WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

EMBEDDING TRUST: INTEGRATING A LIGHTWEIGHT SECURITY FRAMEWORK FOR PERVASIVE IoT

[1]SAKSHI GANGWAR, [2]Er. HARSHIT GUPTA, [3]Dr. RUCHIN JAIN

[1]Student, Department of Computer Science and Engineering, Rajshree Institute of Management and Technology, Bareilly, U.P.
[2]Assistant Professor, Department of Computer Science and Engineering, Rajshree Institute of Management and Technology, Bareilly, U.P.
[3]Head, Department of Computer Science and Engineering, Rajshree Institute of Management and Technology, Bareilly, U.P.

ABSTRACT:

Internet of things consists of things having unique identification of each device and things are connected to the Internet. Traditional devices which are not associated with internet are now having capabilities to get connected with Internet called smart devices. The backbone to IoT network is internet. IoT devices which are connected to Internet called as resource constrained devices because of having low computing power, limited battery power, limited memory etc. Hence when this constrained devices having sensor transmits data over networks are prone to different types of attacks. To avoid attack on IoT network and maintain confidentiality of data over network, it requires cryptographic solution but due to resource constrained nature of devices in IoT comparing with traditional encryption method Lightweight cryptography is more effective. Hence to make cloud based IoT secure transmission we present Fernet Lightweight symmetric encryption based on AES-128 bit in a CBC mode algorithm and HMAC authentication using SHA256 Hash Algorithm. Comparing with other Lightweight Cryptographic algorithms Fernet is found to be best for securing cloud based IoT communication as it is having Key size of 256 bits and it can perform both encryption and authentication. In this research paper, we implemented Fernet Lightweight cryptographic algorithm for Encryption, Authentication and Key generation to secure transmission of sensor data generated by IoT devices.

Keywords: AES (Advanced Encryption Standard), Fernet, IoT (Internet of Things), Lightweight Cryptography, MQTT protocol

INTRODUCTION

IoT devices are referred to as “Things” in IoT. These devices have unique Identification and have different capabilities like sensing from remote location, actuating capabilities. These devices collect and process data. For the processing it can be send on cloud based application. Hence after processing it, perform some task locally or at IoT Infrastructure. IoT devices consist of I/O interface for Sensors, Internet connectivity interface, Audio/Video Interface, storage interface. These devices can sense data from sensors which can sense external environment like light, motion, humidity or temperature etc. The sensor data is then communicated to cloud for storage purpose. After processing of data, information is generated and command is given to devices with the help of actuators to perform some actions like ON/OFF the device. Relay switch connected to IoT Device can perform action of ON/OFF the device based on command received to device over Internet. IoT is not limited to connecting devises to Internet. But while designing application for a user it is required to process on raw data, convert it into meaningful information and extract it into knowledge. So while considering this for example we have data of sensor measurement ((62, 44) ;(72, 55)) which is generated after weather monitoring. Suppose we have to give meaning to this data we have to add context. Suppose we consider tuple in data which is temperature and humidity measurement for per minute. With this context we can get information of the tuple. Further processing this information in terms of Average Temperature and Humidity for last ten minutes by averaging last ten data tuples. Further to extract knowledge understand the relationship in information. Suppose we set Alert if average temperature in last ten minutes is greater than 110F. So that will be useful for any user to getting and alert
On certain conditions and application can be developed on it. Hence IoT has wide range of application like Smart home can have smart lighting, smoke /Gas Detectors; smart cities can have smart parking, smart roads etc.

IoT Functional Blocks

IoT Devices: IoT devices consists of sensors, actuators, functions for monitoring and controlling

- Communication: Various protocols like CoAP, MQTT, AMQP used for communication by IoT systems.
- Services: Services like device monitoring, control, data publishing services are required in IoT architecture.
- Management: This functional block is to govern IoT system.
- Application: It’s an interface for users to monitor and control various aspects of IoT system. Also IoT network has some capability like
  - Collection of data
  - Transmission of data
  - Processing of data and

- Utilization of data.

In first stage that is data collection, in this stage resource constrained devices with the help of sensors can sense or collect information from environment or outside world. In second stage that is data transmission stage which uses Ethernet, Wi-Fi, ZigBee to connect objects and users. In third stage that is data processing, application process the data and obtain useful information .In fourth stage utilization of this information is done. Based on this information certain decisions are made and many initiate commands for taking actions on physical environment.

