Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No.6 (2023)

Privacy and Security in IoT Security Middleware for Critical Applications

Ketan Gupta, Atharva Patil, Kirtan Bhat, Shivam Kumar Sakolia, Yashwant Dongre

Computer Science Department, VIIT, Pune, India

***Abstract*:**IoT middleware is an extra layer that sits between cloud apps and IoT devices, cutting down on cloud processing and data handling. Middleware primarily uses IoT gateways to connect to various IoT devices in a typical IoT system model. The emergence of novel IoT applications on cloud platforms has resulted in fresh risks to data security and privacy. IoT system security is regarded as a significant development. It is found in numerous appliances, including electronic voting, healthcare, and industrial uses. Cryptography as well as Steganography are used to stop hackers from reading covert information, where steganography is found to be most effective for programs designed to keep attackers from discovering who transmitting confidential information. The purpose of this paper is to suggest a new steganography algorithm. The morse code algorithm is the newly suggested steganography algorithm. This algorithm consists of encryption, hiding and decryption. The performance of security of proposed algorithm for iot middleware is found better as compared to existing systems.

Keywords:*IOT, Security, Privacy, Middleware.*

1.INTRODUCTION

The term "Internet of Things" (IoT) refers to a setting in which billions of resources-constrained objects (referred to as "things") are connected to the Internet and engage in autonomous communication.[2]

A. Internet of Things(IOT)

Kevin Ashton is credited with coining the term "Internet of Things" (IoT) when he began a presentation titled 'Internet of Things' in 1999. Subsequently, significant advancements in the areas of security, connectivity, energy efficiency, and many other areas were made. IoT is currently regarded as a topic that is pertinent to consumers, service providers, and researchers. There are numerous architectures for the Internet of Things ecosystem, and the cloud can supply the infrastructure. IoT primarily involves "things" that produce and receive data, with processing carried out remotely.[1]

B. Need of IOT

IoT-based devices may effectively gather data using data analytics, enabling them to share that data on the cloud.There are currently more IoT-connected gadgets than people in the globe. These Internet of Things (IoT)-connected tools and gadgets include RFID inventory monitoring chips and wearables like smartwatches. Networks or cloud-based platforms connected to the Internet of Things are used by IoT-connected devices to communicate. Digital transformation is fueled by the real-time insights obtained from this IoT data collection. The worldwide environmental and humanitarian challenges, commercial operations, industrial performance, and health and safety are all expected to improve greatly as a result of the Internet of Things. As a result, a wide range of sectors are implementing internet of things solutions in order to meet future demand and enhance existing systems.[2],[7]

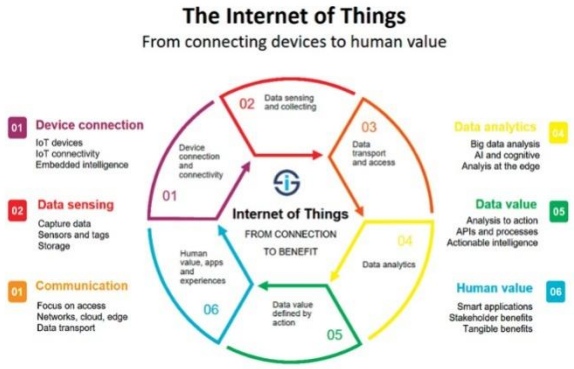
C. Advantages of IOT

a) Emergency preparedness: The internet of things-enabled smart gadgets allows for constant, fine-grained accuracy in learning about tragedies such as forest fires. Smart gadgets are capable of handling situations like

this well and can even notify the containment team in advance, allowing them to act swiftly and effectively as well.[7]

b) Interactive performance: You may effectively communicate with people in real-time with the aid of efficient data analytics. In addition, the business can monitor search terms, location, and time to learn more about the true needs of its clients. In this case, it can be claimed that the internet of things devices is used to produce dynamic interactions, demonstrating in multiple sizes at once.[7]

c) Better functionalities: When IoT devices are used, their sophisticated features provide consumers with enjoyable experiences that are nearly identical to those of mobile payments [7],[1]



D. Middleware

Software that sits in between an operating system and its installed applications is known as middleware. Middleware serves as a sort of covert translation layer that facilitates data management and communication for dispersed applications. In a dispersed network, middleware permits various forms of connectivity or communication between applications or application components. Middleware expedites the creation of applications and reduces time to market by facilitating the connection of apps that weren't intended to be connected and by offering the functionality to connect them in clever ways.[3]

