Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.039940
Article

Wake-Up Security: Effective Security Improvement Mechanism for Low Power Internet of Things

Sun-Woo Yun[1], **Na-Eun Park**[1] and **Il-Gu Lee**[1,2,*]

[1]Department of Future Convergence Technology Engineering, Sungshin Women's University, Seoul, 02844, Korea
[2]Department of Convergence Security Engineering, Sungshin Women's University, Seoul, 02844, Korea
*Corresponding Author: Il-Gu Lee. Email: iglee@sungshin.ac.kr

Received: 25 February 2023; Accepted: 18 May 2023; Published: 11 September 2023

Abstract: As time and space constraints decrease due to the development of wireless communication network technology, the scale and scope of cyber-attacks targeting the Internet of Things (IoT) are increasing. However, it is difficult to apply high-performance security modules to the IoT owing to the limited battery, memory capacity, and data transmission performance depending on the size of the device. Conventional research has mainly reduced power consumption by lightening encryption algorithms. However, it is difficult to defend large-scale information systems and networks against advanced and intelligent attacks because of the problem of deteriorating security performance. In this study, we propose wake-up security (WuS), a low-power security architecture that can utilize high-performance security algorithms in an IoT environment. By introducing a small logic that performs anomaly detection on the IoT platform and executes the security module only when necessary according to the anomaly detection result, WuS improves security and power efficiency while using a relatively high-complexity security module in a low-power environment compared to the conventional method of periodically executing a high-performance security module. In this study, a Python simulator based on the UNSW-NB15 dataset is used to evaluate the power consumption, latency, and security of the proposed method. The evaluation results reveal that the power consumption of the proposed WuS mechanism is approximately 51.8% and 27.2% lower than those of conventional high-performance security and lightweight security modules, respectively. Additionally, the latencies are approximately 74.8% and 65.9% lower, respectively. Furthermore, the WuS mechanism achieved a high detection accuracy of approximately 96.5% or greater, proving that the detection efficiency performance improved by approximately 33.5% compared to the conventional model. The performance evaluation results for the proposed model varied depending on the applied anomaly-detection model. Therefore, they can be used in various ways by selecting suitable models based on the performance levels required in each industry.

Keywords: Internet of Things; security; anomaly detection; low-power architecture; energy efficiency; wake-up security

1 Introduction

Internet of Things (IoT) technology collects, stores, and analyzes data using sensors and wired/wireless communication technologies on various objects, and refers to an interconnected device system [1]. All objects in the IoT have IDs for identification, communication, detection, and calculation functions. This technology can increase productivity and automate tasks in Industry 4.0. The IoT market is expected to grow to USD 131 billion by 2025. Moreover, the IoT network is expected to connect more than 75 billion devices through its use as an essential and efficient technology in various fields such as manufacturing, energy, home, transportation, disaster, and medical care [2].

With an increase in the number of IoT entities, billions of IoT devices are being connected to more extensive wireless networks. Therefore, a security solution for IoT is crucial because all the data sent and received are transmitted to another entity or server following collection and use. Particularly, because the confidentiality and integrity of data transmitted and received in the IoT must be maintained, IoT security and the protection of personal information are important issues that need to be resolved [3]. However, because most nodes composed of IoT devices operate on batteries, the data processing capacity is low, and the storage and bandwidth are limited, rendering the application of a complete security suite challenging [4]. Therefore, utilizing complex security solutions that perform data protection, authentication, and anomaly detection while maintaining the performance of core functions (e.g., continuous data collection and sharing and control of other devices) is difficult because of a lack of resources.

Conventionally, researchers have attempted to propose new security architectures tailored to data processing, power consumption, and battery life requirements or to use combinations of conventional security solutions to realize IoT security. However, these approaches are limited in terms of the data storage space and processing capabilities, which affect their efficiency in resource-constrained IoT environments. Furthermore, existing solutions have primarily focused on improving the resource supply systems or lightening of specific modules to reduce power consumption [5]. Although lightweight encryption algorithms such as Rivest–Shamir–Adleman (RSA), digital signature algorithms (DSA), and PRESENT have gained popularity with the rapid development of the IoT industry, these approaches have failed to provide a high-performance security suite that is essential in an increasingly complex network environment [6,7]. These studies sought to maximize the efficiency of IoT device resources in situations wherein the battery and computing power are constrained and security features are designed with technical limitations [8]. However, with the continuous increase in the number of objects being connected to wireless networks, the demand for a higher level of security is increasing, thus necessitating an energy-security-optimized solution that facilitates efficient power management while maintaining the system and security performance at a certain level. Therefore, research for the development of a low-power security architecture that can provide high-performance security solutions for IoT platforms is required.

In this paper, we propose a wake-up security (WuS). This low-power security architecture operates as a high-performance security module only when a threat is detected owing to the addition of anomaly detection logic to the IoT platform. It can improve the power efficiency of conventional methods that periodically execute the entire security module regardless of abnormal operations. This is because the proposed method consumes low power by periodically waking only the anomaly detection logic and consumes significant power only when necessary. Therefore, the WuS mechanism proposed in this study can solve the problem of lightweight solutions with limitations in security performance and scalability and provide high security performance and power efficiency suitable for IoT platforms and service levels. To evaluate the proposed WuS mechanism, the power consumption, latency, and security performance of the proposed module are measured, compared, and analyzed using a Python simulator based on the UNSW-NB15 dataset.

This study entailed the development of a novel approach for providing high-performance security solutions in resource-constrained IoT environments using the proposed WuS low-power security architecture. The primary contributions of this study are as follows:

- Novel methods for providing a high level of security capability without lightening the security modules in resource-constrained IoT environments were examined.
- Using real-world network traffic datasets, the proposed WuS architecture was evaluated and compared with conventional methods. The results demonstrate that WuS improves power efficiency and security performance over conventional approaches.

The remainder of this paper is organized as follows. Section 2 presents an overview of the IoT and the results of conventional research on IoT security vulnerabilities. Further, studies on conventional lightweight security techniques for resource-limited IoT security are compared and analyzed. Section 3 introduces WuS mechanisms. Thereafter, in Section 4, the performance of conventional and proposed WuS models is evaluated by conducting simulations in terms of power consumption, latency, and security performance. Finally, Section 5 presents the conclusions and future research directions.

2 Related Work

2.1 Internet of Things and Security

IoT is a term for the future of the Internet and ubiquitous computing and refers to the state in which humans and objects, including home appliances, vehicles, and machinery, are connected over the Internet [9]. IoT provides new applications through cooperation between intelligent sensors and objects without direct human intervention [10]. Machine-to-machine (M2M) technology, a connection procedure between the Internet, mobile environments, and intelligent devices, is classified as the foundation of the IoT. Moreover, the IoT can be built through a combination of radio frequency identification (RFID) and other sensors with everyday objects.

The purpose of the IoT is to authenticate all things and human connections without time and space constraints. Therefore, possible limitations such as security, communication, optimization, and legal rights must be addressed. Conventionally, to derive a security solution for the IoT, many studies have classified and analyzed its security vulnerabilities. Particularly, an attack surface is defined as a potential threat that allows unauthorized users to access a system and extract data. In general, attack surfaces can be classified as physical devices, networks, clouds, webs, and application interfaces [11]. The IoT may include hardware devices, such as RFID and sensing technologies, which supervise and interact with the processing between objects in real time. However, physical devices have limited resources depending on the size of the circuit and are prone to losses owing to natural disasters, simple accidents, and physical attacks. Wired and wireless networks that connect IoT devices to network technologies are essential for IoT systems. To provide users with a high level of service, the network scale and mobility between users and devices should be expanded. However, as the network service surface expands, it can be exposed to potential security threats, such as hacking, spoofing, denial of service (DoS), and man-in-the-middle attacks [12]. Furthermore, cloud computing technology enables remote access to shared service resources without time and space constraints and facilitates information collection and sharing. This technology can overcome resource limitations, which are significant constraints of IoT platforms [13] and can serve as a base technology for realizing the vision of IoT [14]. However, they are vulnerable to malicious attacks based on unauthorized access such as cross-site request forgery (CSRF), structured query language (SQL) injection, and cross-site scripting (XSS). Moreover, maintaining the integrity is challenging; thus, database leakage and privacy problems can occur. Moreover, in the case of web and application license interfaces, there is a risk of personal and sensitive information being abused, because services are provided through most remote access services and mobile devices. Thus, attacks such as malware, spyware, DoS, and wiretapping may occur. However, owing to the open nature of mobile operating systems, conducting a thorough security check via a third party is challenging [15,16]. Owing to the interconnected and interdependent characteristics of the IoT environment, vulnerabilities to new attacks that have not been discussed previously may be included. Table 1 summarizes IoT attack surfaces [11].