Cloud for IoT

Cloud computing is a paradigm which can deliver applications and also services over internet. Cloud services can provide computing, networking and storage asperser demand and can provide these services "pay as per requirement" basis. Cloud computing is an IoT enabled technology. Big data of IoT devices can be stored on Cloud. Hence Cloud computing and IoT are two correlated technologies.

IoT Infrastructure and Cloud Storage

In IoT network sensor devices generates big amount of data due to which cloud storage is required to store that data. Analysis of data can be done on Cloud and depending on that actuators get command for doing the task. Cloud based IoT Infrastructure is needed for most of the IoT applications.

RELATED WORK

A number of researchers from all around the world have worked for cryptographic solution in IoT network regarding security issues in constrained devices. Some related research papers are discussed here for implementation of IoT data transmission using lightweight cryptography.

- In [1], the author presented lightweight cryptographic protocols. Also presents analysis and comparative study of popular contemporary cipher. Evaluate block and stream algorithm for security.
- In [2], author presents and analyzed different Lightweight crypto graphic approaches. Lightweight cryptography is the only solution for security and performance in wireless sensor networks. Symmetric and Asymmetric Lightweight cryptographic approaches are presented.
- In [3], the author analyses different LWC algorithms. In this paper author found and check different cryptographic algorithms for resource constrained devices.
- In [4], the author presented to address comparison of LWC with other current cryptographic algorithms using different IoT platforms and analyze which is useful for limited hardware applications.

METHODOLOGY

Cloud based IoT architecture

As IoT devices increases day by day and these devices generates huge amount of data volumes called big data. Hence Cloud has been proposed to solution for IoT that can manage this big data generated by smart devices.

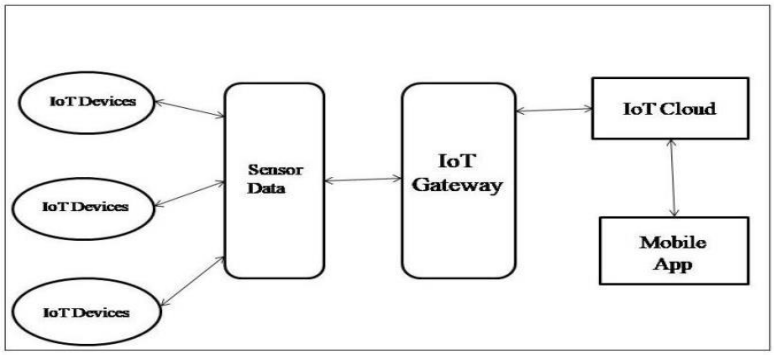

Fig.1.Cloud IoT Architecture

Fig 1 shows following components in Cloud based IoT architecture.

- IoT devices: Theses are smart devices which can sense real world data with the help of sensor and this sensor data then transmitted through network.
- IoT gateway: IoT gateway is responsible for transferring this sensors data to cloud for analysis and processing.
- IoT Cloud: Big data generated by IoT devices can be stored in IoT cloud. Data processing and analysis can be done on this big data for taking decision based on processed information.
- Mobile App: This can be handled by users to control devices in IoT architecture.

But while travelling this sensors data on wireless network to be stored on cloud it is vulnerable to different attacks.

Types of Attacks

Passive Threats: Passive Threats can use data but never affect resources. It affects confidentiality of data. Fig 2 shows how confidentiality of data is affected by Passive Threats.

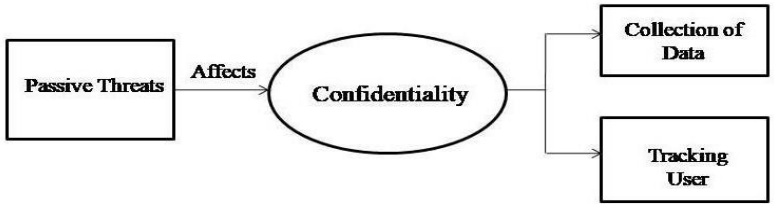

Fig.2.PassiveThreats

Active Threats: Active Threats attempts to alter data and can take control of hardware or resources.
Fig 3 shows security services are affected by active threats.

