ISSN 2063-5346

LWES: THE EFFICIENCY OF NEW LIGHTWEIGHT ENCRYPTION FRAMEWORK FOR INTERNET OF THINGS SECURITY

Ahmed Mohamed Maher[1], S.Suganya[2]

Article History: Received: 01.02.2023 Revised: 07.03.2023 Accepted: 10.04.2023

Abstract

Because of the technological invasion in all areas of life (medical, agricultural, industrial, societal) etc., it has made the Internet of things an indispensable technology in terms of advantages and flexibility. But because of this huge expansion, there has also been a major flaw in terms of the security of mobile information between billions of devices, in addition to attacks and threats to the Internet of Things. This study proposes a Lightweight Encryption Security (LWES) Framework have three- phases for securing, the first phase Registration is Establish a secure connection between the server and the device. The second phase Authentication is the server verifying the authentication of the connected devices. Third phase is securing data by using (LWES). The algorithm is a 16-byte block cypher and wants the data to be encrypted using a 16-byte (128-bit) key for the Internet of Things security. It raises the level of security and improves network functionality by utilizing energy-efficient techniques. Cooja simulator helps in implementing and comparing the performance of the proposed mechanism with existing mechanisms such as the Objective Security Framework and Constrained Application Protocol Even in the existence of attackers, Lightweight Encryption Security Framework beats Objective Security Framework and Constrained Application Protocol in terms of throughput with low computational, storage, and energy overhead.

Keywords: IoT; authentication; confidentiality; cryptography; security framework.

[1]Mcs computer sciences Rathinavel Subramaniam college of arts and science, Coimbatore, India

[2]Mca, Ph.D, Asst. Prof Department of computer sciences Rathinavel Subramaniam college of arts and science, India

DOI:10.31838/ecb/2023.12.s1-B.312

1. Introduction

The Internet of Things (IoT) brings a fundamental shift to consumers' lifestyles [1]. Each IoT-connected gadget functions in an intelligent way, making the world dependent on technology [2]. In many areas, IoT works, for example, in inventories, healthcare, and smart houses [3]. Users, therefore, expect the IoT to offer strong security and confidentiality, which requires a safety framework [4]. The security solutions included in the IoT are vulnerable to attacks like denial of service (D0S), spoofing, and more [5]. An evaluation of its security aspects like authentication, or confidentiality in this situation is carried out in a security framework [6, 7]. The next step is to examine whether or not the data collected are authentic [8]. The latest assessment measures for safety judgments. If the safety requirements are satisfied and judgments can be taken in real-time in a security framework, [9] a competent framework may be established. A security framework tends to provide security for the entire system. These four layers—perception, connectivity, processing, and application layers—care for security at various levels shown in figure 1 The IoT comprises resource-controlled devices, such as the RFID, which is battery-operated sensors. Particular emphasis should therefore be taken to restrict and simultaneously guarantee the usage of their resources. Solutions for lightweight encryption offer both safety and performance. The following are the primary reasons for adopting lightweight IoT encryption:

- **End-to-end communication efficiency:** when two devices that use lightweight solutions communicate, overall energy consumption will be lowered. The end-to-end communication will therefore become effective.
- **Increased number of connections.** Any resources-controlled device may connect to the network as a lightweight solution needs fewer resources. This increases the number of network connections.

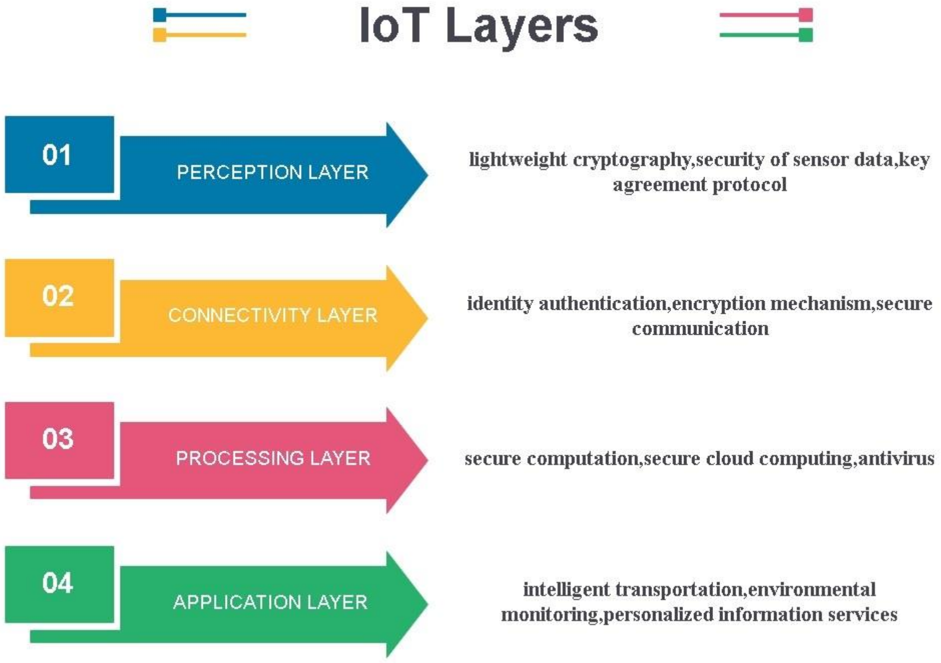

IoT Layers
01
PERCEPTION LAYER
lightweight cryptography,security of sensor data,key agreement protocol
02
CONNECTIVITY LAYER
identity authentication,encryption mechanism,secure communication
03
PROCESSING LAYER
secure computation,secure cloud computing,antivirus
04
APPLICATION LAYER
intelligent transportation,environmental monitoring,personalized information services