E. Security

One of the main concerns when using the Internet of Things is security. Most Internet of Things (IoT) enabled devices are easily accessible by third parties and are not very secure. Therefore, it is imperative to standardize it in order to ensure that the user's privacy is not violated. A compromised object may carry out various attacks, like denial of service, or even reveal sensitive data, like a user's location, regular schedule, or even live video. The consequences of exposing such data are infinite, so platforms should take all reasonable precautions to safeguard user information in addition to offering intrusion detection systems.[4]

F. Privacy

Data privacy pertains to the handling and management of personal data, which includes establishing guidelines to guarantee that customers' personal data is gathered, shared, and utilized appropriately. Users give their consent to this practice in the service agreement, and Facebook and Google derive a significant portion of their revenue from the collection and sale of user data to advertisers. It is impossible to know for sure what information they gather, though. Data disclosures that are willingly made raise serious privacy concerns. The issue becomes even more problematic when gadgets like Google Assistant and Amazon Alexa are utilized.[4],[6]

2.Functions of Middleware

Middleware performs several functions, such as-

1. Message enrichment and routing, in addition to security oversight.

2. Encrypting these messages, validating them, and applying various business rules to them.

3. Request logging and HTTP authentication.

E. Key Management

Selecting the right encryption algorithms for your middleware scenario is one of the first stages towards encrypting data both in transit and at rest. Using a secret key, encryption algorithms are mathematical operations that convert plain text into encrypted text and vice versa. Symmetric and asymmetric encryption techniques are the two primary categories. Asymmetric encryption employs a public key for encryption and a private key for decryption, whereas symmetric encryption utilizes the same key for both operations. Faster and easier to use, symmetric encryption necessitates safe key distribution and storage. Although asymmetric encryption is more scalable and safer, it necessitates sophisticated key management and greater processing power. Your middleware requirements will determine whether to utilize a combination of both types, for example, employing symmetric encryption for data transport and asymmetric encryption for key exchange. Establishing a strong key management system for your middleware is another crucial step in encrypting data both in transit and at rest. The process of creating, storing, distributing, rotating, and rescinding encryption keys is known as key management. Key management makes ensuring that the keys are secure and cannot be lost or compromised, and that only authorized persons may view and decrypt the encrypted data. There are several methods for managing keys: a hardware security module, a distributed key network, or a centralized key server. The concepts of auditability, separation of duties, and least privilege should be adhered to by key management.[3],[2]

II. Working of Middleware

Any web application or website's user interface and data are connected through a layer of software called middleware. Middleware is used by developers, for instance, to control communication between a database and a web server. However, it can also be used for other purposes, such as guiding people to the actions they want to take. Middleware performs the following functions as a transitional program that links operating systems and communication protocols: Conceal a dispersed and fragmented network Using a diverse mix of software programs, create uniformity Give programmers a standardized interface to aid in the creation of applications and to promote interoperability and usability. Provide a range of general-purpose services that allow apps to collaborate and stop systems from working in parallel. Additionally, middleware helps in application development by hiding low-level programming details, masking application heterogeneity and the dispersion of underlying hardware and operating systems, and providing standard programming abstractions.[3]

III. Need of middleware in IOT

Having a middleware platform that acts as a link between objects and cloud applications is one method to manage such heterogeneous applications. The gateway and cloud are connected through the IoT middleware, which facilitates communication with intelligent devices. It controls the data traveling both ways. It allows computer devices that are embedded in common things to communicate with one other and exchange data via the Internet. By enabling computers to collect environmental data without human assistance, we can cut down on luxury, loss, and expense. The Middleware is in charge of database management, IoT device registration, and device identification in response to anauthentication request from the device. It also guarantees data security and privacy.