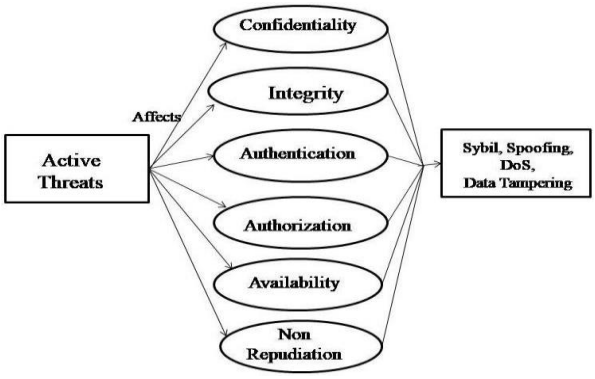

Fig.3.ActiveThreats

Attacks on different IoT components

IoT attacks can be divided into four categories

Attacks at IoT Devices

Physical Devices consists of sensors, actuators and RFIDs. Sensors can sense data from environment .RFIDs are used in wireless communication with the help of unique identifier. These devices are resource constrained due to which attacks can be easily possible on physical devices. Generally DoS, DDo Scan is possible.

Attack at IoT Network

Network service is wireless sensor network with RFID in IoT network. Hence both parts are vulnerable. Possible attacks on RFID include Sybil, synchronization attack and also replay attacks.

Attack at IoT Cloud Service

Cloud computing in IoT architecture facilities to store and obtain information from anytime anywhere. As the system is distributed it is more vulnerable to various attacks like Malicious attack can be done for unauthorized access e.g. SQL injection and cross site scripting. Security attacks or threats can be possible due to inadequate integrity controls. IoT ensure security at Cloud IoT system, these types of security concerns should be minimized.

Attack at Web Application Layer

About all IoT network provides remote access to users using mobile application. The malware designer can hack this system by extracting device information .They can do potential vulnerabilities and also can create botnets. So attacks on different components of IoT architecture which may lead to loose confidentiality, integrity of sensor data. To avoid these types of vulnerabilities on cloud IoT network some robust solutions are required. Theses requires attack detection and mitigation to defend various attacks on Cloud IoT network. Also for secure transmission of sensor data lightweight cryptography can be used.

Traditional Cryptographic Algorithms

Table1: Traditional Cryptographic Algorithms

Authentication Algorithm	**1.(MD5)** **SHA-1** **SHA-256**
Encryption Algorithm	**DES(Symmetric)** **AES(Symmetric0** **RSA(Asymmetric Algorithm)**

For the purpose of protection of data, cryptographic techniques can be used. Some traditional techniques which are used for Authentication and Encryption are listed in Table I.

Lightweight Cryptography for Resource constrained IoT devices

IoT Overview

- IoT devices are of two types:
- Rich in Resource like PC, tablets, smart phones etc.
- Poor in Resources like resource constrained having sensor, RFID tags, actuators etc. Due to the use of this resource constrained devices in many applications these become more and more popular.

Security challenges and Security requirements As these resource constrained IoT devices interact directly with outside world for purpose of data collection are easily exposed to attack .Attackers can mode these devices as a target .So to make these IoT Network secure different security requirements can be fulfilled by IoT Network. These security requirements are shows in following Fig 4.