Figure.1 security layers in IoT

Because of their lightweight design concerns, lightweight block cipher work well with conventional systems as shown below:

- **Small key size:** the key size chosen by the lightweight cipher block should be somewhat lower than the standard cipher block size. Minimum key size is restricted to 112 bits by the National Institutes of Standards and Technology (NIST). A smaller size is more likely to lead to attacks with physical force.
- **Small block size:** the selected block sizes should be less than normal ciphers, for the small cipher. If, for example, the block size is 64 bit, the Advanced Encryption Standard (AES) may encrypt more plaintext blocks than 128 bit. In addition, the memory needs will be lower.
- **Simple round structure:** rounds for lightweight ciphers should not be cryptographical algorithms as usual. For instance, an 8-bit Substitution (S) Box can be replaced by a 4-bit S-Box to facilitate the round. This also reduces memory needs. The level of safety can be reduced by raising the total number of rounds.
- **Simple key schedule:** the key schedule creation function in lightweight designs must quickly produce the sub keys. The simpler an important schedule is, the lower power and memory the algorithm requires. A basic key schedule can lead to assaults such as a weak key, an associated key, or a selected key attack, but can be countered by utilizing a safe and frequent key generation algorithm.
- **Fewer implementation requirements:** A device must be able to encrypt or decrypt data. Rather than implementing the entire encryption, only the necessary operations should be done.

2. Related Work AND Motivation

As technology develops, every internet user is increasingly likely to undertake intelligent tasks that benefit from IoT. Security hazards arise when user information is processed online. This kind of decision will be required of a security architecture for IoT situations. Security frameworks may be developed for the Internet of Things at every architectural layer. The application layer security framework is the project's primary area of interest. IoT security frameworks vary, however currently available solutions depend on asymmetric cryptographic techniques like RSA or traditional heavyweight security mechanisms like AES in a variety of modes. The power sources for the devices may degrade as a result of these resource-intensive methods. As a result, this research study offers a security architecture that makes use of a less resource- and power-intensive security alternative. In this way, this technology also contributes to environmental protection.

after doing theoretical and analytical study, IoT security frameworks. Several security frameworks are currently in place for IoT to accomplish the same goal while pursuing the same strategy [13–15]. Each framework is founded on the same degree of expectation, as seen below:

– Dependence on software for the entire process in each framework

– A collection of protocols required to start and maintain communication between the devices.− Contribution of the IoT Safety and Data Protection Framework.

The primary issue of this section is that used by the framework as a security and authentication framework is expected to provide [16]. The C0AP framework and IoT Object Safety Framework are the frameworks examined in this article (OSCAR).

A. Framework for the Constrained Application Protocol (C0AP)

The constrained restful environment working group (CORE) of the Internet Engineering Task Force (IETF) introduced C0AP [17]. For limited devices, C0AP functions on the application layer. Using IPv6 communication through the Supervisory Power Personal Area Network is supported by IPV6 (6LoWPAN). IoT devices may communicate with C0AP by using the Transport Layer User Datagram

(UDP) and 6LoWPAN [18]. On a limited network, such as the Internet of Things (IoT) running C0AP, connections between devices or the client/server can occur [19]. In this case, one device serves as a client and another serves as a server. Because C0AP functions as an internal network, only a C0AP server may respond to a C0AP client request. In the absence of this, the C0AP may be enlarged and HTTP client requests may be handled using C0AP/HTTP mapping as C0AP functions as an HTTP subset. With the 6LoWPAN border router, this connection may be established (6LBR). The C0AP protocol, which functions at the application layer, the UDP protocol, which operates at the transport layer, and, finally, 6LoWPAN, which operates at the network layer, form the basis of the C0AP [20]. As it promotes constrained network services like IoT, 6LoWPAN is used at the network layer. Quick transfer using UDP The transportation layer uses the less reliable countermeasure control mechanism (TCP). This is as a result of the C0AP message layer's dependability mechanism. In the application layer, C0AP operates in a distinct sub-layer. The C0AP network's Put, Post, and Delete methods [21–23] are used by the request/response layer to get access to its resources. The quantity of inquiries and the mapping between their semantically correct replies are also handled by this layer.

Additional security features are implemented using datagram transport layer security (DTLS) through UDP as opposed to TCP. DTLS is intended to offer complete security. When used in conjunction with UDP, it may be used in a variety of limited applications, including Voice over IPs (VoIP), for real-time communication. Other DTLS security features in C0AP include key sharing, anonymity, and integrity [24]. In his research, the authentication and security features of C0AP are highlighted. NoSec, PreShared Key, Certificates, and C0AP all include security features.