IV. Existing Security Approaches

A. Authentication

Before receiving or transferring any type of data, the device should authenticate itself using machine authentication, biometrics, and two-factor.The purpose of implementing 2FA is to improve security for user credentials and resources. Compared to authentication techniques that rely solely on single-factor authentication, two-factor authentication offers a higher level of security.[4]

B. Encryption

In order to ensure that the data generated or conveyed by the terminal are not intercepted by unauthorized access, encryption techniques like RSA, DSA, and DES should be employed in authentication via cryptographic

hash algorithms that provide digital signatures to the terminals.[4]

C. Data Security

Protected by a number of encryption techniques that thwart attempts at data theft. Additionally, Anti-Dos firewalls are implemented to stop other malevolent users' malicious activities.->Public IP addresses are shielded from Layer 4 to Layer 7 distributed denial of service (DDoS) attacks by the Anti-DDoS service, which also instantly notifies users of impending attacks. Anti-DDoS increases bandwidth usage and guarantees the uninterrupted operation of user services.[4],[5]

V. Proposed Security Algorithm for Securing IOT Data

The hiding algorithm for application layer security in Internet of Things environments is explained in this section. The morse code algorithm is the newly suggested steganography algorithm. A trained listener or observer can directly understand text information transmitted using Morse code, which is a sequence of on-off tones, lights, or clicks that doesn't require specialized equipment. It bears Samuel F. B. Morse's name, who invented the telegraph.

The algorithm is incredibly basic. In English, each character is replaced with a sequence of "dots" and "dashes," or occasionally with a single "dot" or "dash," and vice versa. Each text string is transformed into a series of dashes and dots. For this, a message that has been encoded with each character's Morse code is appended.

A. Encryption

1.When encrypting a word, we take each character (apart from the space) one at a time and compare it to the matching morse code that is kept in the data structure of our choice (dictionaries can be very helpful in this situation if you are coding in Python).

2. We first add a space to our string, which will hold the result, and store the morse code in a variable that will hold our encoded string. 3. There must be one space between each character and two consecutive spaces between each word when encoding in morse code.

4. Add another space to the variable holding the result if the character is a space. We keep doing this until we have gone through the entire string.

B. Decryption

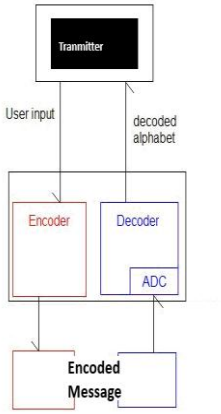
1. When it comes to decryption, we begin by appending a space to the end of the string that has to be decoded.

2. We now continue pulling characters out of the string until there is no more space.

3. Whenever a space appears, we find the English character that corresponds to the extracted character sequence (or our morse code) and add it to a variable that will hold the outcome.

4. Recall that monitoring the space is the most crucial step in this decryption procedure. Our variable containing the decoded string will receive another space as soon as we receive two consecutive spaces.

5. Since space serves as a check for extracting characters and beginning to decode them, the final space at the end of the string will assist us in identifying the final sequence of morse code characters.



C. Pseudo Code

a) Encryption

```
SET MORSE_CODE_DICT TO key value pairs
DEFINE FUNCTION encrypt(message):
   SET cipher TO ''
   FOR letter IN message:
     IF letter != ' ':
        cipher += MORSE_CODE_DICT[letter] + ' '
     ELSE
        cipher += ' '
    RETURN cipher
```

b) Decryption

```
SET MORSE_CODE_DICT TO map of key value pairs
DEFINE FUNCTION decrypt(message):
   message += ' '
   SET decipher TO ''
   SET citextTO ''
   FOR letter IN message:
     SET i TO 0
     IF (letter != ' '):
   citext += letter
     ELSE:
   i += 1
     IF i EQUALS 2:
```

```
    decipher += ‘’
  ELSE:
    Decipher += GET VALUE OF MORSE CODE FROM      MORSE_CODE_DICT
  SET citext TO ‘’
RETURN decipher
```

VI. Experimentation

Experimentation	Details
1.System used for implementation	CPU Core
2.Computer system with I3, I5, I7	I3-1115G4, I5-1135G7, I7-1185G7
3.RAM HDD	EEPROM, EPROM
4.Language	Java, Encryption and decryption pseudo code
5. Attack Dash	MQTT protocol

Table 1

A. CPU Core

A single processing unit in the CPU that is capable of carrying out instructions is called a CPU core. A CPU can do more tasks at once the more cores it has. Having more cores enables you to run multiple programs simultaneously and transition between them more easily than having a slower CPU, which usually helps you load apps more quickly.