Table 1: IoT attack surfaces [11]

Attack surface	Description
Physical device	✓ Physical attacks ✓ Node capture attacks ✓ User tracking
Network services	✓ Wireless based attacks (e.g., jamming, DoS, man-in-the-middle attack, wormholes sinkhole) ✓ Routing attacks ✓ Attacks on networks ports ✓ Internet attacks (viruses, intrusion and hacking, DDoS, replay attack, identity theft)
Cloud services	✓ Malicious attacks (e.g., XSS, SQL injection flaws, cross-site request forgery (CSRF) and insecure storage) ✓ Attacks on data integrity vulnerability ✓ Privacy breaching attacks ✓ Insider attacks ✓ Flooding attacks
Web & application	✓ Malwares, spyware and virus ✓ DoS ✓ Eavesdropping ✓ Bluesnarfing, and bluejacking
Potential new attacks by exploiting	✓ Interconnected IoT environment ✓ Interdependent IoT environment ✓ Social IoT environment

To protect an IoT platform from the threats listed in Table 1, the potential threats must be identified and blocked through continuous monitoring. To respond to various aspects of security threats and provide security services, security mechanisms according to the types of security services are provided in prior art. Table 2 summarizes the types of security mechanisms for each IoT security service.

Table 2: Examples of security mechanisms for security services [6]

Security service [17]	Security mechanisms
Confidentiality	Message encryption, sign-encryption
Integrity	Hash function, message signature
Availability	Pseudo-random frequency hopping, access control, intrusion prevention systems, firewalls
Non-repudiation	Message signature
Authentication	Chain of hash, message authentication code
Privacy	Pseudonymity, unlikability, k-anonymity, Zero-Knowledge Proof (ZKP)

2.2 Lightweight Encryption Algorithm

In a conventional digital communication environment, transmitted and received data are encrypted and authenticated, and an encryption technique is primarily used to prevent data misuse and abuse. However, traditional encryption algorithms are unsuitable for resource-constrained IoT environments including physical devices. Consequently, lightweight encryption algorithms that can be used in low-power environments have been actively conducted [18]. Table 3 lists representative lightweight encryption algorithms.

Table 3: Lightweight encryption algorithms [19]

Encryption Algorithm	Block size	Key size
AES (Advanced Encryption Standard) [20]	128 bits	128, 192, 256 bits
ARIA (Academy, Research Institute, Agency) [21]	128 bits	128, 192, 256 bits
KLEIN [22]	64 bits	64, 80, 96 bits
PRESENT [23]	64 bits	80 bits
LEA (Lightweight Encryption Algorithm) [24]	128 bits	128, 192, 256 bits
CLEFIA [25]	128 bits	128, 192, 256 bits
KATAN [26]	32, 48, 64 bits	80 bits
QTL [27]	64 bits	64, 128 bits
ANU [28]	64 bits	80, 128 bits

In addition to the cryptographic algorithms listed in Table 3, numerous lightweight cryptographic algorithms exist and several studies on lightweight cryptographic algorithms are still in progress. Conventional lightweight encryption algorithms mainly improve standard lightweight encryption or propose new algorithms suitable for a specific environment. Moreover, a lightweight encryption algorithm is developed by reducing the structure and block size of the algorithm and simplifying the formula [29]. Because IoT devices are lightweight, several methods using ciphertext policy attribute-based encryption (CP-ABE) technology have been proposed to facilitate safe offloading of data into cloud environments [30]. In addition, various approaches have been proposed to enhance data security, such as revisiting traditional shift cipher techniques and developing secure cryptosystems using DNA cryptography and steganography [31,32]. Many studies have been conducted on methods to lighten the encryption algorithm to overcome the limitations of IoT resource constraints.

2.3 Security Solutions for Low-Power Environments

Several studies have been conducted on low-power security architectures or proposed a convergence security solution with other technologies, rather than lightening the traditional encryption algorithm and improving its performance. Table 4 summarizes the key technologies, limitations, and open issues in studies on security solutions in low-power environments. According to Table 4, four types of conventional low-power security solutions have been proposed: frameworks, architectures, protocols, and algorithms.

Table 4: Research on security solutions of low-power environment

Research	Type	Key technologies	Open issues
[33–35]	Framework	✓ Trust management framework that implements mixed and lightweight ciphers ✓ Intrusion detection system using software-defined network and deep learning	✓ The rules of the framework are dependent on the findings of the study ✓ Contains elements that degrade security performance ✓ Failure to consider responses after detecting abnormal behavior
[36–41]	Architecture	✓ Low-power, low-cost architecture with in-house modules ✓ Authentication scheme in a cloud computing environment	✓ Depending on a specific module ✓ Security was suggested by implementing authentication and identification steps, but the system complexity increased ✓ Performance improvement of security module is not considered
[42,43]	Protocol	✓ Low-power security protocol targeting distributed network environments and specific secure channels	✓ Leverage traditional lightweight cryptographic algorithms ✓ Research on memory management transactions due to data immutability is needed
[44–48]	Algorithm	✓ Energy-management algorithm for attacks on the IoT ✓ Mutual authentication algorithm ✓ Network and system compression algorithms ✓ Potential security threat prediction model	✓ Lack of research on security performance ✓ Focused on specific performance improvement for each algorithm

Regarding framework type, a security and privacy protection framework combined with multilevel trust management was proposed, and a user-centered privacy protection service was provided via the introduction of a hybrid encryption algorithm and lightweight encryption [33]. IoT-based healthcare systems have emerged as promising areas wherein patient data can be stored and transmitted securely [34]. This framework can provide the processing power to register personal health information (PHI) while limiting protection leakage in medical emergencies. However, momentary judgment errors may occur during opportunistic computing and support node-selection processes and may include factors that degrade security performance such as a short key with a low trust level. Intrusion detection systems that use software-defined networks with programmable approaches to separate the control and data planes have also been proposed [35]. Such systems were optimized to avoid burdening IoT devices; however, they were not considered for the response process after anomaly detection.

In contrast, in the case of low-power architecture, a self-developed sensor or authentication scheme in a cloud computing environment was proposed. To implement low-power, low-cost IoT networks for smart agriculture, the Indian Institute of Technology Hyderabad (IITH), remotely for monitoring soil moisture content, was used as the sink and sensor nodes to extend the system life by approximately 83% [36]. Additionally, through the introduction of cloud technology to IoT, security robustness is ensured based on the authentication system, and the communication costs are reduced [38,39]. However, the conventional low-power architecture solution has limitations such as an increase in the system complexity and latency because the results depend on a specific module or the introduction of a separate authentication system.

In the case of protocol solutions for low-power environments, low-power security protocols targeting a distributed network environment, including a blockchain or a specific security channel, have been proposed. A low-power blockchain protocol implemented by nodes in a blockchain network was proposed, and lightweight software that downloaded valid data structures was implemented [42]. Although this study can increase privacy while maintaining a constant communication cost, errors may occur because of the fork function of the blockchain and data characteristics owing to immutability. Furthermore, an energy-efficient security protocol for wireless sensor network (WSN) systems is proposed [43]. It comprises a mutual authentication mechanism and a symmetric security channel and is light weight with an AES encryption-based security channel. However, it offers weak reliability for a protocol with an executed log; thus, impersonation, eavesdropping, and spoofing attacks may occur.