A. A. IoT Object Security Framework (OSCAR)

OSCAR requires limited server services for a large number of customers/clients [25]. As in the traditional consumer-producer paradigm. To control consumer access to resources, OSCAR requires permission servers. For authentication, OSCAR use the basic elliptical curve concept of digital signatures. AES in CCM mode is used to develop a cryptographic security solution that protects confidentiality. Numerous more security add-ons for IoT are often created in the literature. With the increasing number of IoT surveillance sensors required, large-scale sensor-based designs become important for system operation. Routers or switches are represented to allow real-time applications with minimal latency [26]. Moreover, interconnectivity is required across all IoT-connected components requiring networking integration [27]. This openness of IoT sensors also contains various threats and flaws, which the authors identify [28]. [29] examined e-sectoral IoT security solutions. The approaches described above are platform-independent and contribute to energy savings. IoT protection is a critical problem with energy and energy in smart healthcare applications. The most common thing to do with a low-security framework in such apps is to create passwords. Some password improvement technology in IoT applications protects users' privacy in [30]. IoT devices help to assure the validity of devices and the data generated by these devices. In [30], User privacy in IoT applications is protected by a unique password reinforcement technique. IoT devices are an important component of IoT security because they ensure the authenticity of devices and the data acquired from these devices. Several authentication approaches are available in the literature to check data obtained from devices to confirm the application's authenticity. The data may be carefully collected to increase a third party's trust in the IoT system. [31] proposes a policy-based method for sensing sensitive data. The suggested system, Real Alert, demonstrates the trustworthiness of both the technologies and the data gathered. [32] announces the author of the IoT Validation Protocol (AAoT). This configuration The downside of this strategy is that dynamic vulnerabilities cannot be corrected. Authenticated key exchange [33] is still vulnerable to random secret value leaks, but it protects against lateral canal attacks and may be used in key certificate management. A SCADA model was presented in [34] for threat identification. Modern assaults employ a variety of clever ways, and one of their intelligent security strategies is provided to deal with attack detection data [35]. The authors of [36] investigated the history, concerns, gaps in IoT, and difficulties of these assaults. In [37], Zigbee technology authenticates devices among devices. This is crucial because heterogeneous IoT devices require a framework for inter-device security.

1.1. Organization

The rest of the document is formatted similarly. The current IoT security framework is demonstrated in the second half of the literature study by its security properties and operational methodologies. This section will go through two existing frameworks: The first is the C0AP, and the second is the object architecture for IoT security (OSCAR). The third section introduces a new LWES safety architecture for IoT, which is divided into three phases: registration, authentication, and data security. This section outlines the process of completing a single registration. Authentication and data security must be maintained at all times while transferring data. Following that, the current C0AP and OSCAR safety frameworks will be compared with the proposed LWES safety framework using the COOJA simulator, and various debates and choices will be made depending on the results. Memory needs, overhead energy, overhead compute, and communication rate are the performance criteria to compare. Lastly, the final part concludes with the proposed LWES's state of the art and operational efficacy.

3.Research Method

a) A critical reading of the security techniques used in the Internet of Things.
b) A contradictory analysis of the security techniques used in the Internet of Things.
c) Take advantage of the above and access the technical design (LWES) compared to the traditional techniques used like C0AP and OSCAR.
d) The general framework of the technique of three phases of protection Registration, Authentication and Data Security has been reached.
e) Application of (LWES) technology to identify the extent of its efficiency compared to the traditional techniques used like C0AP and OSCAR.

4.Proposed Security Framework for Lightweight Encryption (LWES)

While developing an IoT security framework, three primary security steps are considered: registration, authentication, and data security.

- **Registration:**is the initial stage of each gadget. When a device joins to the network, its identification is saved on the server. This is an unusual approach.

- **Authentication:** After the device is registered and has information to communicate with the server, it must first confirm its identity on the server. As a response, the server authenticates the device. After reciprocal authentication is complete, the data transmission procedure can commence.
- **Data Security:** Last, and most importantly, data must be safe whenever it is transferred between the device and the server. During this stage, a third party cannot read or change the provided data. The advice for a security framework that employs a lower-key size, fewer rounds,

and a simple yet complex round structure throughout the recording, authentication, and data security process is therefore an essential input from this work. The current frameworks, on the other hand, employ a large key size and a challenging circular structure. This improves the suggested framework. A lightweight solution in comparison to existing alternatives.

For further security services, the proposed LWES comprises secrecy, integrity, and authentication. For this aim, LWES is divided into three phases. The first step is to register the new network devices. The lightweight second phase authentication is intended to function as an authentication device for the centralised server.

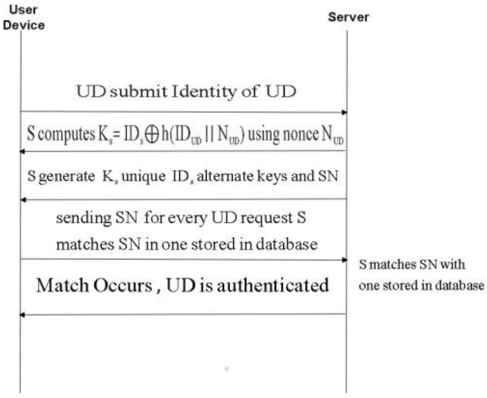

User Device
Server
UD submit Identity of UD
S computes $K_s = ID_s \oplus h(ID_{UD} || N_{UD})$ using nonce N_{UD}
S generate K_s unique ID_s alternate keys and SN
sending SN for every UD request S matches SN in one stored in database
S matches SN with one stored in database
Match Occurs , UD is authenticated