VII. Results and discussion

A. Results on privacy

A middleware server is first started as part of the test suite procedure. Three requests are then sent to "warm-up" the

middleware: a registration client request, a notification insert request, and a client deletion request. Every device is turned on simultaneously and operates in parallel. Device simulators stop running their applications after the allotted time has passed, stop the middleware, and gather the results.We receive a single set of results from a single test suite, which consists of entries with the following information:Request send and associated response received timestamps are on the client (device) side; request processing start and finish timestamps are

on the server side; and request processed with or without errors is on the server side.data sets, including processing time (measured as the difference between the client-side timestamps of the request and response events). The following setups were examined:Taskset -c 0-19 denotes 10 physical cores of CPU 0 and 10 physical cores of CPU 1, Taskset -c 0-39 denotes all of the system's logical cores Taskset -c 0-1 denotes two physical cores of CPU 0.The numbers of requests that were successfully processed are shown in Figures 1. by settings that are containerized and bare metal, respectively. Four consecutive lines, as shown below, stand for four distinct middleware-side configuration.

Figure 3: Total number of requests fulfilled successfully

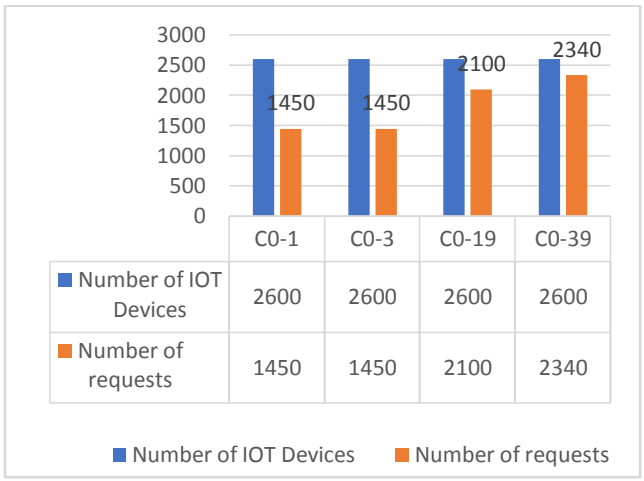


Figure 3. Container for the number of requests that were processed successfully. CPU core counts range from C0–1–2 to C0–3–4 to C0–19–20 to C0–39–40 (including Hyper Threading).The client-side request-response times for both containerized and bare-metal scenarios are shown in Figures 2. we concentrated our attention on the configurations with respective optimal values for C0–3, C0–19, and C0–39 based on the numbers of successfully processed requests.

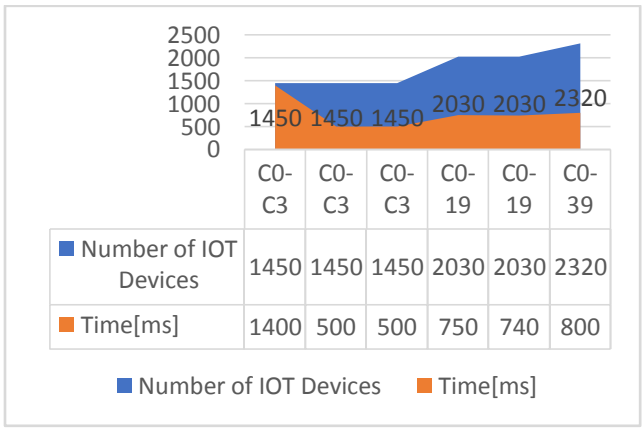


Figure 4: Shows the client-side request-response processing time

B. Containerization's effect on scalability

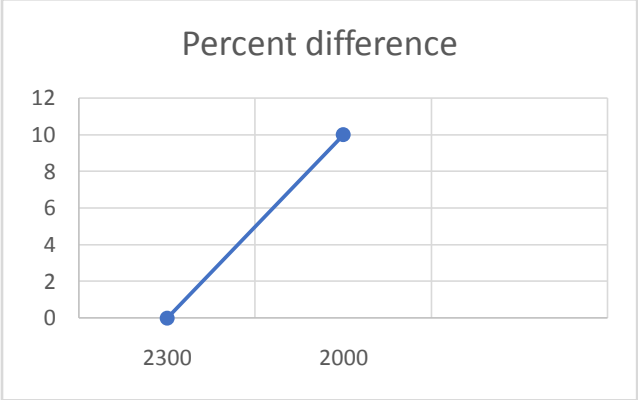
Based on performances, it is anticipated that launching a container with the same core affinity as the same setup on bare metal will produce results that are quite close. The normalized difference between the number of successfully completed requests N on bare metal and the same metric on the container was displayed. The value is computed as follows:

$NprocessedPh\ ysical - NprocessedDocker/NprocessedPh\ ysical$

These findings, along with the findings from earlier charts in Figures 4, as well as Figures 3, make it abundantly

evident that containerization has no discernible effect on the level of scalability attained with varying server core counts.