Finally, in the case of low-power algorithms, energy-management algorithms for attacks targeting IoT, mutual authentication, and network and system compression have been proposed. The storage compression consensus (SCC) algorithm, which compresses the blockchain on each device to secure the storage capacity, reduces the storage capacity by 63% compared to the existing algorithms [44]. A secure and lightweight mutual authentication algorithm in the Dolev-Yao attack model [45] and an energy consumption management algorithm to improve the lifespan of IoT against battery consumption attacks have also been proposed [46]. These algorithms have proven their effectiveness on specific evaluation indicators such as memory capacity, security performance, and energy efficiency. However, they cannot degrade other performance evaluation indicators or utilize high-performance security. Furthermore, to predict potential security threats in IoT environments, researchers have explored algorithms such as the improved radial basis function neural network [47]. However, with the increasing complexity of IoT networks, adversaries have devised new ways to exploit system vulnerabilities; this has resulted in the development of abusive adversarial agents and attack strategies [48].

Conventional studies on security solutions for low-power environments can reduce overall power consumption and memory usage. However, reaching the level at which the optimal energy security solution is applied is challenging because the security performance deteriorates when security-related logic is not considered or when the weight is reduced. Particularly, conventional solutions have prioritized lightweight logic that performs a security function, such as a cryptographic algorithm, among many types of logic, which is insufficient to defend against cyberattacks occurring on a large-scale network. Moreover, in a conventional IoT platform, computing performance, such as data processing, is more critical than security performance. Consequently, many studies have highlighted or did not consider the security module. To overcome these limitations, this study simultaneously improves the power efficiency and security performance by applying high-performance security modules in a low-power environment.

3 Proposed Method

3.1 Wake-Up Security Architecture

The WuS mechanism adds a small amount of logic to perform anomaly detection on the IoT platform and operates the security module only when necessary to utilize the high-performance security module in a low-power environment. Fig. 1 shows a structural diagram of the WuS mechanism.

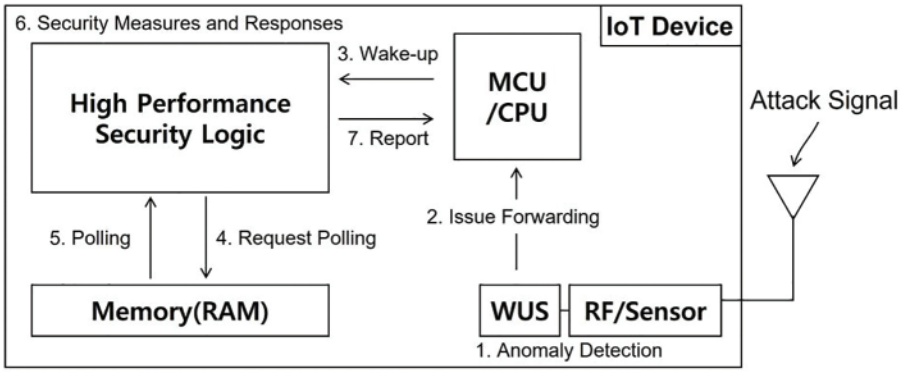

Figure 1: Wake-up security mechanism

WuS is located within the input part of the IoT platform and includes a microcontroller unit (MCU) corresponding to the control logic, a sensor that receives external signals, a security module that performs security functions, and memory. WuS logic is separated from security modules, which perform security functions and detect abnormal behavior by receiving data from sensor units that collect external signals. The total power efficiency can be improved by waking up a high-performance security module only when an abnormal behavior occurs. In other words, when no abnormal behavior occurs, the other logics are switched to the power-saving mode, and only the WuS logic remains active. For anomaly detection technology, studies on realistic datasets and learning methods have been actively conducted [49], and there may be differences in performance depending on the machine-learning algorithms and datasets. The anomaly-detection model used in the WuS logic is not generated by the IoT platform itself. Instead, it uses an external platform with sufficient resources to perform data preprocessing, normalization, feature extraction, dataset training, segmentation, and classification. The generated model can be updated continuously depending on the situation. In this study, an anomaly-detection model was constructed using a decision-tree classifier learned from the UNSW-NB15 dataset. The model performed pattern-based detection based on 15 highly correlated label values among the traffic information. Moreover, the anomaly-detection model can be applied by selecting a conventional anomaly-detection model based on IoT functions and user preferences.

Because the already learned model was added to the WuS logic to detect it, it is possible to detect it in advance with very little energy compared with conventional high-performance and lightweight security modules. Moreover, the saved memory space can only store and manage logs containing meaningful information, making it easier to track and manage security audits and control histories.

3.2 Operation of WuS Mechanism

According to the operating principle of the WuS mechanism, the flowchart comprises an abnormal-behavior detection step and a security check step. A flowchart of the operational structure of the WuS mechanism is shown in Fig. 2.

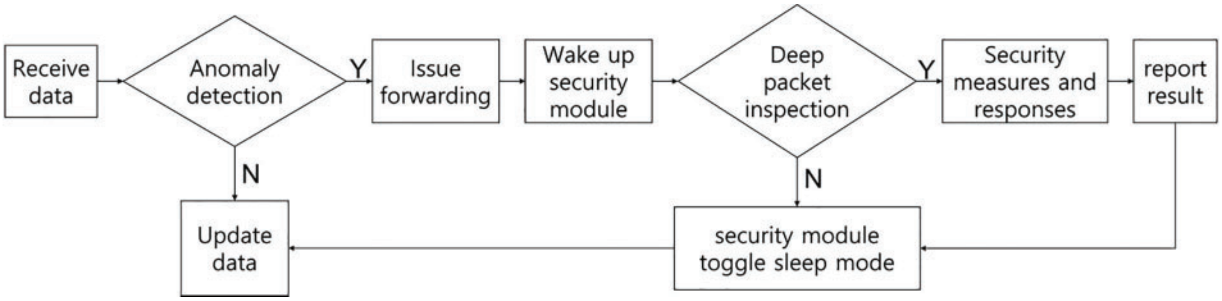

Figure 2: WuS mechanism flow chart

The operation of the IoT, including WuS logic, is as follows: The IoT periodically or irregularly receives signals from the outside. The signal data received by the sensor mounted on the IoT enter the abnormal-behavior detection stage using WuS logic. The WuS logic, which performs anomaly detection in the WuS mechanism, may utilize conventional anomaly detection techniques, such as attack pattern-based detection of known attacks, transmission signal anomaly detection, and integrity detection of transmitted and received data. That is, all the received signals are classified according to the outlier criteria of the anomaly-detection model applied to the WuS logic. After the detection of abnormal behavior, such as an abnormality in integrity violation, the WuS logic calls the high-performance security module to perform security measures corresponding to additional detection and response. Anomaly detection using the WuS logic can be performed even when the security module operates. In contrast, if no abnormal behavior occurs, the changes generated by the received traffic are updated and saved without invoking a separate security module. This method has a lower latency than the conventional method, which periodically scans through the security module regardless of abnormal behavior, and enables a quick response to additional threats. Particularly, it is possible to efficiently detect abnormal behaviors and manage response logs by saving resources in environments where many events do not occur.