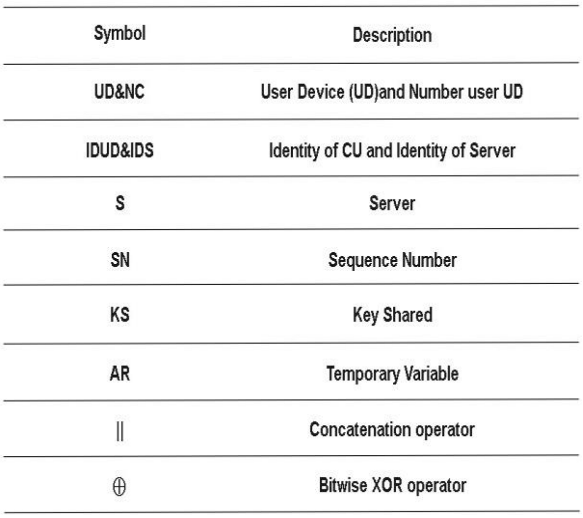
Figure.2 Registration Phase in (LWES)

every devices. Following authentication, the last stage focuses on protecting data when numerous devices interact. An example of inventory automation is used throughout LWES, and it includes elements such as a coordinating unit, items, a database, a server (S), and an internet service provider. Equipment in [37]. This is crucial because heterogeneous IoT devices require a single security framework across all devices.

4.1 Security Features

The proposed LWES provides efficient techniques for authentication, data secrecy, and validation. The entire procedure is broken down into three stages: registration, authentication, and data security. Table 1 lists the notations that are used in each algorithm.

Table 1. Notations and Description



Symbol	Description
UD&NC	User Device (UD)and Number user UD
IDUD&IDS	Identity of CU and Identity of Server
S	Server
SN	Sequence Number
KS	Key Shared
AR	Temporary Variable
||	Concatenation operator
⊕	Bitwise XOR operator

1.2. Registration Phase

As seen in Figure 2, when a new device joins a network, the key-sharing method is used to log the credentials to the server first. A new UD must present its identification to S when it joins during the registration process. S utilises its own identity, the identity of the UD, and a nonce value to calculate a secret key KS, as well as distinct IDs, alternative keys, and a unique sequence number for that specific UD. The sequence number of UD is compared to the one stored in S when it submits a connection request to S. If a match is found, the server verifies UD; if not, it uses alternative keys to establish its identity with S.

1.3. Authentication Phase

Once the device gets the credentials, as illustrated in figure 3, the mutual authentication between the client and the server is completed. The process of mutual authentication starts when UD computes a variable computational hash of its identifier, a nonce, and an already-provided secret key. Then, UD makes an authentication request with the help of a variable, its ID, and an already-held SN. If a match is discovered between the request and the SN returned by UD, S starts the authentication process. S does this by creating a temporary variable using a nonce, a secret key, and a hash function on the variable that UD acquired. UD obtains its variable from a message that it has received from S. UD also authenticates S if a match is discovered. Mutual authentication must occur before data connection may start.

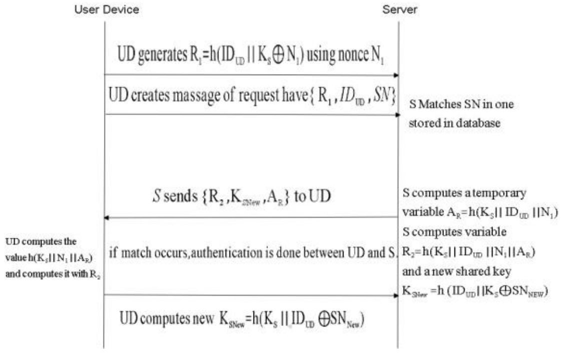

User Device
Server
UD generates R_1=h(ID_{UD} || $K_S \oplus N_1$) using nonce N_1
UD creates massage of request have{ R_1, ID_{UD}, SN}
S Matches SN in one stored in database
S sends {R_2, K_{SNew}, A_R} to UD
S computes a temporary variable A_R=h(K_S|| ID_{UD} ||N_1)
S computes variable R_2=h(K_S|| ID_{UD} ||N_1||A_R)
and a new shared key K_{SNew} =h (ID_{UD}||$K_S \oplus SN_{NEW}$)
UD computes the value h(K_S||N_1||A_R) and computes it with R_2
if match occurs,authentication is done between UD and S.
UD computes new K_{SNew}=h(K_S || ID_{UD} $\oplus SN_{New}$)

Figure 3. Authentication Phase in (LWES)

1.4. Securing Data in LWES

As illustrated in Figure 4, when the device has been approved, the data transferred to and from it is secured using a security technique. Data security in communications is achieved by computation and permutation operations on the data being transferred. The key is split into two halves first, which allows us to do calculations like finding 1s and the sum of 1s. The Ex-OR procedure is then carried out on the two separate parts. Finally, the encrypted text is produced by permuting the halves and using the crossover operation.

1.5. Security Analysis of LWES

The proposed LWES uses three separate defence strategies to thwart attacks: registration, authentication, and data transfer. Each LWES phase seeks to protect the overall framework by thwarting assaults based on its workings. In order to make LWES less vulnerable to assaults, a security analysis of the system will examine the kinds of attacks that are prevented by each step.

- **Attack Resistance during Registration**

A secret key is produced during this registration process and will be used for authentication and data security. A UD must first register with the server in order to join the network. Upon registration, S assigns a sequence number to UD. The sequence number must be shown by the UD whenever a communication starts. If an intruder attempts to connect to S, S will seek a sequence number or an alternative key, which the intruder will not have. If there is a mismatch, S won't permit an outsider to join the network. The server will be protected from a denial of service attack using the matching SN idea.