Figure 5: Bare metal and container requests that were successfully processed differently.



C. Containerization's Effect on Performance

We looked into how containerization affected performance. We only utilized the two strongest variants for this comparison. The first one used cores 0–19, whereas the second used cores 0–39 total. Figures 4 and 5 show variations in processing times as measured from the client side and the server side, respectively. We see that the bare-metal and containerized environments do not significantly differ from one another.

$$difference = tphysical - tdocker / tphysical$$

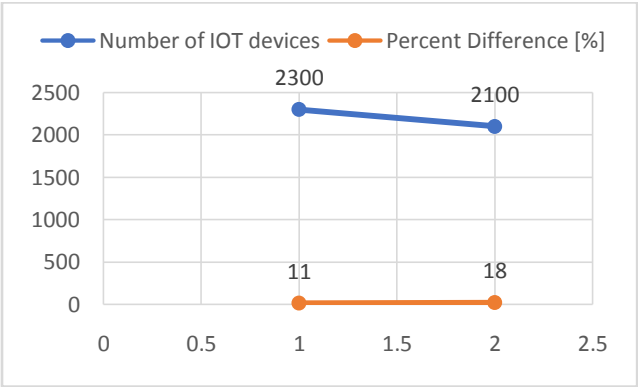


Figure 6: Variation in processing times on the server side

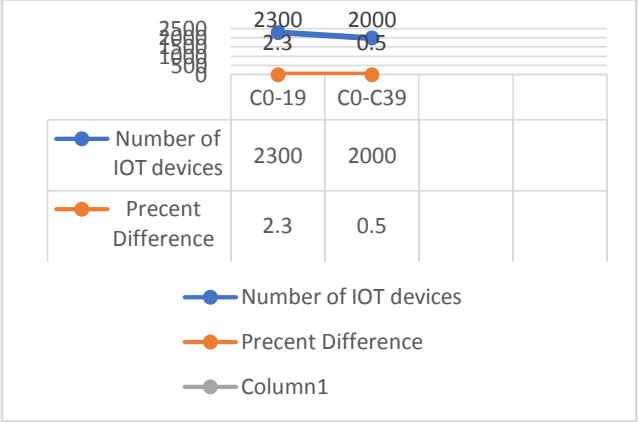


Figure 7: Variation in client-side processing time

D. Results on security

Security Devices and servers have a two-way communication established in an IOT system. A gadget receives control orders from the server and transmits data to it. The level of authentication involves verifying the legitimacy of the device and the server. Lightweight authentication and encryption technologies are highly sought after in the Internet of Things (IoT) in order to offer a secure environment for communication and access control. An encryption is required in order to transfer data between two or more linked devices. The cloud of things is difficult to manage because of its low complexity, limited development, and mobility. There exist multiple encryption technologies to guarantee. confidentiality. Data transit between two or more linked devices requires an encryption. The low complexity, constrained development, and mobility of the cloud of things make it challenging to manage. To provide confidentiality, several encryption technologies are available. A hacker must be inside the device's coverage area in order to launch an attack and obtain credentials. There won't be any evidence of the SCA attack in the future. The potential for these kinds of attacks to compromise bank cards, mobile devices, or even medical equipment may increase when SCA is utilized.A sensor network's secure routing is a crucial component. For data to be delivered and executed between a server and a device, each sensor node is dependent on a routing protocol. A weak routing protocol can seriously compromise a user's privacy and leave behind evidence that a hacker could exploit to access the network. A hacker may release his own commands to control a sensor or servers if he is able to capture packets from the network and see what orders are being delivered.

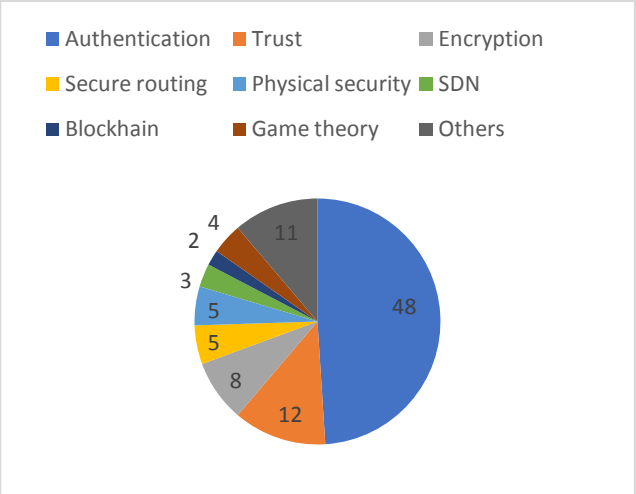


Figure 8. IOT security publications published in Elsevier, IEEE<Hindawi, and Springer between 2018 and 2023.