The specifications of the security module are as follows: When the logic detects abnormal behavior, it reports the abnormal detection results to the MCU, corresponding to the control logic. Subsequently, the security module is activated after receiving an anomaly detection event from WuS. In this case, the security module may differ depending on the service and industry provided by the IoT platform. In this study, we assumed that deep packet inspection (DPI) is used for the security function of the security module. Generally, DPI involves analyzing data in depth by checking whether the entire string matches the received data [50]. However, because of the problem of complicating the signature depending on the data format, data inspection is performed by configuring a regular expression for a network intrusion detection system. Most DPI applications are based on pattern-based checking using a finite-state machine (FSM) to recognize languages expressed by regular expressions and use signature matching [51]. The security module that performs DPI maintains the power-saving mode during typical times in the absence of attacks. However, after the detection of abnormal behavior, it is awakened by the control logic. Consequently, the security module switches to the active state, performs DPI to determine whether an actual attack has occurred and the level of intrusion, and transmits it to an external server to implement countermeasures. This method can improve the performance of the IoT security architecture compared with the conventional method of processing all data in external networks using cloud technology. Furthermore, by limiting the scope of data leakage, the WuS mechanism can prevent secondary damage such as the leakage of personal and sensitive information. Following the completion of all security-related tasks, the security module reports problems and measures to the MCU and then returns to sleep mode. The control logic updates and patches the IoT devices and memory.

The use of energy-efficient mechanisms is crucial in resource-constrained IoT environments where energy consumption is directly related to the system complexity and circuit size of embedded systems. The WuS mechanism utilizes a power-saving mode when no abnormal behavior is detected following the completion of security work, resulting in lower power and memory consumption. Compared to conventional methods of reducing the complexity of modules that perform specific functions, the WuS mechanism can provide high-quality services for IoT platforms by reducing unnecessary power consumption. Thus, relatively complex security solutions or energy-security optimal solutions can be utilized even in resource-constrained IoT environments. In this way, the WuS mechanism is expected to enable the implementation of high-performance security functions at the level of a complete security suite, which is necessary for a highly secure IoT network.

4 Evaluation

4.1 Evaluation Environment

4.1.1 Experimental Environment

In this study, the power consumption, latency, and security performances of the proposed and comparative models were measured, compared, and evaluated using a Python-based simulator. The simulation was conducted in a PC environment using Windows 10 Home, RAM 8 GB, and 11th Gen Intel(R) Core(TM) i5–1135G7 @ 2.40 GHz. The simulation was evaluated using the "UNSW_NB15_testing-set.csv" for the UNSW-NB15 public dataset. This dataset comprised 16,235 traffic events: 12,326 normal and 3,909 attack traffic events. The simulation measured the power consumption, latency, and security performance during the analysis of the entire dataset.

4.1.2 Dataset and Detection Model

In this study, the UNSW-NB15 dataset was used to detect abnormal behavior. This dataset was generated by capturing real network traffic using the IXIA PerfectStorm tool of the CyberScope Laboratory of the Australian Cyber Security Center (ACCS), which consisted of normal and nine abnormal traffic [52]. Moreover, this dataset included nine attack types with 49 features: fuzzers, analysis, backdoors, DoS, exploits, generics, reconnaissance, shellcode, and worms. Therefore, the threat model used in this study was based on the UNSW-NB15 dataset, which provides a complex and dynamic environment for evaluating intrusion detection systems and security mechanisms in a network. In the simulation evaluation, an experiment was conducted based on binary classification depending on the presence or absence of an attack on normal and attack traffic using the "label" feature value.

Data preprocessing, encoding, and normalization were performed on the UNSW-NB15 dataset using the GitHub open-source code for the network intrusion detection system (NIDS) [53]. In the simulation evaluation, “UNSW_NB15_testing-set.csv” was used for the UNSW-NB15 dataset. The dataset consists of 45 features and 81,173 rows, excluding null values, with normal traffic accounting for 75.99%, and abnormal traffic accounting for 24.01%. A data frame with categorical properties was generated through one-hot encoding, and data normalization was performed using the MinMax Scaler for the generated data frame. Consequently, the binary dataset consisted of 81,173 rows and 61 columns, and the “label” features were classified as “normal” and “abnormal.” To extract other meaningful features, the linear relationship between two variables was analyzed using the Pearson correlation coefficient method. The larger the absolute value of the coefficient, the stronger the relationship between the variables. In this study, 15 features with a target feature label and correlation coefficient of 0.3 or higher were selected [53].

Simulations were performed using a normalized binary classification dataset. Among the binary datasets from which the feature extraction was completed, 80% and 20% were randomly partitioned into the training and test data, respectively, and a decision tree was used for training. A decision tree is a supervised learning model that classifies and regresses data using a set of rules. Because the resulting model has a tree structure, it has been used in many previous studies [54]. The tree-based binary classification results showed an accuracy of approximately 98.1, mean absolute error (MAE) and mean squared error (MSE) of approximately 0.019, and root mean square error (RMSE) of approximately 0.138. The trained basic dataset has a uniform distribution of normal and attack traffic. In this study, the distribution of attack traffic was arbitrarily adjusted for the simulation evaluation. The modified dataset is divided into three sections according to the number of attacks. The first, second, and third sections distributed 60%, 5%, and 35% of the total attack traffic, respectively. Fig. 3 shows the distribution of the actual and redistributed attack traffic obtained by calculating the cumulative amount of attack traffic.

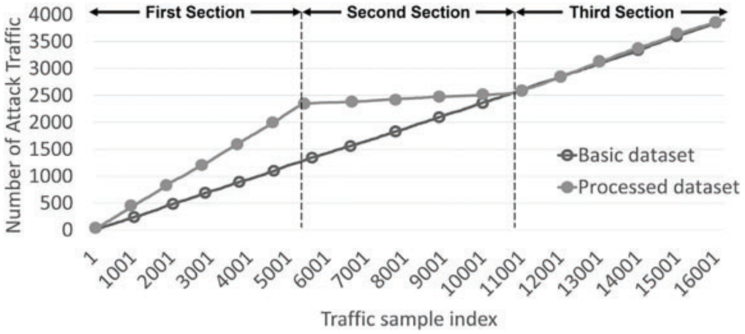

Figure 3: Distribution of attack traffic

4.1.3 Comparative Models and Assumptions

The simulation evaluation compared the proposed WuS mechanism with a high-performance lightweight security module-based mechanism. In the proposed model, the security module operates when anomalies are detected using the WuS logic, which detects abnormal behavior. In this case, the security module was the same as the high-performance security module of the conventional model. However, in the conventional model, the high-performance or lightweight security module operates periodically regardless of abnormal behavior. In this study, the entire dataset is evaluated as a single scenario, whereas in the conventional model, a security module is operated for each traffic event. Zero-day attacks were not considered in this experiment.

A conventional comparative model can be divided into two models that use only a high-performance security module or a lightweight security module. Although both high-performance and lightweight security modules perform DPI for every traffic event, the difference lies in the amount of power and computation consumed. The power consumption ratio of each logic is based on the complexity of the cryptographic algorithm. Complexity refers to the amount of resources required to complete a program after executing it and is generally proportional to the size of the input value. The larger the input value of the system, the more power it consumes [55]. High-performance security modules require greater memory and battery capacities, resulting in larger circuits. With an increase in the circuit size, the power consumption increases with complexity. Therefore, the higher the performance of the security module, the more complex the encryption algorithm. Table 5 presents a comparison of the complexities of the encryption algorithms [56,57].

Table 5: Comparison of complexities of encryption algorithm [56,57]

Algorithm	Before encryption	After encryption	After decryption
XOR	130 KB	130 KB	130 KB
DES (Data Encryption Standard)	130 KB	188 KB	130 KB
Triple-DES (TDES)	130 KB	360 KB	130 KB
Blowfish	130 KB	544 KB	130 KB

The power consumption ratio of each security module was determined based on the complexity after encryption, as illustrated in Table 5. WuS logic has a complexity value of XOR because it performs anomaly detection based on binary classification data. Blowfish, which has the highest complexity among the cryptographic algorithms listed in Table 5, is assumed to be the complexity value of the high-performance security module. Furthermore, the lightweight security module was divided into DES and TDES, and both lightweight models were included in the experiment. Table 6 lists the power consumption setting values for each security module.