- **Attack Resistance during Authentication**

With the use of a secret key, a random number generator, and an SN created during the registration phase, UD and S may mutually authenticate one another. Many criteria must be met in order for UD to create authentication messages and for S to produce a response message in order for the full security process to take place during the authentication phase. Even if a hacker manages to get their hands on the secret key created during the registration procedure, they will be unable to authenticate themselves to the S because they lack the SN. Due to its resistance to man-in-the-middle assaults, this will help in the prevention of denial of service attacks.

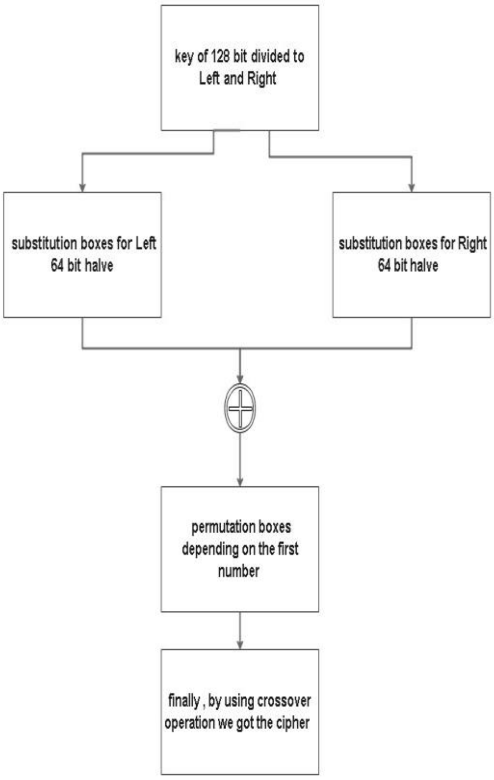

key of 128 bit divided to Left and Right
substitution boxes for Left 64 bit halve
substitution boxes for Right 64 bit halve
permutation boxes depending on the first number
finally , by using crossover operation we got the cipher

Figure.4 Data security Phase in (LWES)

- **Attack Resistance during Data Transit**

Data is safely sent during this stage. On the secret key and the plain text, several operations like EX-OR, permutation, and cross-over are carried out in order to decipher the cypher text. Because it is reversible and the predicted outcome depends on both components, the EX-OR operation is used. The straightforward yet sneaky data security method guards against compromising and replay attacks by creating a new secret key each time it is necessary to transmit information.

5. Results and Discussion

This section examines the performance of the proposed framework LWES with that of current frameworks like C0AP and OSCAR. To start, the security effectiveness of the frameworks is assessed by looking at factors like memory requirements, energy usage, computation overhead, communication speed, and denial-of-service threats. The effectiveness of the entire framework is then assessed, starting with authentication, data collection, data security, data mining, and decision making, in terms of throughput, latency, and packet delivery ratio. Because frameworks differ, specific research hypotheses are created and used for performance evaluation.

1.6.Simulation Tool and Simulation Parameters

The effectiveness of LWES, C0AP, and OSCAR is evaluated using the COOJA simulator, which Adam Dunkels developed in 2002 and runs on CONTIKI OS. COOJA offers a framework for connecting, exchanging, and sharing data across sensor motes. The motes exchange data, but each security framework is tested on data to see how well it works. The simulation settings utilised to carry out this investigation are displayed in Table 2.

Table 2. Simulation parameters

Parameter Name	Value
Radio medium	Unit Disk Graph Medium
Transmission range	50 m
Inference range	100 m
Type of Channel	Wireless
Nodes Position	Random
Sensing interval	10 s
MAC protocol	Contiki MAC
Routing protocol	RPL
Map Area	1000 * 1000 m2

1.1. Memory Requirements

The read-only memory (ROM) and random access memory (RAM) for C0AP, OSCAR, and the planned HLSF are evaluated using COOJA. Figure 5 shows the memory needs as a percentage of the total amount of memory in C0AP, OSCAP, and LWES. Figure 5 demonstrates that the ROM requirements for LWES are 3% lower than those for C0AP and 13% lower than those for OSCAR. In contrast, the RAM requirement of LWES is 2% lower than that of C0AP and 7% lower than that of

OSCAR. As a consequence, it may be concluded that LWES requires less RAM overall than OSCAR and C0AP.

