Sudip Maitra dalam *"Performance Evaluation of IoT Encryption Algorithms: Memory, Timing, and Energy"* juga menyoroti konsumsi memori, waktu eksekusi, dan konsumsi energi sebagai metrik utama dalam mengevaluasi kinerja algoritma enkripsi IoT. Studi ini secara khusus menyoroti kelayakan algoritma XTEA untuk digunakan pada platform embedded dengan sumber daya terbatas [4].

Selain studi berbasis eksperimen, penelitian Suryateja et al. *dalam "A Survey on Lightweight Cryptographic Algorithms in IoT"* memberikan tinjauan komprehensif mengenai *lightweight cryptography (LWC)* yang dirancang untuk mengamankan perangkat IoT. Studi ini menegaskan bahwa kriptografi ringan harus memperhatikan tiga aspek utama, yaitu keamanan, biaya, dan kinerja, agar dapat diterapkan secara optimal pada perangkat dengan daya dan komputasi terbatas [18].

Lebih lanjut, penelitian oleh Noor Mahesa Naser et al. dalam *"A Systematic Review of Ultra-Lightweight Encryption Algorithms"* membahas secara mendalam algoritma enkripsi ringan dan ultra-ringan yang dirancang untuk perangkat dengan keterbatasan sumber daya, seperti IoT. Fokus utama penelitian ini adalah efisiensi serta ketahanan algoritma block cipher dan stream cipher terhadap serangan, sekaligus memastikan bahwa algoritma tetap memenuhi kebutuhan perangkat dengan keterbatasan daya, memori, dan komputasi [19].

Berdasarkan berbagai penelitian tersebut, penelitian ini mengadopsi pendekatan evaluasi yang mempertimbangkan keamanan, efisiensi, dan keterbatasan sumber daya perangkat IoT. Dengan demikian, instrumen yang digunakan dalam penelitian ini dirancang agar selaras dengan standar evaluasi yang telah dikaji dalam berbagai studi sebelumnya.

TABEL II
INSTRUMEN PERBANDINGAN

No	Algoritma	Keterangan	Pengujian	Bobot
1	Keamanan	Uji keamanan algoritma dalam menghadapi serangan	Differential Cryptanalys	25%
2	Kecepatan	Uji kecepatan enkripsi dan dekripsi	Kode dalam bahasa C	25%
3	Penggunaan CPU	Uji penggunaan CPU saat proses enkripsi dan dekripsi	Perf stat	25%
4	Penggunaan Memory	Uji penggunaan memory saat proses enkripsi dan dekripsi	Glances	25%
5	Konsumsi Daya	Analisis penggunaan arus saat proses enkripsi dan dekripsi	USB *power meter*	25%

III. HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk mengevaluasi performa algoritma PRESENT64/128, SIMON64/128, SPECK64/128, XTEA64/128, AES128, dan pada perangkat Raspberry Pi dalam berbagai aspek yang berkaitan dengan efisiensi dan keamanan. Aspek-aspek yang diuji mencakup keamanan, kecepatan eksekusi, penggunaan sumber daya CPU, penggunaan memory, serta konsumsi daya selama proses enkripsi dan dekripsi. Setiap aspek diuji secara menyeluruh dengan mengikuti serangkaian langkah yang telah dirancang guna memastikan hasil yang akurat.

A. *Pengujian Keamanan*

Pengujian keamanan dilakukan dengan menghitung *Differential Resistance Score (DRS)*, yang merupakan metrik gabungan untuk mengevaluasi ketahanan algoritma terhadap serangan *Differential Cryptanalysis*. DRS dihitung dengan mempertimbangkan empat aspek utama, yaitu *Avalanche Effect (AE)*, *Differential Uniformity (DU)*, *Bit Independence Coefficient (BIC)*, dan *Average Bit Change Ratio (ABCR)*[20][21]. Setiap aspek dalam perhitungan DRS memiliki bobot yang mencerminkan tingkat kepentingannya dalam menentukan ketahanan algoritma:

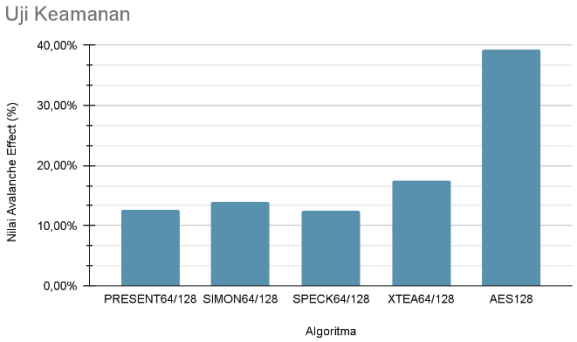
- *Avalanche Effect (AE)* – 30%: AE mengukur sejauh mana perubahan kecil pada plaintext menyebar dalam ciphertext. Semakin dekat nilainya ke 50%, semakin sulit bagi penyerang untuk menemukan pola dalam ciphertext.
- *Differential Uniformity (DU)* – 30%: DU mengukur keacakan perubahan ciphertext akibat perubahan pada plaintext. Nilai DU yang lebih rendah menunjukkan distribusi perbedaan yang lebih merata, yang berarti lebih sulit untuk dieksploitasi.
- *Bit Independence Coefficient (BIC)* – 20%: BIC mengukur independensi perubahan antar bit dalam ciphertext. Semakin tinggi nilai BIC (mendekati 1.0), semakin independen perubahan setiap bit, yang meningkatkan keamanan terhadap serangan diferensial.
- *Average Bit Change Ratio (ABCR)* – 20%: ABCR menunjukkan rata-rata jumlah bit dalam ciphertext yang berubah setiap kali ada perubahan di plaintext. Semakin tinggi ABCR, semakin baik penyebaran efek enkripsi, yang berkontribusi terhadap ketahanan algoritma.

Perhitungan DRS dilakukan dengan menggabungkan bobot dari masing-masing metrik untuk memberikan skor resistensi diferensial dalam skala 0 hingga 10. Skor ini memberikan gambaran tentang seberapa sulit algoritma dapat dianalisis menggunakan metode diferensial. Algoritma dengan DRS tinggi menunjukkan ketahanan lebih baik terhadap serangan, sedangkan algoritma dengan DRS rendah lebih rentan terhadap eksploitasi pola dalam ciphertext.

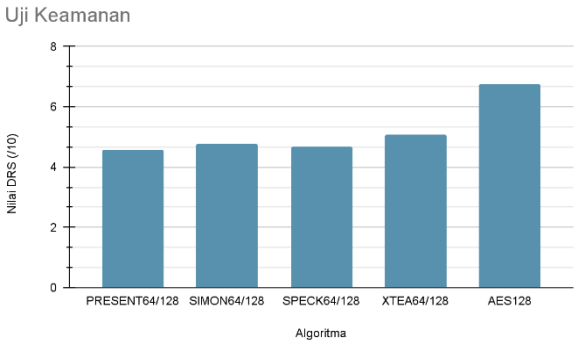
TABEL III
HASIL UJI KEAMANAN

No	Algoritma	Avalanche Effect (%)	DRS (/10)
1	PRESENT64/128	12.66%	4.57
2	SIMON64/128	14.06%	4.79
3	SPECK64/128	12.50%	4.67
4	XTEA64/128	17.58%	5.07
5	AES128/128	39.29%	6.76

Uji keamanan dilakukan dengan mengenkripsi 25 pasangan plaintext yang memiliki perbedaan 1 bit atau 1 byte menggunakan kunci yang sama. Hasil enkripsi kemudian dianalisis menggunakan metrik di atas untuk mendapatkan nilai DRS bagi setiap algoritma yang diuji. Hasil uji keamanan akan ditampilkan *Avalanche Effect* dan DRS pada Tabel III, Gambar 2, serta Gambar 3.



Gambar 2. Grafik *Avalanche Effect*



Gambar 3. Grafik DRS

Berdasarkan Tabel III, Gambar 2, serta Gambar 3, AES128 menunjukkan tingkat keamanan tertinggi terhadap serangan diferensial, dengan *Avalanche Effect* sebesar 39.29% dan Differential Resistance Score (DRS) sebesar 6.76/10. Hal ini menunjukkan bahwa perubahan kecil dalam plaintext menyebabkan perubahan yang luas dalam ciphertext, sehingga menyulitkan analisis pola melalui *Differential Cryptanalysis*.

Di antara algoritma enkripsi ringan, XTEA64/128 memiliki ketahanan terbaik, dengan AE sebesar 17.58% dan DRS 5.07/10, yang lebih tinggi dibandingkan PRESENT, SIMON, dan SPECK. Hal ini menunjukkan bahwa XTEA memiliki propagasi perubahan yang lebih baik dibandingkan algoritma ringan lainnya, meskipun masih jauh dari tingkat keamanan yang ditawarkan oleh AES128.