Table 6: Power consumption of each security module [56,57]

Method	High-performance security module	Lightweight security module		WuS logic
Algorithm	Blowfish	DES	TDES	XOR
Total power	544 mW	188 mW	360 mW	130 mW

According to Table 6, the high-performance security module consumes 544 mW of power for each traffic event, whereas the lightweight security module consumes 188 and 360 mW. In contrast, the proposed model consumes 130 mW of power for each traffic event and operates the high-performance security module only when abnormal behavior is detected in the WuS logic, consuming an additional 544 mW of power.

4.2 Evaluation Results and Analysis

In this study, we present a comparative analysis of the proposed WuS mechanism against existing models that use high-performance and lightweight security modules. We conducted experiments to evaluate the power consumption, latency, and security performance of each model and analyzed the results. To evaluation power consumption, we calculated the power consumption ratio of each security module mathematically, based on the spatial complexity of the cryptographic algorithm, as shown in Table 6. For latency evaluation, we used a Python simulator implemented each comparison model to measure the operating time of the security module or logic according to the traffic flow. Finally, for security performance, we measured the detection accuracy and efficiency using well-known confusion matrices and machine-learning prediction accuracy formulas. The specific evaluation results are presented as below.

4.2.1 Power Consumption

Power comprises dynamic and static power, which can be expressed as follows [46]:

$$P = P_{static} + P_{dynamic} \tag{1}$$

Static power (P_{static}) refers to the constant leakage current present even when the circuit is disabled [58]. In contrast, the dynamic power ($P_{dynamic}$) has different values depending on the circuit capacity (C), voltage (V), and frequency (F). This can be expressed as [59]:

$$P_{dynamic} = C \times F \times V^2 \tag{2}$$

As presented in Table 6, the amount of power consumed by each security module was set based on the complexity value of the encryption algorithm. This shows that as the complexity of the module increases, the power consumption increases proportionally with the circuit size. Furthermore, in the simulation, the total power consumption was calculated by ignoring static power, and the power consumption for each model was calculated and measured. The length of each traffic flow or transaction is not reflected in the power consumption calculation.

Fig. 4 shows a graph of the accumulated power consumption for every 100 traffic cycles over time. “WuSM” represents the WuS Mechanism, whereas “HPSM” represents the high-performance security mechanism. Moreover, in the case of a lightweight security mechanism, “LSM1” has the complexity of DES, whereas “LSM2” has the complexity of TDES. LSM1 had the lowest total power consumption because it consumed power for every traffic type, and its value was similar to that of the proposed model. Furthermore, other models, except WuSM, consume power for all traffic; therefore, the power consumption increases faster over time. Particularly, the HPSM consumes the highest power increase with the steepest slope. However, the WuSM slope of the graph differed according to the attack frequency. This is because when WuSM detects abnormal behavior, the high-performance security module is activated and additional power is consumed. The power consumption increased with the steepest slope in the first section and highest attack frequency. Moreover, it can be observed that the power consumption increases with the gentlest slope in the second section, which has the lowest attack frequency. In terms of the total power consumption, WuSM consumed approximately 107.3% less power than HPSM, 37.2% less power than LSM2%, and 28.3% more power than LSM1. WuSM consumes the second-least power of the total power consumption. Based on these results, the WuS mechanism is expected to consume less power in an environment with fewer abnormal behaviors. Thus, WuS can improve the power efficiency of resource-constrained IoT platforms.

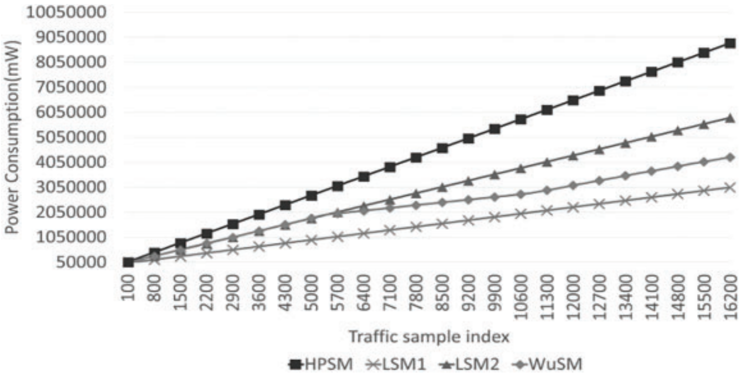

Figure 4: Evaluation results of accumulated power consumption

In addition to the accumulated power consumption, the instantaneous power consumption evaluation results are presented in Fig. 5.

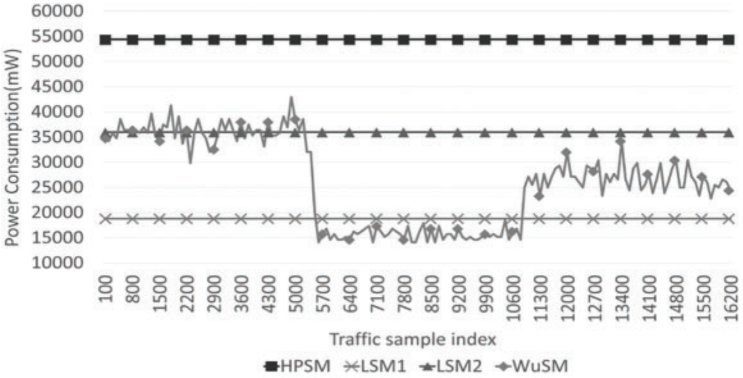

Figure 5: Evaluation results of instantaneous power consumption

Other models, except WuSM, draw a horizontal line because they always consume the same amount of power, regardless of the abnormal behavior. However, in the case of WuSM, it is evident that the difference in the instantaneous power consumption is large according to the attack frequency. The most power was consumed in the first section, whereas the least power was consumed in the second section. In the second section, the power consumption of WuSM is less than that of LSM2. The experiments revealed that the proposed WuS mechanism demonstrated improved power efficiency compared with conventional methods. This is because of the anomaly detection logic, which selectively activates high-power-consumption security modules only when an abnormal behavior is detected, thereby resulting in reduced power consumption. Compared with traditional models, which exhibit increased power consumption with the complexity of the module, WuS implemented a power consumption strategy that was dependent on the anomaly detection status, which improved the efficiency in module operations. The WuS mechanism is particularly useful in low-power environments with infrequent infringement situations because it improves efficiency via the reduction of power consumption upon the detection of normal behavior. Our results show that the proposed mechanism can provide a high level of security without reducing the security modules in resource-constrained IoT environments, as demonstrated by the improved power efficiency observed in our experiments using real-world network traffic datasets.

4.2.2 Latency

Latency refers to the time at which a security module and its related logic operate. In the simulation, the DPI, which operates the security module, was modeled as a mathematical operational device. The evaluation results for latency when comparing the accumulated execution time for every 100 traffic instances are shown in Fig. 6.

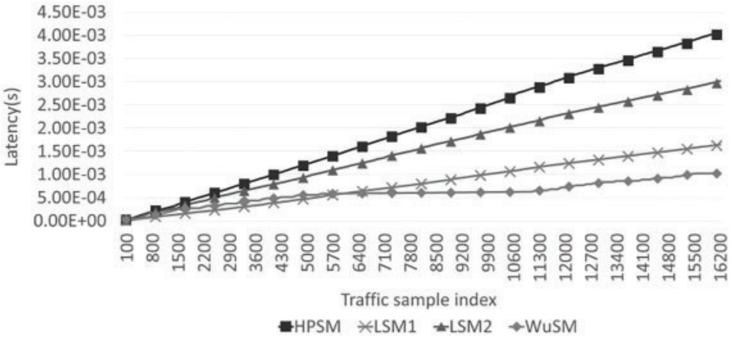

Figure 6: Evaluation results of accumulated latency

According to the evaluation results shown in Fig. 6, the total latency was the smallest for WuSM and highest for HPSM. The latencies of all the models, except WuSM, increased proportionally with time. However, the latency of WuSM differed in the slope of the rising graph according to the attack frequency. In the first section, the latency of WuSM increased faster than that of LSM2. Thereafter, in the second section, the latency of WuSM barely increased and increased again as it entered the third section. This is because WuS operates the security module when an abnormal behavior is detected in WuSM; the greater the latency, the greater the slope of the graph. As the amount of analyzed traffic increases toward the latter part of the dataset, the latency difference between each module gradually increases. Thus, based on these results, the WuS mechanism is expected to experience more minor delays in an environment with less abnormal behavior, thereby improving system efficiency and service availability. In addition to the latency, the instantaneous latency evaluation results are shown in Fig. 7.