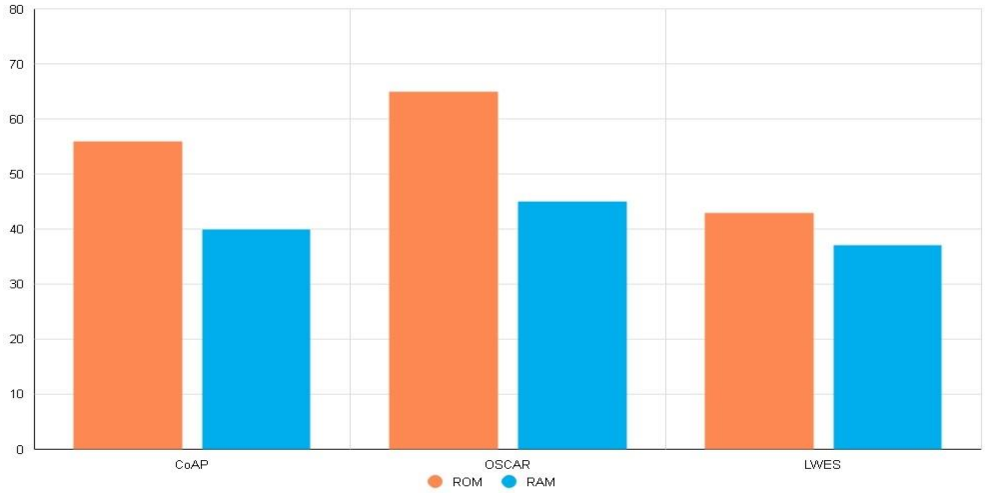


Figure.5 Percentage memory requirement for C0AP, OSCAR, and LWES

1.2. Energy Overhead

The lifetime of the sensors is directly impacted by the overhead energy of the safety framework, which also has an impact on rate of application transfer. As a result, a single application's life span gradually shortens as overhead energy grows. The overhead energy for a complex security Architecture is higher. Figure 6 displays the predicted overhead energy in mill joules for COAP, OSCAR, and LWES. Figure 6 demonstrates how packet size affects overhead energy. As packet sizes grow, so does the overhead energy. At the largest packet size, the overhead energy of the proposed LWES is 18% lower than that of C0AP and 55% lower than that of OSCAR.

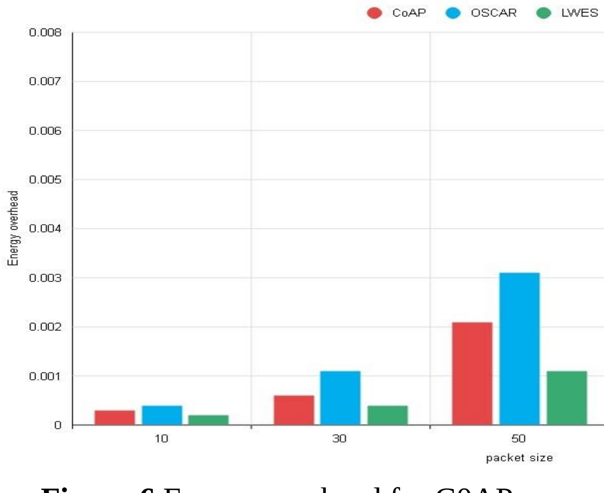


Figure.6 Energy overhead for C0AP, OSCAR, and LWES.

1.3. Computational Overhead

Computational overhead is the additional time required to employ a security framework to offer security in IoT applications. Figure 7 displays the milliseconds of the overhead computer assessment for C0AP, OSCAR, and LWES.

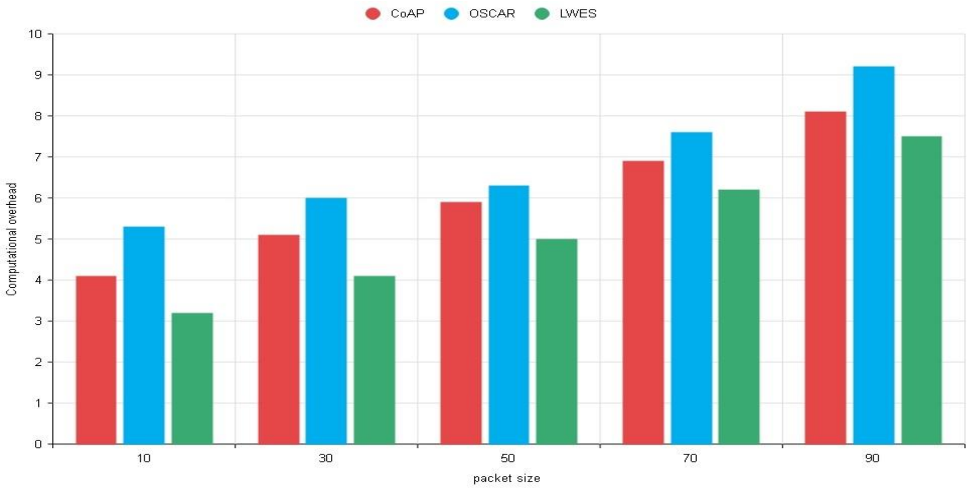


Figure.7 Computational overhead for C0AP, 0SCAR, and LWES.

Figure 7 shows how the overhead changes as the size of the packet. The amount of computer overhead continues rising as packet sizes grow. The computational overhead for LWES is 8% lower than that of C0AP and 24% lower than that of 0SCAR, even with the highest packet size.

1.4. Communication Rate

The quantity of packets transferred in a given period of time is the pace of communication. The type of safety framework used in an application circumstance affects the rate of communication. A secure application often has a lower communication rate than a non-secure application. Figure 8 displays the communication ratios for C0AP, OSCAR, and LWES using a 64-byte packet size.

More than 2% of packets are transferred every unit of time, which is the communication rate. Compared to C0AP, higher in LWES, and more than 10% higher in OSCAR from Figure 8.

Figure 8 can be concluded. In summary, the overall result is:

- With fewer rounds required, LWES has lower memory needs than C0AP and OSCAR.
- LWES does not work on an asymmetric algorithm, therefore has less energy overhead. LWES has a less complex structure.
- LWES requires less time to encrypt data, as each round produces the round keys of optimal size, which reduces its overall computational volume.