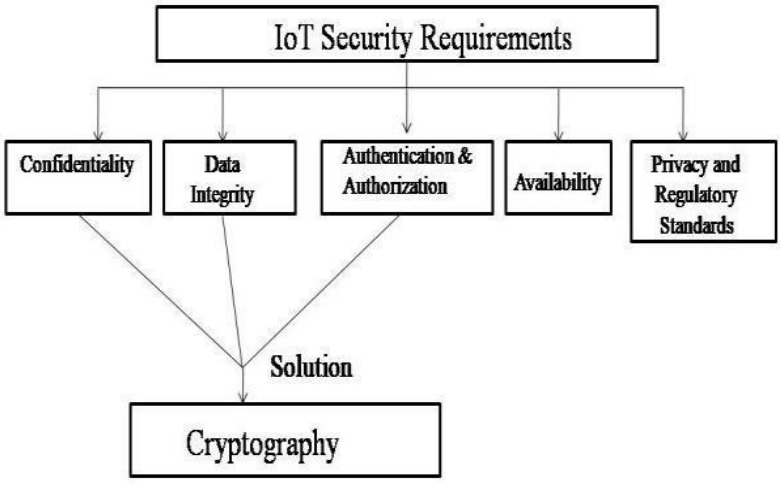

Fig.4. IoT Security Requirements and Solution

In above diagram different IoT Security requirements are explained. Amongst this cryptography can be effective measures to guarantee of data confidentiality, integrity and authentication& authorization of data at the time of transmission .So cryptography can be a solution for secure transmission of data over network as well as secure storage of data. However conventional cryptography algorithms do not suit for IoT Devices which are resource constrained as these requires high resource demand. So it requires lighter version of these conventional algorithm.

Challenges in implementing traditional cryptography

Following are some challenges in IoT devices:

1. Limited memory 2.Limited battery power 3.Low computing power 4.Real time response. Limitation in IoT devices make their performance low and not acceptable for traditional cryptography applied on IoT. But all these issues are addressed by lightweight cryptography. One more reason lightweight cryptography is no just applicable to resource constrained devices but it is also applicable to all the devices which are rich in resource which are also involved in IoT network directly or indirectly.

Characteristics offered by Lightweight cryptography

Characteristics of lightweight cryptography areas follows.

- Physical Cost
- Performance
- Security

Amongst these first two characteristics are satisfied by LWC algorithms. But the characteristics security is only fulfilled by different internal structure adoption for defending against attacks. These structures are SPN, FN, GFN, ARX, NLFSR, and Hybrid. Following Fig 5 shows structure wise classification of cryptography.

Classification of Cryptographic algorithms is in two main categories.

- Symmetric Key
- Asymmetric Key

Symmetric key cryptography uses same single key for encryption and decryption but sharing of this key safety is the major issue and which can be solved by using trusted third party. Asymmetric cryptography uses two private public key pairs. It also provides confidentiality, data integrity and authentication of data. In block cipher both encryption and decryption take place on a fixed size block (64bitsormore).But instream cipher continues processing of input elements is done bit by bit. Two properties are there in cryptography

- Confusion
- Diffusion.

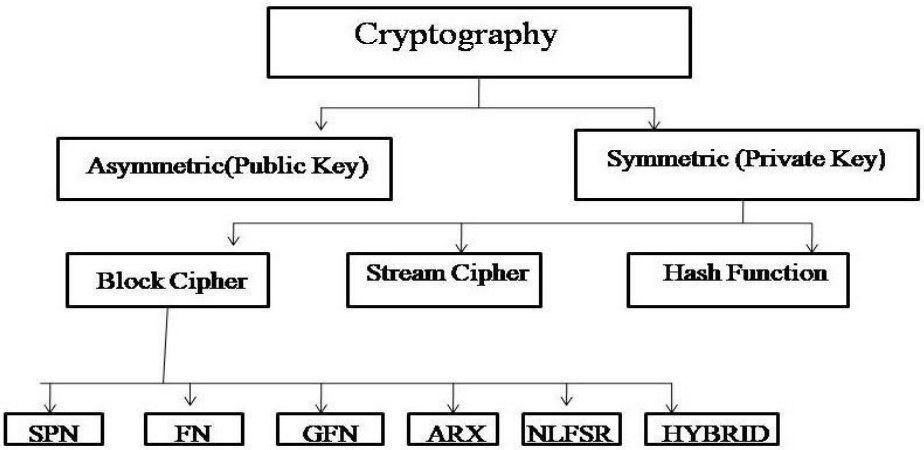

Fig.5.Classification of cryptography

It is used to strength the cipher. The confusion is used to make key and cipher text relationship more and more complex by using S-box substitution. The diffusion uses permutation to dissipate plaintext over bulk of cipher text. But stream cipher uses only confusion property and block cipher uses both confusion and diffusion. Hence reverse of encryption for extracting plain text is very difficult in block cipher and easy in stream cipher. Hence block cipher is preferred in resource constrained IoT Devices

Table2.Comparison of different Types of Cryptographic Algorithms.