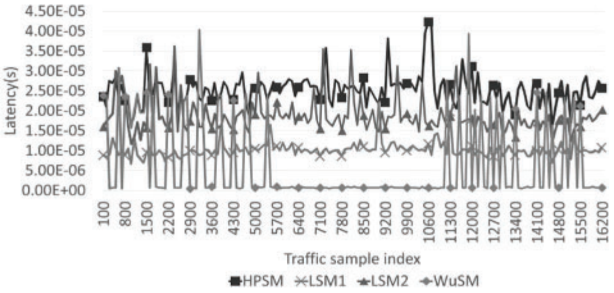

Figure 7: Evaluation results of instantaneous latency

Referring to the results in Fig. 7, the latency of WuSM in the first and third sections was longer than that of the HPSM, and, on average, the latency of LSM2 was the least required. WuSM had the shortest instantaneous latency in the second section, whereas the other models had longer latencies than the other sections. The experimental results clearly indicate that WuSM significantly affected the latency, with the degree of impact varying based on the attack frequency. This is because WuSM is designed to adapt the logic behavior based on anomaly detection, resulting in improved operational efficiency. Consequently, WuSM improves the latency performance compared to conventional security mechanisms by reducing the battery usage time and improving the processing speed in low-risk situations.

4.2.3 Security Performance

The security performance was evaluated by dividing it into detection accuracy and efficiency. In the evaluation, the average value was calculated by measuring the performance every 1,000 traffic cycles using the confusion matrix. The definition of the confusion matrix is the same as that in Table 7.

Table 7: Confusion matrix for evaluating security performance

	Normal (predict)	Abnormal (predict)
Normal (actual)	TP (True Positive: The number of cases correctly detected as normal traffic)	FN (False Negative: The number of cases incorrectly detected as abnormal traffic)
Abnormal (actual)	FP (False Positive: The number of cases incorrectly detected as normal traffic)	TN (True Negative: The number of cases correctly detected as abnormal traffic)

The detection accuracy is a measure of the probability that each model will succeed in detecting abnormal traffic; that is, it indicates the ratio of correctly detected abnormal traffic to all abnormal traffic in the dataset. The ratio was calculated using the following formula:

$$Detection\ Accuracy = \frac{TN}{Number\ of\ Abnormal\ Data} \times 100 \tag{3}$$

The detection efficiency is an evaluation index that checks the degree to which the operation of each model significantly impacts the IoT security. This confirms the accuracy of the model in predicting the normal or abnormal traffic. In other words, it measures how accurately each model can detect whether incorrect detection wastes power or increases the waiting time. This can be calculated using the following formula, which measures the prediction accuracy of machine learning [60]:

$$Detection\ Efficiency = \frac{TP + TN}{TP + TN + FN + FP} \times 100 \tag{4}$$

Fig. 8 shows the evaluation results for the detection accuracy calculated using Eq. (3) as an average for every 100 traffic cycles. Because all the models except WuSM perform DPI for all traffic types, they have the same detection accuracy and efficiency for the same dataset scenario. Therefore, in the security performance evaluation of this study, HPSM, LSM1, and LSM2 were unified and marked as "Conventional Methods (CM)."

According to the evaluation results, all models exhibited a high detection accuracy of 93% or higher for the entire section. Because zero-day attacks were not considered in this study, the detection accuracy of CM was 100%. However, WuSM may differ in detection accuracy depending on the anomaly-detection model. The detection accuracy of WuSM using the decision-tree algorithm was an average of approximately 96.5%. WuSM exhibited a relatively low and irregular detection accuracy in the first section; however, it increased as traffic was collected. In the third section, where the attack traffic increased again, the accuracy decreased; however, a high accuracy of more than 96% was achieved.

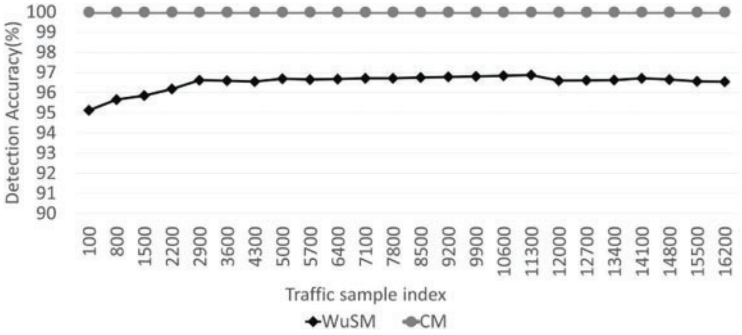

Figure 8: Evaluation results of detection accuracy

Fig. 9 shows the evaluation results calculated by averaging the detection efficiencies measured using Eq. (4), for every 100 traffic events.

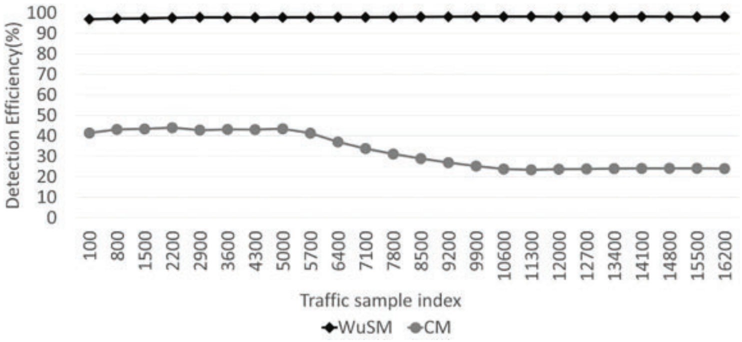

Figure 9: Evaluation results of detection efficiency

According to the evaluation results, WuSM achieved a high detection efficiency of approximately 98% on average. The detection efficiency may vary depending on the performance of the anomaly-detection model applied to WuSM. However, the CM operated a security module for each traffic event; thus, it operated as if it were detecting all the abnormal traffic. Consequently, the detection efficiency of CM was an average of approximately 32.8%, which was significantly lower than that of WuSM, and this figure decreased over time. Thus, as the amount of traffic increases, the detection efficiency of WuSM performing primary detection significantly increases. Although the security performance of the proposed scheme can vary depending on the specific anomaly-detection algorithms and models utilized in the WuS logic and high-performance security modules, the results show that in case of similar performance of the applied models, the detection accuracy of the proposed scheme and conventional methods are comparable. However, the detection efficiency was significantly improved with WuSM owing to its ability to minimize unnecessary behavior when operating security modules. This optimized efficiency ultimately results in improved security performance in resource-constrained IoT environments.