PRESENT64/128 memiliki nilai *Avalanche Effect* terendah (12.66%) dan DRS 4.57/10, yang merupakan skor ketahanan diferensial paling lemah dalam pengujian ini. Nilai ini menunjukkan bahwa perubahan kecil dalam plaintext tidak cukup menyebar dalam ciphertext, yang bisa menjadi kelemahan dalam konteks keamanan. SIMON64/128 dan SPECK64/128 memiliki performa yang relatif mirip, dengan DRS masing-masing 4.79/10 dan 4.67/10, serta *Avalanche Effect* yang tidak jauh berbeda.

Secara keseluruhan, hasil pengujian menunjukkan bahwa algoritma enkripsi ringan memiliki tingkat keamanan yang bervariasi, dengan XTEA sebagai pilihan terbaik di antara mereka, sementara AES128 tetap menjadi standar dengan tingkat keamanan tertinggi. PRESENT menunjukkan ketahanan diferensial paling rendah, yang bisa menjadi pertimbangan dalam memilih algoritma untuk aplikasi yang membutuhkan keamanan tinggi.

B. Pengujian Kecepatan

Pengujian kecepatan eksekusi dilakukan untuk menilai efisiensi setiap algoritma dalam melakukan proses enkripsi dan dekripsi. Pengujian ini mengukur waktu yang dibutuhkan untuk mengenkripsi dan mendekripsi data menggunakan fungsi pencatat waktu bahasa C.

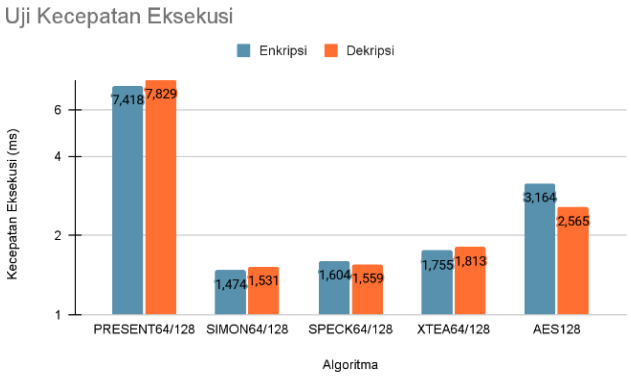
TABEL IV
HASIL UJI KECEPATAN ENKRIPSI

No	Algoritma	64 byte	512 byte	2048 byte	Rata-rata
1	PRESENT64/128	5.780 ms	7.075 ms	9.4 ms	7.418 ms
2	SIMON64/128	1.138 ms	1.504 ms	1.780 ms	1.474 ms
3	SPECK64/128	1.439 ms	1.612 ms	1.761 ms	1.604 ms
4	XTEA64/128	1.376 ms	1.455 ms	2.433 ms	1.755 ms
5	AES128/128	1.871 ms	2.135 ms	5.487 ms	3.164 ms

TABEL V
HASIL UJI KECEPATAN DEKRIPSI

No	Algoritma	64 byte	512 byte	2048 byte	Rata-rata
1	PRESENT64/128	5.787 ms	7.931 ms	9.769 ms	7.829 ms
2	SIMON64/128	1.192 ms	1.609 ms	1.793 ms	1.531 ms
3	SPECK64/128	1.417 ms	1.56 ms	1.7 ms	1.559 ms
4	XTEA64/128	1.311 ms	1.4 ms	2.728 ms	1.813 ms
5	AES128/128	1.598 ms	2.014 ms	4.083 ms	2.565 ms

Setiap algoritma diuji sebanyak 25 kali pada tiga variasi ukuran data input, yaitu 64 byte (mewakili komunikasi sensor IoT real time), 512 byte (mewakili komunikasi IoT periodik atau batch kecil), dan 2048 byte (mewakili transfer data log atau non-real-time). Nilai rata-rata dari seluruh iterasi dihitung untuk mendapatkan gambaran akurat mengenai performa algoritma dalam menangani data dengan ukuran berbeda. Pengujian ini bertujuan untuk mengevaluasi sejauh mana efisiensi algoritma dipengaruhi oleh kompleksitas operasionalnya dan seberapa cepat algoritma dapat memproses data dalam skenario penggunaan nyata. Hasil pengujian dapat dilihat pada Tabel IV, Tabel V, serta Gambar 4.