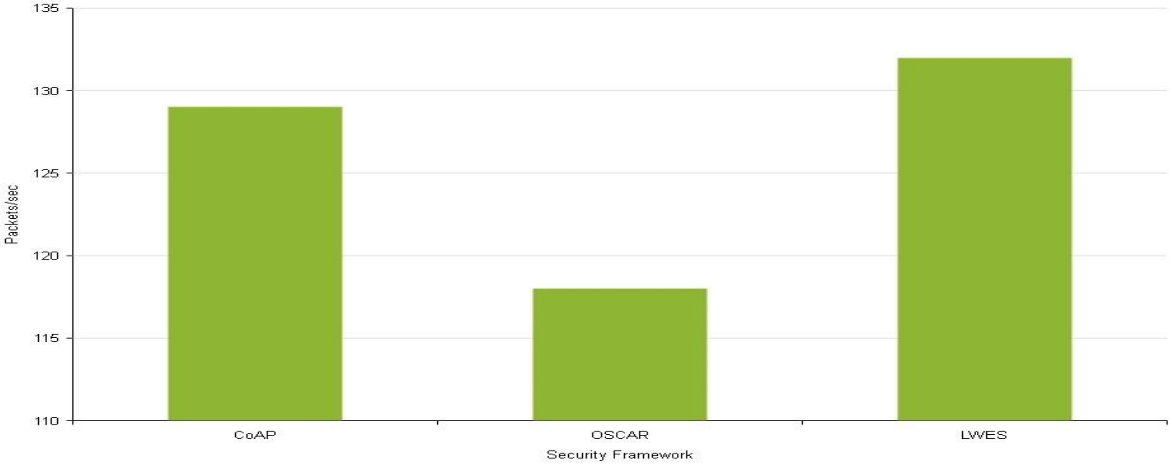


Figure.8 Communication Rate for C0AP, OSCAR, and LWES.

• Every round produces a unique set of keys, even if the number of rounds is less and the structure is less sophisticated but difficult. This reduces the vulnerability to denial of service attack, as the intruder passes the often changing key. Moreover.

2.Conclusions

Things will become intelligent around us, completing specified tasks in a self-administering manner, resulting in a new type of communication between humans and things, as well as between entities themselves. One of the needs of next-generation IoT is to provide security and network functionality while using less energy. As a result, the Lightweight Encryption Security Framework (LWES) has been proposed, which is divided into three phases: registration, authentication, and data security. LWES is a low-cost security solution that uses a smaller key and changes the pattern of the key regularly. Using COOJA simulator The Researcher compared our suggested technique to C0AP and OSCAR, two well-known frameworks. In comparison to C0AP and OSCAR, the memory required for LWES is 3% and 13% lower, respectively, according to the simulation results. In terms of computational and energy overhead, LWES surpasses the C0AP and OSCAR. In comparison to C0AP and OSCAR, the proposed technique has an 8 percent and a 24 percent lower computational overhead. LWES uses 18 percent less energy than C0AP and 55 percent less energy than OSCAR. Furthermore, the throughput of LWES is more than 2% higher than that of C0AP and 10% higher than that of OSCAR.

References

1. Verikoukis, C.; Minerva, R.; Guizani, M.; Datta, S.K.; Chen, Y.; Muller, H.A. Internet of Things: Part 2.IEEE Commun. Mag. 2017, 55, 114–115.
2. Silva, J.S.; Zhang, P.; Pering, T.; Boavida, F.; Hara, T.; Liebau, N.C. People-Centric Internet of Things.IEEE Commun. Mag. 2017, 55, 18–19.
3. Hasan, M.; Islam, M.M.; Islam, I.; Hashem, M.M.A. Attack and Anomaly Detection in IoT Sensors in IoT Sites Using Machine Learning Approaches. Internet Things 2019, 7, 100059.
4. Yang, Y.; Wu, L.; Li, G.Y.L.; Zhao, H. A survey on security and privacy issues in internet-of-things.IEEE Internet Things J. 2017, 4, 1250–1258.
5. Ling, Z.; Luo, J.; Xu, Y.; Gao, C.; Wu, K.; Fu, X. Security vulnerabilities of the internet of things: A case study of the smart plug system. IEEE Internet Things J. 2017, 4, 1899–1909.
6. Cheng, C.; Lu, R.; Petzoldt, A.; Takagi, T. Securing the Internet of Things in a quantum world. IEEE Commun. Mag. 2017, 55, 116–120.
7. Allho , F.; Henschke, A. The Internet of Things: Foundational ethical issues. Internet Things 2018, 1, 55–66.
8. Kawamoto, Y.; Nishiyama, H.; Kato, N.; Shimizu, Y.; Takahara, A.; Jiang, T. E ectively collecting data for the location-based authentication in the Internet of Things. IEEE Syst. J. 2017, 11, 1403–1411.
9. Garcia-de-Prado, A.; Ortiz, G.; Boubeta-Puig, J. COLLECT: Collaborative Context-aware service-oriented architecture for intelligent decision-making in the Internet of Things. Expert Syst. Appl. 2017, 85, 231–248.
10. Fussler, C.; James, P. Eco-Innovation: A Break thorough Discipline for Innovation and Sustainability; Pitman: London, UK, 1996.
11. Correia, E.; Carvalho, H.; Azevedo, S.G.; Govindan, K. Maturity models in supply chain sustainability: A systematic literature review. Sustainability 2017, 9, 64.