Algorithm	Operation	Type	Key size
AES	Encryption	Symmetric	128 bits, 192bits, 256bits
DES	Encryption	Symmetric	56 bits
RSA	Encryption	Asymmetric	1024bits, 2048bits, 3072bits, 7680bits, 15360bits
Present	Encryption (Lightweight)	Symmetric	80 bits, 128 bits
Clefia	Encryption (Lightweight)	Symmetric	128 bits

Fernet	**Encryption (Lightweight) & Authentication**	**Symmetric**	**256 bits**

Table 2, shows comparison of different cryptographic algorithms. Amongst these in SPN i.e. Substitution Permutation Network AES is a best example having 128bit block with different keys like 128,192,256 bits. Based on AES 128 bit, Fernet is found to be a best Lightweight cryptographic algorithm because comparing with others Key Size is 256 bits and also it provides both Encryption and Authentication which can be used for securing sensors data of IoT network in terms of confidentiality and integrity.

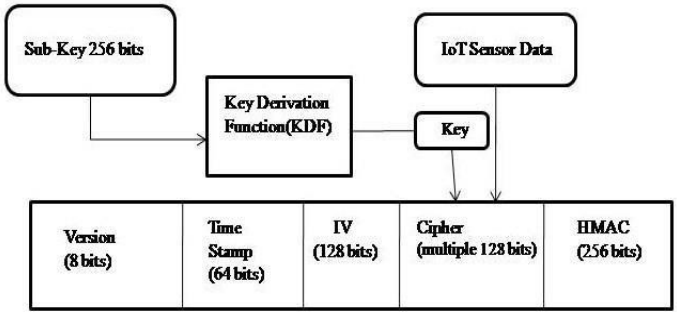

Fig.6.Fernet Symmetric Encryption Architecture

Fig6.ShowsarchitectureofFernetSymmetricEncryption. IoT sensors data generated from IoT devices is used as an input for encryption. It requires Key which is derived from KDF. By using Key encryption of IoT sensor data is done and it gets encoded to form Cipher text. The same secret key is shared with recipient for authentication and decryption purpose.

4. PROPOSED LIGHTWEIGHT SECURITY FRAMEWORK ARCHITECTURE

4.1. Overall Architecture:

Present a high-level architectural diagram of the proposed framework. Illustrate the different layers (device, network, application) and the key components of the framework.

4.2. Framework Principles: Outline the core principles guiding the design of the framework:

- Lightweightness: Minimizing resource consumption (CPU, memory, power).
- Modularity: Allowing for flexible integration and adaptation to different IoT devices and use cases.
- Scalability: Supporting a large number of diverse devices.
- Trust Embedding: Integrating trust mechanisms at different layers.
- Security by Design: Incorporating security considerations from the initial design phase.

4.3. Key Components: Detail the main components of the framework:

- Secure Device Boot and Integrity Verification: Mechanisms to ensure the device boots into a trusted state and that its software is not tampered with.
- Lightweight Cryptographic Module: A set of resource-efficient cryptographic algorithms and protocols for authentication, confidentiality, and integrity.
- Trust Management Module: Mechanisms for evaluating, propagating, and managing trust relationships between devices and entities.
- **Secure Communication Protocol:** A lightweight protocol for secure data exchange between devices and the network. **Secure Firmware Update Mechanism:** A robust and secure method for updating device firmware remotely.
- **Access Control and Authorization Module:** Mechanisms to enforce access policies and control which entities can interact with the device and its data.

5. DETAILED DESIGN AND MATHEMATICAL DERIVATIONS

This section will delve into the technical details of the key components and include mathematical derivations for critical security functions.

5.1. Secure Device Boot and Integrity Verification:

5.1.1. Trusted Platform Module (TPM) or Hardware Security Module (HSM) Integration: Discuss how lightweight hardware security elements can be used for secure key storage and cryptographic operations.

5.1.2. Chain of Trust Establishment: Explain the process of establishing a secure boot chain, starting from a hardware root of trust.

5.1.3. Integrity Measurement and Verification: Describe how hashes or digital signatures can be used to verify the integrity of loaded software components. Mathematical Derivation Example:

5.1.4 Derivation of a Lightweight Hash Function for Integrity Checking: Introduce the concept of a hash function H(M) that produces a fixed-size output h from an input message M.