Gambar 4. Grafik Hasil Uji Kecepatan

Berdasarkan hasil uji kecepatan enkripsi (Tabel IV), dekripsi (Tabel V), dan Gambar 4, dapat disimpulkan bahwa algoritma SIMON64/128 memiliki performa terbaik dalam hal kecepatan, dengan rata-rata waktu eksekusi 1.474 ms untuk enkripsi dan 1.531 ms untuk dekripsi. Algoritma SPECK64/128 juga menunjukkan kinerja yang kompetitif dengan rata-rata 1.604 ms untuk enkripsi dan 1.559 ms untuk dekripsi.

Di sisi lain, PRESENT64/128 adalah algoritma dengan waktu eksekusi paling lama, baik dalam proses enkripsi (7.418 ms) maupun dekripsi (7.829 ms), yang menunjukkan bahwa algoritma ini lebih lambat dibandingkan dengan yang lain dalam skenario pengujian ini. AES128/128 memiliki performa yang cukup baik dalam dekripsi (2.565 ms) tetapi lebih lambat dalam enkripsi (3.164 ms) dibandingkan algoritma berbasis SIMON dan SPECK.

Secara umum, algoritma berbasis SIMON dan SPECK lebih unggul dalam hal kecepatan dibandingkan PRESENT dan AES, yang menunjukkan efisiensi mereka dalam lingkungan dengan keterbatasan sumber daya seperti IoT.

C. Pengujian Penggunaan CPU

Pengujian penggunaan CPU dilakukan untuk mengevaluasi efisiensi setiap algoritma dalam memanfaatkan sumber daya pemrosesan saat menjalankan proses enkripsi dan dekripsi. Pengukuran dilakukan dengan memantau tingkat penggunaan CPU selama eksekusi algoritma pada berbagai ukuran data input, yaitu 64 byte, 512 byte, dan 2048 byte.

Setiap pengujian dijalankan sebanyak 25 kali untuk memastikan konsistensi hasil dan mengurangi kemungkinan fluktuasi yang disebabkan oleh faktor eksternal. Data yang diperoleh dari pengujian ini memberikan gambaran mengenai seberapa besar beban yang ditimbulkan oleh masing-masing algoritma terhadap CPU, sehingga dapat digunakan untuk menilai efisiensi pemrosesan dalam berbagai skenario penggunaan.

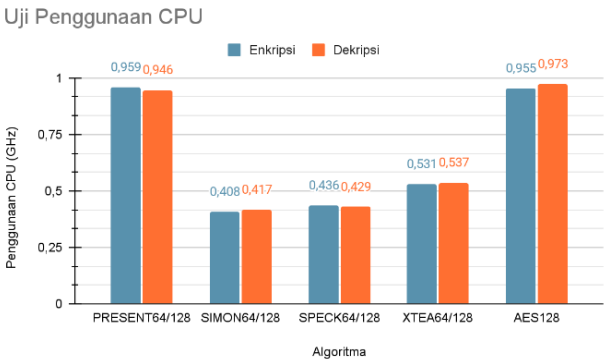
Hasil pengujian dapat dilihat pada Tabel VI, Tabel VII, serta Gambar 5 di bawah ini.

TABEL VI
HASIL UJI PENGGUNAAN CPU ENKRIPSI

No	Algoritma	64 byte	512 byte	2048 byte	Rata-rata
1	PRESENT64/128	0.561 GHz	1.063 GHz	1.252 GHz	0.959 GHz
2	SIMON64/128	0.161 GHz	0.358 GHz	0.704 GHz	0.408 GHz
3	SPECK64/128	0.178 GHz	0.369 GHz	0.761 GHz	0.436 GHz
4	XTEA64/128	0.205 GHz	0.443 GHz	0.945 GHz	0.531 GHz
5	AES128	0.551 GHz	1.063 GHz	1.252 GHz	0.955 GHz

TABEL VII
HASIL UJI PENGGUNAAN CPU DEKRIPSI

No	Algoritma	64 byte	512 byte	2048 byte	Rata-rata
1	PRESENT64/128	0.540 GHz	1.071 GHz	1.227 GHz	0.946 GHz
2	SIMON64/128	0.170 GHz	0.349 GHz	0.733 GHz	0.417 GHz
3	SPECK64/128	0.189 GHz	0.366 GHz	0.733 GHz	0.429 GHz
4	XTEA64/128	0.233 GHz	0.446 GHz	0.932 GHz	0.537 GHz
5	AES128	0.785 GHz	0.915 GHz	1.220 GHz	0.973 GHz