VIII. Conclusion

In this paper, a novel steganography algorithm for secure data at the Internet of Things application layer is proposed. By contrasting it with earlier studies, the new secure algorithm for steganography based on Morse Code. We have developed a system that uses morse code, wherein the sender sends a message that is encrypted by our system and sent to the recipient in a much more secure format that can be readily decrypted to reveal the original message.

References

[1] Jihad DAZINE, Abderrahim MAIZATE, Larbi HASSOUNI “Internet of things security” ,2018 IEEE.

[2] Mauro A. A. da Cruz , Joel J. P. C. Rodrigues , Fellow, IEEE, Pascal Lorenz , Senior Member, IEEE, Valery V. Korotaev, and Victor Hugo C. de Albuquerque , Senior Member, IEEE, IEEE “INTERNET OF THINGS” JOURNAL, VOL. 8, NO. 10, MAY 15, 2021.

[3] Dongre, Y. ., & Patil, P. . (2023). Genetic Algorithm based Optimal Service Selection of Composition in Middleware using QoS Correlation. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(2), 20–29. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2591

[4] Rana Wafeek Mansy, Mohamed Helmy Megahed , Marwa Salah, Mahmoud M. El-khouly," A Proposed Security Algorithm for Securing IoT Data", Vol. 19, No. 6, June 2021.

[5] HittuGarg , Mayank Dave ," Securing IoT Devices and SecurelyConnecting the Dots Using REST API and Middleware", 2019 IEEE6] Omar H. Alhazm , Khalid S. Aloufi ," Fog-Based Internet of Things: A Security Scheme",2017 IEEE.

[6] Shivangi Vashi, Jyotsnamayee Ram, Janit Modi "A Vision, Architectural Elements, and Security Issues", (I-SMAC 2017).

[7] Mauro A. A. da Cruz, Joel José P. C. Rodrigues , Senior Member, IEEE, Jalal Al-Muhtadi, Valery V. Korotaev, and Victor Hugo C. de Albuquerque, Member, IEEE "A Reference Model for Internet of Things Middleware", VOL. 5, NO. 2, APRIL 2018.

[8] Misra, P.; Simmhan, Y.L.; Warrior, J. Towards a Practical Architecture for the Next Generation Internet of Things. *CoRR* 2015. [Google Scholar].

[9] Mashal, I.; Alsaryrah, O.; Chung, T.Y.; Yang, C.Z.; Kuo, W.H.; Agrawal, D.P. Choices for interaction with things on internet and underlying issues. *Ad Hoc Netw.* 2015, *28*, 68–90. [Google Scholar] [CrossRef].

[10] Saied, Y.B.; Olivereau, A.; Zeghlache, D.; Laurent, M. Lightweight collaborative key establishment scheme for the Internet of Things. *Comput. Netw.* 2014, *64*, 273–295. [Google Scholar] [CrossRef].

[11] Razzaque, M.A.; Milojevic-Jevric, M.; Palade, A.; Clarke, S. Middleware for Internet of Things: A Survey. *IEEE Internet Things J.* 2016, *3*, 70–95. [Google Scholar] [CrossRef].

[12] Fersi, G. Middleware for Internet of Things: A Study. In Proceedings of the International Conference on Distributed Computing in Sensor Systems 2015, Fortaleza, Brazil, 10–12 June 2015; pp. 230–235. [Google Scholar].

[13] Xu Xiaohui, "Study on Security Problems and Key Technologies of The Internet of Things", International Conference on Computational and Information Sciences, 2013.

[14] Mayuri A. Bhabad and Sudhir T. Bagade , "Internet of Things: Architecture, Security Issues and Counter measures", International Journal of Computer Applications,2015.

[15] Xue Yang, Zhihua Li, Zhenmin Geng and HaitaoZhang ,"Internet of Things" International Workshop, IOT 2012, Changsha, China, August , 2012.