- Discuss the properties of a good hash function (pre-image resistance, second pre-image resistance, collision resistance).
- Present a simplified example of a lightweight hash function, focusing on its structure and operations suitable for resource-constrained devices. This could involve:
- Defining the input block size B and output hash size L.
- Describing the iterative process of processing input blocks.
- Defining a compression function C(Hi−1,Mi) that combines the previous hash value H i−1 and the current message block Mi to produce the next hash value H i .
- Illustrating the operations within the compression function, e.g., bitwise operations (XOR, AND, NOT), rotations, and additions, specifically chosen for their efficiency on lightweight processors.
- Derivation: Define the compression function C mathematically. For example, using a simplified structure similar to a block cipher's round function:

$$H_i = C(H_i - 1, M_i)$$

Where C could involve a series of rounds. Let's consider a single round R(h,m) where h is a portion of the current hash state and mmm is a portion of the current message block.

$$R(h,m) = ((h \oplus m) <<< r_1) + (h \wedge m) >>> r_2$$

where $\oplus$ is XOR, $\wedge$ is AND, + is addition (modulo a certain value), $\lll$ is left rotation, $\ggg$ is right rotation, and r1,r2 are rotation constants optimized for the target architecture. The full compression function C would apply this round function multiple times to different parts of Hi−1 and Mi , potentially with varying constants and key schedules (if it's a keyed hash).

6. FRAMEWORK IMPLEMENTATION AND EVALUATION

6.1. Implementation Details: Describe the implementation of the proposed framework on representative IoT platforms (e.g., microcontrollers like ARM Cortex-M, resource-constrained operating systems like Contiki-NG or Zephyr).

6.2. Experimental Setup: Detail the hardware and software used for evaluation. Describe the network topology and traffic patterns simulated or used in real-world testbeds.

6.3. Performance Metrics: Define the metrics used to evaluate the framework's performance and resource overhead: Computational Overhead: CPU cycles consumed by security operations. Memory Footprint: RAM and flash memory usage. Energy Consumption: Battery life impact. Communication Overhead: Additional bytes added to messages due to security protocols. Latency: Time taken for secure operations.

6.4. Evaluation Results: Present the results of the performance evaluation. Use graphs and tables to illustrate the findings. Compare the performance of the proposed lightweight framework with traditional security approaches (if applicable) or baseline scenarios.

6.5. Security Analysis: Discuss the security properties of the framework. Analyze its resilience against common IoT attacks based on the implemented security mechanisms. Provide arguments for the effectiveness of the lightweight approach in providing adequate security for pervasive IoT.

7. DISCUSSION

7.1. Trade-offs between Security and Resource Constraints: Discuss the inherent trade-offs involved in designing lightweight security solutions. Acknowledge that lightweight solutions may offer a lower level of security compared to heavyweight alternatives but are necessary for pervasive IoT.

7.2. Applicability to Different IoT Use Cases: Discuss how the framework can be adapted to different IoT use cases with varying security requirements and resource constraints.

7.3. Future Directions: Identify areas for future research and development, such as: Developing more advanced lightweight cryptographic algorithms. Exploring hardware-based security solutions for enhanced trust. Integrating machine learning for anomaly detection and proactive security. Addressing the challenges of large-scale key management in pervasive IoT. Developing formal methods for verifying the security of lightweight protocols.

RESULTS AND DISCUSSION

In this proposed research work we implemented Fernet lightweight cryptographic algorithm for encryption and authentication of sensor data generated in IoT network using Python. Fernet is a lightweight symmetric cryptographic method which is using symmetric encryption .It is based on symmetric AES-128 in a CBC mode which offers 128 bits multiple length and provide authentication using HMAC with SHA256 Hashing algorithm.