5 Conclusion and Future Work

This study devised a low-power security architecture, WuS, which presents a novel approach for providing high-performance security solutions in IoT environments with limited resources. In contrast to conventional methods that are reliant on lightening of specific security modules, WuS involves the addition of a small anomaly detection logic that can activate the high-performance security modules only when an attack is suspected. This approach significantly reduces the power consumption and improves the power efficiency. Consequently, the use of relatively complex security solutions or optimal energy security solutions is facilitated even in resource-constrained IoT environments. A key contribution of this study is the discussion on methods for providing high-level security capabilities without lightening the security modules. The proposed WuS architecture was evaluated using the UNSW-NB15 public dataset, and the results demonstrated its superiority over conventional methods in terms of power consumption, latency, and security performance. By leveraging a learned anomaly detection algorithm, WuS can effectively detect attacks while consuming minimal power, thus rendering it a promising solution for energy-efficient IoT security. The evaluation revealed that WuSM consumed approximately 107.3% less power than the HPSM and 37.2% less power than LSM2. Furthermore, the latency of WuSM was the lowest among all models, and the instantaneous latency was larger in the interval with a high attack frequency. Finally, among the security performance evaluation indicators, the detection accuracy achieved high values of 100% for CM and 96.5% for WuSM. The detection efficiency was improved by approximately 33.5%, with approximately 32.8% for CM and 98% for WuSM. The security performance of WuSM may vary depending on the performance of the anomaly-detection model. In future studies, a realistic model will be developed by adding an actual security module corresponding to a complete security suite. The evaluation in this study was designed based on the operating principle of the proposed architecture. However, implementing hardware and software modules with embedded systems will yield more realistic results. Moreover, by combining artificial intelligence, it is possible to research automatic data processing methods and select an anomaly-detection model suitable for flexible environmental changes in the IoT platform, field of application, and data characteristics. For example, when developing a machine-learning model, the proposed architecture can detect unknowns, not specific types of attacks, by variously thresholding the signatures for anomalous behaviors.

Acknowledgement: This study was revised and supplemented by a paper presented at the 6th International Symposium on Mobile Internet Security (MobiSec 2022).

Funding Statement: This work was partly supported by a National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (MSIT) (No. 2020R1F1A1061107), a Korea Institute for Advancement of Technology (KIAT) grant funded by the Korean Government (MOTIE) (P0008703, the Competency Development Program for Industry Specialists), and MSIT under the ICAN (ICT Challenge and Advanced Network of HRD) Program (No. IITP-2022-RS-2022-00156310) supervised by the Institute of Information and Communication Technology Planning and Evaluation (IITP).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Ashton, “That Internet of Things thing,” *RFID Journal*, vol. 22, no. 7, pp. 97–114, 2009.

[2] M. V. Vinayak and T. Jarin, "An overview of security issues in internet of things based smart environments," *EAI Endorsed Transactions on Energy Web*, vol. 9, no. 37, pp. e5, 2021.

[3] A. U. Rehman, S. U. Rehman, I. U. Khan, M. Moiz and S. Hasan, "Security and privacy issues in IoT," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 8, no. 3, pp. 147–157, 2016.

[4] A. Kumar, M. Zhao, K. J. Wong, Y. L. Guan and P. H. J. Chong, "A comprehensive study of IoT and WSN MAC protocols: Research issues, challenges and opportunities," *IEEE Access*, vol. 6, pp. 76228–76262, 2018.

[5] Ł. Apiecionek, M. Großmann and U. Krieger, "Harmonizing IoT-architectures with advanced security features—A survey and case study," *Journal of Universal Computer Science*, vol. 25, no. 6, pp. 571–590, 2019.

[6] D. E. Kouicem, A. Bouabdallah and H. Lakhlef, "Internet of Things security: A top-down survey," *Computer Networks*, vol. 141, pp. 199–221, 2018.

[7] S. A. Ajagbe, A. O. Adesina, T. J. Odule and O. Aiyeniko, "Evaluation of computing resources consumption of selected symmetric-key algorithms," *The Journal of Computer Science and its Applications*, vol. 26, no. 2, pp. 64–76, 2020.

[8] O. Yousuf and R. N. Mir, "A survey on the Internet of Things security: State-of-art, architecture, issues and countermeasures," *Information & Computer Security*, vol. 27, no. 2, pp. 292–323, 2019.

[9] A. Shah and M. Engineer, "A survey of lightweight cryptographic algorithms for IoT-based applications," in *Proc. ICSICCS-2018*, Singapore, Springer, vol. 851, pp. 283–293, 2019.

[10] R. P. Kumar and D. S. Smys, "A novel report on architecture protocols and applications in Internet of Things (IoT)," in *2018 2nd Int. Conf. on Inventive Systems and Control (ICISC)*, Coimbatore, India, pp. 1156–1161, 2018.

[11] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali *et al.,* "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[12] Y. Liu, C. Cheng, T. Gu, T. Jiang and X. Li, "A lightweight authenticated communication scheme for smart grid," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 836–842, 2016.

[13] M. Díaz, C. Martín and B. Rubio, "State-of-the-art, challenges, and open issues in the integration of Internet of Things and cloud computing," *Journal of Network and Computer Applications*, vol. 67, pp. 99–117, 2016.

[14] C. Stergiou, K. E. Psannis, B. G. Kim and B. Gupta, "Secure integration of IoT and cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 964–975, 2018.

[15] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. S. Gaur *et al.,* "Android security: A survey of issues, malware penetration, and defenses," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 998–1022, 2015.

[16] J. Huang, X. Zhang, L. Tan, P. Wang and B. Liang, "Asdroid: Detecting stealthy behaviors in android applications by user interface and program behavior contradiction," in *Proc. of the 36th ICSE*, New York, NY, USA, pp. 1036–1046, 2014.

[17] F. Dalipi and S. Y. Yayilgan, "Security and privacy considerations for IoT application on smart grids: Survey and research challenges," in *Proc. of 2016 IEEE 4th Int. Conf. on FiCloudW*, Vienna, Austria, pp. 63–68, 2016.

[18] N. A. Gunathilake, A. Al-Dubai and W. J. Buchana, "Recent advances and trends in lightweight cryptography for IoT security," in *2020 16th Int. Conf. on Network and Service Management (CNSM)*, Izmir, Turkey, pp. 1–5, 2020.

[19] S. H. Mun, M. U. Kim and T. G. Gwon, "Trends in lightweight encryption technology for IoT communication environments," *Information and Communications Magazine*, vol. 33, no. 3, pp. 80–86, 2016.

[20] J. Daemen and V. Rijmen, "*The Design of Rijndael: AES–The Advanced Encryption Standard*," DBLP, vol. 23, pp. 362–366, 2013.

[21] D. Kwon, J. Kim, S. Park, S. H. Sung, Y. Sohn *et al.,* “New block cipher: ARIA,” in *Information Security and Cryptology-ICISC 2003*, Seoul, Korea, pp. 432–445, 2004.

[22] Z. Gong, S. Nikova and Y. W. Law, “KLEIN: A new family of lightweight block ciphers,” in *RFID. Security and Privacy–7th Int. Workshop, RFIDSec 2011*, Amherst, USA, vol. 7055, pp. 1–18, 2011.

[23] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann *et al.,* “Present: An ultra-lightweight block cipher,” in *Cryptographic Hardware and Embedded Systems-CHES 2007: 9th Int. Workshop*, Vienna, Austria, pp. 450–466, 2007.

[24] D. Hong, J. K. Lee, D. C. Kim, D. Kwon, K. H. Ryu *et al.,* “LEA: A 128-bit block cipher for fast encryption on common processors,” in *Information Security Applications: 14th Int. Workshop*, Jeju Island, Korea, pp. 3–27, 2014.

[25] T. Shirai, K. Shibutani, T. Akishita, S. Moriai and T. Iwata, “The 128-bit blockcipher CLEFIA,” in *Proc. Fast Software Encryption*, Luxembourg, Luxembourg, pp. 181–195, 2007.

[26] T. Eisenbarth, Z. Gong, T. G¨uneysu, S. Heyse, S. Indesteege *et al.,* “Compact implementation and performance evaluation of block ciphers in ATtiny devices,” in *Proc. of Cryptology—AFRICACRYPT 2012*, Ifrane, Morocco, Africa, pp. 172–187, 2012.