[16] Qi Jing, Athanasios V. Vasilakos, Jiafu Wan, Jingwei Lu and Dechao Qiu, "Security of the Internet of Things: perspectives and challenges", Wireless Netw ,2014.

[17] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey,"Comput. Netw., vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[18] B. B Prahlada Rao ; Paval Saluia ; Neetu Sharma ; Ankit Mittal ; Shivay Veer Sharma, Cloud computing for Internet of Things & sensing based applications, in Proceedings of the 6 th International Conference on Sensing Technology 2012.

[19] Gubbi, Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswamia. "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," Future Generation Computer Systems 29, no. 7, 2013, pp:1645-1660.

[20] O. Said and M. Masud. "Towards internet of things: survey and future vision," International Journal of Computer Networks 5, no. 1, 2013, pp. 1–17.

[21] R. Khan, S. U. Khan, R. Zaheer, and S. Khan. "Future Internet: the internet of things architecture, possible applications and key challenges," in Proceedings of the 10th International Conference on Frontiers of Information Technology (FIT '12), (2012), pp. 257–260.

[22] https://www.techtarget.com/searchapparchitecture/tip/How-to-address-security-risks-posed-by-middleware-tools.
[23] Omar Alhazmi, "A Survivable Internet of Things Scheme,". Journal of Advanced Research in Computing and Application, Volume 13, No. 1, December 2018.
[24] Gan, W.; Lin, C.W.; Fournier-Viger, P.; Chao, H.C.; Tseng, V.; Yu, P. A survey of utility-oriented patternmining. IEEE Trans. Knowl. Data Eng. 2019. [CrossRef]
[25] Pan, J.S.; Lee, C.Y.; Sghaier, A.; Zeghid, M.; Xie, J. Novel systolization of subquadratic space complexity multipliers based on toeplitz matrix–vector product approach. IEEE Trans. Very Large Scale Integr. Syst. 2019, 1614–1622.
[26] Chen, C.M.; Xiang, B.; Liu, Y.; Wang, K.H. A secure authentication protocol for internet of vehicles. IEEE Access 2019, 7, 12047–12057.
[27] Wu, T.Y.W.; Chen, C.M.; Wang, K.H.; Meng, C.; Wang, E.K. A provably secure certificateless public key encryption with keyword search. J. Chin. Inst. Eng. 2019, 42, 20–28. [CrossRef].
[28] Chen, C.M.C.; Wang, K.H.; Yeh, K.H.; Xiang, B.; Wu, T.Y. Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications. J. Ambient Intell. Hum. Comput. 2019, 10, 3133–3142. [CrossRef].
[29] Xiong, H.; Zhao, Y.; Peng, L.; Zhang, H.; Yeh, K.H. Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing. Fut. Gener. Comput. Syst. 2019, 97, 453–461. [CrossRef].
[30] Lin, J.C.W.; Zhang, Y.; Zhang, B.; Fournier-Viger, P.; Djenouri, Y. Hiding sensitive itemsets with multiple objective optimization. Soft Comput. 2019. [CrossRef].
[31] https://www.ibm.com/topics/middleware.
[32] https://www.talend.com/resources/what-is-middleware/.
[33] https://www.researchgate.net/publication/pdf/359252324_Middlewar_101_What_to_know_now_and_for_the_future.
[34] https://dl.acm.org/doi/introductiontomiddleware/pdf/10.1145/3526211.
[35] https://www.linkedin.com/advice/0/what-some-best-practicencrypting-data-transit-rest
[36] A Reference Model for Service Oriented Middleware January 2008 Authors: Marco Autili, Mauro Caporuscio, Valérie Issarny.
[37] https://www.howtogeek.com/194756/cpu-basics-multiple-cpus-cores-and-hyper-threading-explained/.
[38] https://link.springer.com/The Internet of Things: Definitions, Key Concepts, and Reference Architectureschapter/10.1007/978-3-030-41110-7_1.
[39] https://Different Applications and Technologies of Internet of Things (IoT)/arxiv.org/ftp/arxiv/papers/2110/2110.10452.pdf.
[40] https://www.academia.edu/21991714/Internet_of_Things_IoT_An_Overview.
[41] https://www.researchgate.net/publication/339069624_DATA_BREACHES_IN_IOT_A_STUDY_AND_SOLUTION_BY_PUF_APPROACH.