Gambar 5. Grafik Hasil Uji Penggunaan CPU

Berdasarkan hasil uji penggunaan CPU untuk proses enkripsi (Tabel VI), dekripsi (Tabel VII), dan Gambar 5, dapat disimpulkan bahwa algoritma SIMON64/128 memiliki efisiensi penggunaan CPU terbaik, dengan rata-rata penggunaan 0.408 GHz untuk enkripsi dan 0.417 GHz untuk dekripsi. SPECK64/128 juga menunjukkan efisiensi yang baik dengan rata-rata penggunaan 0.436 GHz untuk enkripsi dan 0.429 GHz untuk dekripsi.

Sebaliknya, algoritma PRESENT64/128 dan AES128 menunjukkan konsumsi CPU yang lebih tinggi. PRESENT64/128 menggunakan rata-rata 0.959 GHz untuk enkripsi dan 0.946 GHz untuk dekripsi, sementara AES128 menggunakan rata-rata 0.955 GHz untuk enkripsi dan 0.973 GHz untuk dekripsi. Hal ini menunjukkan bahwa kedua algoritma ini lebih berat dalam penggunaan CPU dibandingkan dengan SIMON dan SPECK.

Dari hasil ini, dapat disimpulkan bahwa SIMON64/128 dan SPECK64/128 lebih efisien dalam penggunaan CPU, menjadikannya lebih cocok untuk perangkat dengan sumber daya terbatas seperti IoT. AES128 dan PRESENT64/128, meskipun memiliki tingkat keamanan yang lebih tinggi, memerlukan daya komputasi yang lebih besar, sehingga lebih sesuai untuk perangkat dengan sumber daya yang lebih kuat.

D. Pengujian Penggunaan Memory

Pengujian penggunaan memori dilakukan untuk mengevaluasi efisiensi setiap algoritma dalam mengalokasikan dan memanfaatkan sumber daya memori selama proses enkripsi dan dekripsi. Pengukuran dilakukan dengan memantau konsumsi memori yang digunakan oleh setiap algoritma saat memproses data dengan berbagai ukuran input, yaitu 64 byte, 512 byte, dan 2048 byte.

TABEL VIII
HASIL UJI PENGGUNAAN MEMORI ENKRIPSI

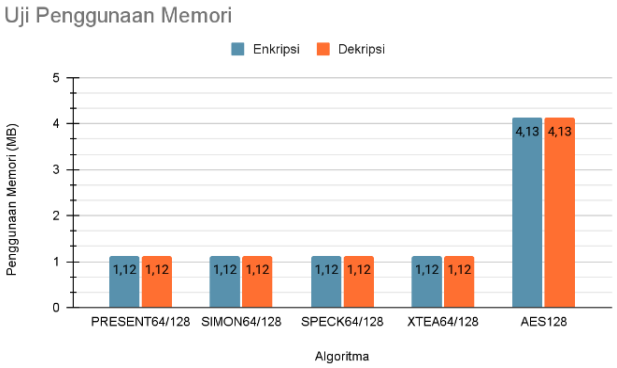
No	Algoritma	64 byte	512 byte	2048 byte	Rata-rata
1	PRESENT64/128	1.12 M	1.12 M	1.12 M	1.12 M
2	SIMON64/128	1.12 M	1.12 M	1.12 M	1.12 M
3	SPECK64/128	1.12 M	1.12 M	1.12 M	1.12 M
4	XTEA64/128	1.12 M	1.12 M	1.12 M	1.12 M
5	AES128	2.62 M	4.88 M	4.88 M	4.13 M

TABEL IX
HASIL UJI PENGGUNAAN MEMORI DEKRIPSI

No	Algoritma	64 byte	512 byte	2048 byte	Rata-rata
1	PRESENT64/128	1.12 M	1.12 M	1.12 M	1.12 M
2	SIMON64/128	1.12 M	1.12 M	1.12 M	1.12 M
3	SPECK64/128	1.12 M	1.12 M	1.12 M	1.12 M
4	XTEA64/128	1.12 M	1.12 M	1.12 M	1.12 M
5	AES128	2.62 M	4.88 M	4.88 M	4.13 M

Setiap pengujian dijalankan sebanyak 25 kali untuk memastikan konsistensi hasil serta mengurangi pengaruh fluktuasi yang mungkin disebabkan oleh faktor eksternal. Data yang diperoleh memberikan gambaran mengenai seberapa besar penggunaan memori oleh masing-masing

algoritma, sehingga dapat digunakan untuk menilai efisiensi dalam berbagai skenario implementasi. Hasil pengujian penggunaan memori untuk setiap algoritma dapat dilihat pada Tabel 8, Tabel 9, serta Gambar 6.