[27] L. Li, B. Liu and H. Wang, “QTL: A new ultra-lightweight block cipher,” *Microprocessors and Microsystems*, vol. 45, pp. 45–55, 2016.

[28] G. Bansod, A. Patil, S. Sutar and N. Pisharoty, “An ultra-lightweight encryption design for security in pervasive computing,” in *2016 IEEE 2nd Int. Conf. on BigDataSecurity, HPSC and IDS*, New York, NY, USA, pp. 79–84, 2016.

[29] S. Singh, P. K. Sharma, S. Y. Moon and J. H. Park, “Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions,” *Ambient Intelligence and Humanized Computing*, pp. 1–18, 2017. https://link.springer.com/article/10.1007/s12652-017-0494-4

[30] S. Das and S. Namasudra, “MACPABE: Multi-authority-based CP-ABE with efficient attribute revocation for IoT-enabled healthcare infrastructure,” *International Journal of Network Management*, pp. e2200, 2022. https://onlinelibrary.wiley.com/doi/full/10.1002/nem.2200

[31] R. Verma, A. Kumari, A. Anand and V. S. S. Yadavalli, “Revisiting shift cipher technique for amplified data security,” *Journal of Computational and Cognitive Engineering*, 2022. https://doi.org/10.47852/bonviewJCCE2202261

[32] S. Namasudra, “A secure cryptosystem using DNA cryptography and DNA steganography for the cloud-based IoT infrastructure,” *Computers and Electrical Engineering*, vol. 104, pp. 108426, 2022.

[33] P. Guo, J. Wang, S. Ji, X. H. Geng and N. N. Xiong, “A lightweight encryption scheme combined with trust management for privacy-preserving in body sensor networks,” *Medical Systems*, vol. 39, no. 12, pp. 190, 2015.

[34] F. J. Abdullayeva, “Internet of things–based healthcare system on patient demographic data in Health 4.0,” *CAAI Transactions on Intelligence Technology*, vol. 7, no. 4, pp. 644–657, 2022.

[35] A. Wani and R. Khaliq, “SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL),” *CAAI Transactions on Intelligence Technology*, vol. 6, no. 3, pp. 281–290, 2021.

[36] S. Heble, A. Kumar, K. V. V. D. Prasad, S. Samirana, P. Rajalakshmi *et al.,* “A low power IoT network for smart agriculture,” in *2018 IEEE 4th WF-IoT*, Singapore, pp. 609–614, 2018.

[37] D. Y. Kim and M. W. Jung, “Data transmission and network architecture in long range low power sensor networks for IoT,” *Wireless Personal Communications*, vol. 93, no. 1, pp. 119–129, 2017.

[38] L. Zhou, X. Li, K. H. Yeh, C. Su and W. Chiu, “Lightweight IoT-based authentication scheme in cloud computing circumstance,” *Future Generation Computer Systems*, vol. 91, pp. 244–251, 2019.

[39] R. Martínez-Peláez, H. Toral-Cruz, J. R. Parra-Michel, V. García, L. J. Mena *et al.,* “An enhanced lightweight IoT-based authentication scheme in cloud computing circumstances,” *Sensors*, vol. 19, no. 9, pp. 2098, 2019.

[40] S. J. Yu, K. S. Park and Y. H. Park, “A secure lightweight three-factor authentication scheme for IoT in cloud computing environment,” *Sensors*, vol. 19, no. 16, pp. 1–20, 2019.

[41] S. Das and S. Namasudra, “A lightweight and anonymous mutual authentication scheme for medical big data in distributed smart healthcare systems,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–12, 2022.

[42] P. Danzi, A. E. Kalør, Č. Stefanović and P. Popovski, “Delay and communication tradeoffs for blockchain systems with lightweight IoT clients,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2354–2365, 2019.

[43] M. T. Hammi, E. Livolant, P. Bellot, A. Serhrouchni and P. Minet, “A lightweight IoT security protocol,” in *Proc. of IEEE 1st CSNet*, Rio de Janeiro, Brazil, pp. 1–8, 2017.

[44] T. Kim, J. Noh and S. Cho, “SCC: Storage compression consensus for blockchain in lightweight IoT network,” in *Proc. of IEEE ICCE*, Las Vegas, NV, USA, pp. 1–4, 2019.

[45] A. Adeel, M. Ali, A. N. Khan, T. Khalid, F. Rehman *et al.,* “A multi-attack resilient lightweight IoT authentication scheme,” *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, pp. e3676, 2022.

[46] B. J. Mohd and T. Hayajneh, “Lightweight block ciphers for IoT: Energy optimization and survivability techniques,” *IEEE Access*, vol. 6, pp. 35966–35978, 2018.

[47] Z. Chen, “Research on internet security situation awareness prediction technology based on improved RBF neural network algorithm,” *Journal of Computational and Cognitive Engineering*, vol. 1, no. 3, pp. 103–108, 2022.

[48] V. S. Gaur, V. Sharma and J. McAllister, “Abusive adversarial agents and attack strategies in cyber–physical systems,” *CAAI Transactions on Intelligence Technology*, vol. 8, no. 1, pp. 149–165, 2023.

[49] Á. L. P. Gómez, L. F. Maimó, A. H. Celdrán, F. J. G. Clemente, C. C. Sarmiento *et al.,* “On the generation of anomaly detection datasets in industrial control systems,” *IEEE Access*, vol. 7, pp. 177460–177473, 2019.

[50] G. De La Torre Parra, P. Rad and K. K. R. Choo, “Implementation of deep packet inspection in smart grids and industrial Internet of Things: Challenges and opportunities,” *Network and Computer Applications*, vol. 135, pp. 32–46, 2019.

[51] C. Xu, S. Chen, J. Su, S. M. Yiu and L. C. K. Hui, “A survey on regular expression matching for deep packet inspection: Applications, algorithms, and hardware platforms,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2991–3029, 2016.

[52] N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive dataset for network intrusion detection systems (UNSW-NB15 network dataset),” in *Proc. of IEEE MilCIS*, Canberra, Australia, pp. 1–6, 2015.

[53] A. Dubey, “IoT-network-intrusion-detection-system-UNSW-NB15,” 2021. [Online]. Available: https://github.com/abhinav-bhardwaj/IoT-Network-Intrusion-Detection-System-UNSW-NB15

[54] J. H. Joloudari, M. Haderbadi, A. Mashmool, M. Ghasemigol, S. S. Band *et al.,* “Early detection of the advanced persistent threat attack using performance analysis of deep learning,” *IEEE Access*, vol. 8, pp. 186125–186137, 2020.

[55] J. B. Gross, D. Jacoby, K. Coogan and A. Helman, “Motivating complexity understanding by profiling energy usage,” in *Proc. of the 2021 ACM SIGPLAN Int. Symp. on New Ideas, New Paradigms, and Reflections on Programming and Software*, Chicago, IL, USA, pp. 85–96, 2021.

[56] V. Singh and S. K. Dubey, “Analysing space complexity of various encryption algorithms,” *International Journal of Computer Engineering and Technology (IJCET)*, vol. 4, no. 1, pp. 414–419, 2013.

[57] J. M. Gnanasekar, “Light weight cryptographic algorithm to improve avalanche effect for data security using prime numbers and bit level operations,” *International Journal of Applied Engineering Research*, vol. 10, no. 21, pp. 41977–41983, 2015.

[58] K. Cushon, C. Leroux, S. Hemati, S. Mannor and W. J. Gross, “A min-sum iterative decoder based on pulsewidth message encoding,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 11, pp. 893–897, 2010.

[59] D. A. Neri, R. P. Medina and A. M. Sison, “An Xbox-based key generation technique for vigenere algorithm,” in *Proc. of the 3rd ICCSP*, New York, NY, USA, pp. 66–70, 2019.

[60] N. Moustafa, “Designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic,” Ph.D. Dissertation, University of New South Wales, Australia, 2017.