12. Li, W.; Xu, J.; Zheng, M. Green governance: New perspective from open innovation. Sustainability 2018, 10, 3845.

13. Wang, J.; Gao, Y.; Zhou, C.; Sherratt, R.S.; Wang, L. Optimal Coverage Multi-Path Scheduling Scheme with Multiple Mobile Sinks for WSNs. Comput. Mater. Contin. 2020, 62, 695–711.

14. Wang, J.; Gao, Y.; Yin, X.; Li, F.; Kim, H. An Enhanced PEGASIS Algorithm with Mobile Sink Support for Wireless Sensor Networks. Topol. Control Emerg. Mobile Netw. 2018.

15. Min, Z.; Yang, G.; Wang, J.; Kim, G. A Privacy-preserving BGN-type Parallel Homomorphic Encryption Algorithm Based on LWE. J. Internet Technol. 2019, 20, 2189–2200.

16. Choo, K.R.; Gritzalis, S.; Park, J.H. Cryptographic Solutions for Industrial Internet-of-Things: Research Challenges and Opportunities. IEEE Trans. Ind. Inform. 2018, 14, 3567–3569.

17. Eclipse Organization. Mqtt and Coap, IoT Protocols. Available online: http://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php (accessed on 5 February 2014).

18. Shelby, Z.; Hartke, K.; Bormann, C. The Constrained Application Protocol (CoAP); Standards Track RFC 7252; Center for Computing Technologies (TZI), University of Bremen: Bremen, Germany, June 2014.

19. Miši´c, J.; Ali, M.Z.; Miši´c, V.B. Architecture for IoT Domain with CoAP Observe Feature. IEEE Internet Things J. 2018, 5, 1196–1205.

20. Miši´c, J.; Miši´c, V.B. Proxy cache maintenance using multicasting in CoAP IoT domains. IEEE Internet Things J. 2018, 5, 1967–1976.

21. Correia, N.; Sacramento, D.; Schütz, G. Dynamic aggregation and scheduling in CoAP/observe-based wireless sensor networks. IEEE Internet Things J. 2016, 3, 923–936.

22. Betzler, A.; Gomez, C.; Demirkol, I.; Paradells, J. CoAP congestion control for the internet of things. IEEE Commun. Mag. 2016, 54, 154–160.

23. Son, S.; Kim, N.; Lee, B.; Cho, C.; Chong, J.Atime synchronization technique for coap-based home automation systems. IEEE Trans. Consum. Electron. 2016, 62, 10–16.

24. Park, C.; Park, W. A Group-Oriented DTLS Handshake for Secure IoT Applications. IEEE Trans. Autom. Sci. Eng. 2018, 99, 1–10.

25. Vucinic, M.; Tourancheau, B.; Rousseau, F.; Duda, A.; Damon, L. OSCAR: Object Security Architecture for the Internet of Things. In Proceedings of the 2014 IEEE 15th International Symposium, Sydney, Australia, 19 June 2014; pp. 3–16.

26. Aly, M.; Khomh, F.; Haoues, M.; Quintero, A.; Yacout, S. Enforcing Security in Internet of Things Frameworks: A Systematic Literature Review. Internet Things 2019, 6, 100050.

27. Younis, M. Internet of everything and everybody: Architecture and service virtualization. Comput. Commun. 2018, 131, 66–72.

28. Alaba, F.A.; Othman, M.; Hashem, I.A.T.; Alotaibi, F. Internet of Things security: A survey. J. Netw. Comput. Appl. 2017, 88, 10–28.

29. Hellaoui, H.; Koudil, M.; Bouabdallah, A. Energy-efficient mechanisms in the security of the internet of things: A survey. Comput. Netw. 2017, 127, 173–189.

30. He, D.; Ye, R.; Chan, S.; Guizani, M.; Xu, Y. Privacy in the Internet of Things for Smart Healthcare. IEEE Commun. Mag. 2018, 56, 38–44.

31. Li, W.; Song, H.; Zeng, F. Policy-based secure and trustworthy sensing for the internet of things in smart cities. IEEE Internet Things J. 2018, 5, 716–723.

32. Feng,W.; Qin, Y.; Zhao, S.; Feng, D. AAoT: Lightweight attestation and authentication of low resource things in IoT and CPS. Comput. Netw. 2018, 134, 167–182.

33. Ruan, O.; Zhang, Y.; Zhang, M.; Zhou, J.; Harn, L. After-the-fact leakage-resilient identity based authenticated key exchange. IEEE Syst. J. 2018, 12, 2017–2026.

34. Huda, S.; Yearwood, J.; Hassan, M.M.; Almogren, A. Securing the operations in SCADA-IoT platform-based industrial control system using ensemble of deep belief networks. Appl. Soft Comput. 2018, 71, 66–77.

35. Miloslavskaya, N.; Tolstoy, A. Internet of Things: Information security challenges and solutions.mCluster Comput. 2019, 22, 103–119.

36. Adat, V.; Gupta, B.B. Security in Internet of Things: Issues, challenges, taxonomy, and architecture. Telecommun. Syst. 2018, 67, 423–441.

37. Alshahrani, M.; Traore, I.; Woungang, I. Anonymous mutual IoT inter-device authentication and keymagreement scheme based on the ZigBee technique. Internet Things 2019, 7, 100061.