Gambar 6. Grafik Hasil Uji Penggunaan CPU

Berdasarkan hasil uji penggunaan memori untuk enkripsi (Tabel VIII), dekripsi (Tabel IX), dan Gambar 6, dapat disimpulkan bahwa algoritma PRESENT64/128, SIMON64/128, SPECK64/128, dan XTEA64/128 memiliki efisiensi memori yang sangat baik dan konsisten, dengan penggunaan memori tetap di 1.12 MB untuk semua ukuran data.

Sebaliknya, AES128 memiliki konsumsi memori yang jauh lebih tinggi dibandingkan algoritma lainnya, dengan penggunaan memori mencapai 2.62 MB hingga 4.88 MB, dan rata-rata 4.13 MB baik dalam proses enkripsi maupun dekripsi.

Dari hasil ini, dapat disimpulkan bahwa PRESENT, SIMON, SPECK, dan XTEA lebih ramah terhadap perangkat dengan keterbatasan memori, menjadikannya lebih cocok untuk implementasi pada sistem dengan sumber daya terbatas seperti IoT. AES128, meskipun lebih kuat dari segi keamanan, memiliki kebutuhan memori yang jauh lebih besar, sehingga lebih sesuai untuk perangkat dengan kapasitas memori yang lebih besar.

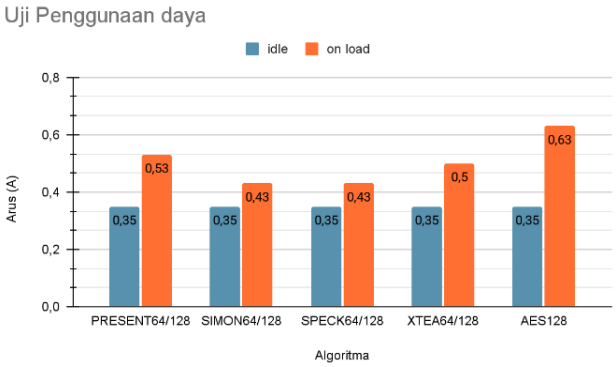
E. Pengujian Konsumsi Daya

Pengujian konsumsi daya dilakukan untuk mengevaluasi efisiensi energi dari setiap algoritma selama proses enkripsi dan dekripsi. Pengukuran dilakukan menggunakan perangkat USB power meter, yang mencatat arus yang digunakan selama eksekusi algoritma. Setiap iterasi enkripsi dan dekripsi dianalisis untuk memperoleh rata-rata arus per proses.

Pengujian ini dilakukan dengan parameter tetap, yaitu ukuran data input sebesar 64 byte. Untuk memastikan konsistensi hasil, setiap algoritma diuji sebanyak 25 kali. Hasil dari pengujian ini memberikan gambaran tentang efisiensi energi dari masing-masing algoritma. Hasil pengujian dapat dilihat pada Tabel X serta Gambar 7.

TABEL X
ALGORITMA YANG DIGUNAKAN

No	Algoritma	Tegangan	Arus Idle	Arus On Load
1	PRESENT 64/128	5.07 V	0.35 A	0.53 A
2	SIMON64/ 128	5.07 V	0.35 A	0.43 A
3	SPECK64/ 128	5.07 V	0.35 A	0.43 A
4	XTEA64/1 28	5.07 V	0.35 A	0.50 A
5	AES128	5.07 V	0.35 A	0.63 A



Gambar 7. Grafik Hasil Uji Penggunaan Daya

Analisis dalam Tabel X dan Gambar 7 menunjukkan bahwa Algoritma SIMON64/128 dan SPECK64/128 memiliki konsumsi arus terendah, yaitu 0.43 A, mengindikasikan efisiensi daya yang lebih baik dibandingkan algoritma lainnya. XTEA64/128 menggunakan sedikit lebih banyak daya dengan arus 0.50 A, sedangkan PRESENT64/128 memiliki konsumsi arus 0.53 A, lebih tinggi dari XTEA tetapi masih lebih rendah dibandingkan AES. AES128 menunjukkan konsumsi daya tertinggi dengan arus 0.63 A, mencerminkan beban pemrosesan yang lebih besar dibandingkan algoritma enkripsi ringan lainnya.