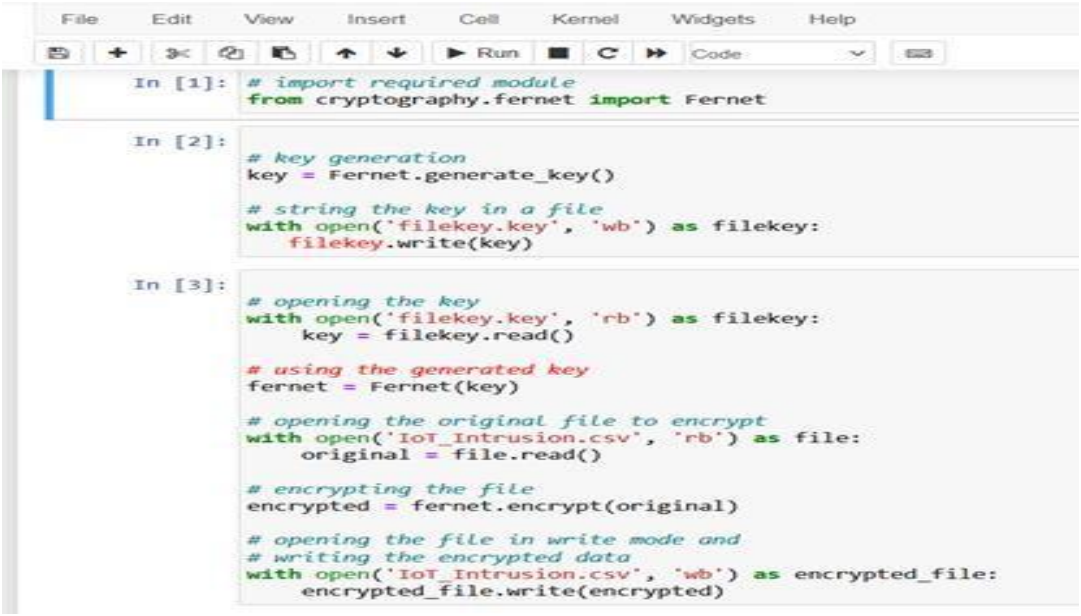

Fig.7.Implementation of Fernet Algorithm

Fig.7 shows our implementation uses Python to apply Fernet symmetric encryption.

Python facilities some libraries for generating keys. Fernet is a lightweight symmetric cryptographic technique which offers encryption as well as authentication .It generates secret key hence it is impossible to decrypt data without key which is shared only with authenticated recipient. Fernet supports private key which is a single key for encryption and decryption. To enhance security in this each character in keyundergoesbase-64URL-safeencoding.Encryption procedure performs Sub Bytes (), Row Shifts (), Add Round Key (), Mix Columns () functions. Also reverse of this is for decryption purpose.

CONCLUSION

In this proposed research work, we have implemented Fernet lightweight cryptographic algorithm for end to end secure cloud based IoT communication. This ensures a fully encrypted transmission of sensor data which will resolve a problem of confidentiality and integrity of data to large extent. In this research paper, after doing research on different Lightweight Cryptographic algorithms and Key management systems, we conclude to use Fernet lightweight symmetric cryptographic algorithm because Fernet uses two different algorithms: AES128 bits in CBC mode as encryption algorithm and HMAC using SHA256 authentication algorithm and having Key size of 256 bits which resolve our problem of secure communication of cloud based IoT network using lightweight cryptography.

REFERENCES

1. M. Rana ,Q.Mamun, R. Islam "Lightweight cryptography in IoT networks: A survey", Journal FutureGenerationComputerSystems,Vol.129,pp.77- 89,2022,doi.org/10.1016/j.future.2021.11.011
2. H. Tawalbeh , S. Hashish ,"Security in Wireless Sensor Networks Using Lightweight Cryptography", in Journal of Information Assurance and Security, ISSN 1554-1010 ,Volume 12, pp. 118-123, 2017.
3. Silva, V. A. Cunha, J. P. Barraca, R. L. Aguiar "Analysis of the Cryptographic Algorithms in IoT Communications", in Springer Information Systems Frontiers, doi: 10.1007/s10796-023-10383-9,2023.
4. A.Fotovvat, G.M.E.Rahman, S.S.VedaeiandK. A. Wahid, "Comparative Performance Analysis of Lightweight Cryptography Algorithms for IoT Sensor Nodes," in IEEE Internet of Things Journal, vol. 8,no. 10, pp. 8279-8290, 15 May15, 2021, doi: 10.1109/JIOT.2020.3044526.
5. V. A. Thakor, M. A. Razzaque and M. R. A. Khandaker, "Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review, Comparison and Research Opportunities," in IEEE Access, vol. 9, pp. 28177-28193, 2021, doi: 10.1109/ACCESS.2021.3052867.
6. Hussain, R. Hussain, S. A. Hassan and E. Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," in IEEE Communications Surveys & Tutorials, vol. 22, no. 3,
7. pp. 1686-1721, third quarter 2020, doi: 10.1109/COMST.2020.2986444.
8. Mishra, S.; Albarakati, A.; Sharma, S.K.," Cyber Threat Intelligence for IoT Using Machine Learning", Journal Processes,Vol.10,pp.2673 doi. 10.3390/pr10122673.
9. Makkar, S. Garg, N. Kumar, M. S. Hossain, A. Ghoneim and M. Alrashoud, "An Efficient Spam Detection Technique for IoT Devices Using Machine Learning," in IEEE Transactions on Industrial Informatics, vol. 17, no. 2, pp. 903-912, Feb. 2021, doi: 10.1109/TII.2020.2968927.