Hasil ini menunjukkan bahwa SIMON64/128 dan SPECK64/128 merupakan pilihan terbaik dalam hal efisiensi daya, sedangkan AES128, meskipun lebih kuat dalam aspek keamanan, memerlukan konsumsi daya yang lebih besar, yang kurang optimal untuk perangkat dengan keterbatasan sumber daya seperti IoT.

IV. KESIMPULAN

Berdasarkan hasil pengujian, pemilihan algoritma enkripsi terbaik untuk perangkat IoT bergantung pada keseimbangan antara keamanan dan efisiensi sumber daya. AES128 tetap menjadi pilihan terbaik dalam hal keamanan, dengan nilai Avalanche Effect tertinggi sebesar 39.29% dan DRS 6.76/10, menunjukkan ketahanan lebih baik terhadap serangan kriptografi dibandingkan algoritma ringan lainnya. Namun, AES128 memiliki kelemahan dalam konsumsi daya dan

memori yang lebih tinggi, sehingga kurang optimal untuk perangkat IoT dengan keterbatasan sumber daya.

Untuk perangkat dengan spesifikasi lebih kuat seperti Raspberry Pi, ada dua pendekatan yang dapat diambil. Jika keamanan maksimal menjadi prioritas, maka AES128 tetap merupakan pilihan terbaik, meskipun lebih berat dalam konsumsi daya dan sumber daya komputasi. Namun, jika yang diutamakan adalah efisiensi dan kecepatan, maka SIMON64/128 atau SPECK64/128 lebih direkomendasikan karena memiliki konsumsi daya lebih rendah (~0.43 A) dan penggunaan CPU yang lebih ringan (~0.42 GHz rata-rata), sementara tetap menawarkan keamanan yang cukup baik dibandingkan PRESENT atau XTEA.

Untuk perangkat IoT dengan sumber daya sangat terbatas, seperti ESP8266, ESP32, atau mikrokontroler berbasis ARM Cortex-M, penggunaan AES128 kurang disarankan karena terlalu berat dalam konsumsi memori dan daya prosesor, yang dapat mengurangi performa sistem secara keseluruhan. SIMON64/128 dan SPECK64/128 menjadi pilihan yang lebih realistis karena lebih ringan dan cepat. Namun, perlu diperhatikan bahwa algoritma ini memiliki indikasi potensi backdoor dari NSA, yang bisa menjadi risiko keamanan jangka panjang. Sebagai alternatif, algoritma seperti XTEA dapat dipertimbangkan, karena memiliki konsumsi daya yang masih rendah dan tidak memiliki indikasi potensi backdoor, meskipun tingkat keamanannya tetap perlu diuji lebih lanjut dalam skenario tertentu.

Secara keseluruhan, pemilihan algoritma harus disesuaikan dengan kebutuhan spesifik perangkat IoT, apakah lebih mengutamakan keamanan maksimal atau efisiensi daya dan kecepatan pemrosesan. Untuk perangkat berdaya tinggi seperti Raspberry Pi, AES masih menjadi opsi terbaik jika keamanan utama, sementara SIMON dan SPECK cocok untuk performa lebih ringan. Untuk perangkat berdaya rendah seperti ESP, SIMON dan SPECK lebih efisien tetapi memiliki potensi risiko keamanan, sehingga XTEA bisa menjadi alternatif kompromi antara keamanan dan efisiensi.

DAFTAR PUSTAKA

[1] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," IEEE Access, vol. 8, pp. 23022-23040, 2020.

[2] M. Usman, I. Ahmed, M. I. Aslam, S. Khan, and U. A. Shah, "SIT: a lightweight encryption algorithm for secure Internet of Things," arXiv preprint arXiv:1704.08688, 2017.

[3] L. Ertaul and A. Chauhan, "IoT Security: Implementation of XTEA, SIMON/SPECK Lightweight Block Ciphers," in 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), 2023, pp. 2478-2485.

[4] S. Maitra, D. Richards, A. Abdelgawad, and K. Yelamarthi, "Performance evaluation of IoT encryption algorithms: Memory, timing, and energy," in 2019 IEEE Sensors Applications Symposium (SAS), 2019, pp. 1-6.

[5] H. Kwon, Y. B. Kim, S. C. Seo, and H. Seo, "High-speed implementation of PRESENT on AVR microcontroller," Mathematics, vol. 9, no. 4, p. 374, 2021.

[6] R. Bharathi and N. Parvatham, "Light-Weight PRESENT Block Cipher Model for IoT Security on FPGA," Intelligent Automation & Soft Computing, vol. 33, pp. 35-49, 2022.

[7] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "SIMON and SPECK: Block ciphers for the Internet of Things," IACR Cryptol. ePrint Arch., p. 585, 2015.

[8] Y. Fatma, S. Soni, and M. Amseno, "Perbandingan algoritma SIMON dan SPECK dalam pengamanan citra digital," Jurnal CoSciTech (Computer Science and Information Technology), vol. 5, pp. 250-259, 2024.

[9] B. Y. Yustiarini, F. Dewanta, and H. H. Nuha, "A comparative method for securing Internet of Things (IoT) devices: AES vs. Simon-Speck encryptions," in 2022 1st International Conference on Information System & Information Technology (ICISIT), Yogyakarta, Indonesia, 2022, pp. 392-396, doi: 10.1109/ICISIT54091.2022.9872666.

[10] P. Panahi, C. Bayılmış, U. Çavuşoğlu, et al., "Performance evaluation of lightweight encryption algorithms for IoT-based applications," Arabian Journal for Science and Engineering, vol. 46, pp. 4015–4037, 2021. doi: 10.1007/s13369-021-05358-4.

[11] P. M. Ansyah, M. H. H. Ichsan, and A. Kusyanti, "Analisis performa algoritma SPECK pada Raspberry Pi," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 3, no. 1, pp. 1085–1092, 2018.

[12] V. Thakor, M. A. Razzaque, and M. Khandaker, "Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities," IEEE Access, vol. 9, pp. 28177-28193, 2021.

[13] P. Bright, "NSA-recommended IoT encryption algorithms are rubbish, says expert," The Register, Apr. 25, 2018. [Online]. Available: https://www.theregister.com/2018/04/25/nsa_iot_encryption/. [Diakses: Mar. 17, 2025].

[14] M. Appel, A. Bossert, S. Cooper, T. Kussmaul, J. L. Löffler, C. Pauer, and A. Wiesmaier, "Block ciphers for the IoT – SIMON, SPECK, KATAN, LED, TEA, PRESENT, and SEA compared," Cryptology ePrint Archive, 2016.

[15] P. E. A. Adriaanse, "A comparative study of the TEA, XTEA, PRESENT and SIMON lightweight cryptographic schemes," Cyber Security Group, Department of Intelligent Systems, Delft University of Technology, 2021.

[16] International Organization for Standardization, "ISO/IEC 29192-2:2012—Information security—Lightweight cryptography—Part 2: Block ciphers," 2012. Available: https://www.iso.org/standard/78477.html.

[17] I. Makarenko, S. Semushin, S. Suhai, S. M. Kazmi, A. Oracevic, and R. Hussain, "A comparative analysis of cryptographic algorithms in the Internet of Things," in 2020 International Conference on Innovations in Intelligent Systems and Applications (INISTA), 2020, pp. 1-8.

[18] P. S. Suryateja and K. V. Rao, "A survey on lightweight cryptographic algorithms in IoT," Cybernetics and Information Technologies, vol. 24, no. 1, pp. 21-34, 2024.

[19] N. M. Naser and J. R. Naif, "A systematic review of ultra-lightweight encryption algorithms," International Journal of Nonlinear Analysis and Applications, vol. 13, no. 1, pp. 3825-3851, 2022.

[20] Y. Y. Chan, C. Y. Khor, J. S. Teh, W. J. Teng, and N. Jamil, "Differential cryptanalysis of lightweight block ciphers SLIM and LCB," in Emerging Information Security and Applications. EISA 2022. Communications in Computer and Information Science, vol. 1641, J. Chen, D. He, and R. Lu, Eds. Cham, Switzerland: Springer, 2022. doi: 10.1007/978-3-031-23098-1_4.

[21] K. Mohamed, M. Nazran, F. Hani, and S. Ariffin, "Analyse on avalanche effect in cryptography algorithm," in Proceedings of the European Proceedings of Management and Science (EPMS), 2022. doi: 10.15405/epms.2022.10.57.