10. Liu, Y. Zhang, J. Xu, J. Zhao and S. Xiang, "Ensuring the Security and Performance of IoT Communication by Improving Encryption and Decryption With the Lightweight Cipher uBlock," in IEEESystemsJournal,vol.16,no.4,pp.5489-5500, Dec. 2022, doi: 10.1109/JSYST.2022.3140850.
11. Jamalipour and S. Murali, "A Taxonomy of Machine-Learning-Based Intrusion Detection Systems for the Internet of Things: A Survey," in IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9444-9466,15June15,2022,doi:10.1109/JIOT.2021.3126811.
12. Pathak, S. Saguna, K. Mitra and C. Åhlund, "Anomaly Detection using Machine Learning to Discover Sensor Tampering in IoT Systems," ICC 2021 - IEEE International Conference on Communications, Montreal, QC, Canada, 2021, pp.1-6, doi: 10.1109/ICC42927.2021.9500825.
13. G.Said, A.Ghani, A.Ullah, M.Azeem, M.Bilaland K. S. Kwak, "Light-Weight Secure Aggregated Data Sharing in IoT-Enabled Wireless Sensor Networks," in IEEE Access, vol. 10, pp. 33571-33585,2022, doi: 10.1109/ACCESS.2022.3160231.
14. R. Sivakumar, J. Jayapriya and N. Krishnan, "Comparison Study on SPN Type Light Weight Cryptography Algorithms for IoT," 2022International Conference on Inventive Computation Technologies (ICICT), Nepal, 2022, pp. 1051-1055, doi: 10.1109/ICICT54344.2022.9850849.
15. A. I. Regla and E. D. Festijo, "Performance Analysis of Light-weight Cryptographic Algorithms for Internet of Things (IoT) Applications: A Systematic Review," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp1-5, doi: 0.1109/I2CT54291.2022.9824108.
16. Elena Petrova, Predictive Analytics for Customer Churn in Telecommunications, Machine Learning Applications Conference Proceedings, Vol 1 2021.
17. Dhabliya,D.(2021).Feature Selection Intrusion
18. Weng and Chia-Hsin Cheng and Yung-Fa Huang,"A securitygatewayapplicationforEnd-to-EndM2M communications", Computer Standards & Interfaces, Vol.44, pp.85-93, 2016, doi.org/10.1016/j.csi.2015.09.001.
19. Anand,R.,Ahamad,S., Veeraiah,V.,Janardan,S.K., Dhabliya, D., Sindhwani, N., Gupta, A. Optimizing 6G wireless network security for effective communication (2023) Innovative Smart Materials Used in Wireless Communication Technology,pp.1- 20.

9. AUTHOR'S PROFILE

1. Sakshi Gangwar is pursuing Master of Technology in Computer Science & Engineering from Rajshree Institute of Management and Technology, Bareilly(U.P.), India,Affiliated Dr. A. P. J. Abdul Kalam Technical University, Lucknow(U.P.), India. Her area of interest includes Machine Learning and Networking.

2. Er. Harshit Gupta is Assistant Professor in Department of Computer Science & Engineering, Rajshree Institute of Management and Technology, Bareilly (U.P.), India. He is M. Tech. in Computer Science & Engineering. His areas of interest include ML, AI, IoT, ICT, Block chain Technologies, fuzzy/Neural Networks, Panda, Big Data, Data Analytics, and Pattern Recognition. He has published more than 50 Papers/Chapters in national & international Journals/Conferences.

3. Dr. Ruchin Jain is Head of Department of Computer Science & Engineering, Rajshree Institute of Management and Technology, Bareilly (U.P.), India. He has more than 20 years research and national & international teaching Experience.