Saurav Verma, et. al. International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 13, Issue 4, April 2023, pp. 01-07

RESEARCH ARTICLE OPEN ACCESS

An analysis of modeling techniques and security algorithm for machine-to-machine communication in IoT network

Saurav Verma*, Dr.Chetana Prakash**
**(Department of Information technology, MPSTME NMIMS University, India)*
*** (Department of CS&E, BIET, VTU University, India)*

ABSTRACT
The Internet of Things (IoT) is a network of physical objects such as cars, appliances, and other household items that are connected to one another and exchange data. These objects are implanted with electronics, software, sensors, and connectivity. The concept of IoT has revolutionized the way we interact with technology and the world around us, allowing for smarter and more efficient automation and control of various aspects of our lives. With the growth of IoT devices and platforms, there is a greater need for security and privacy measures to protect sensitive data and prevent malicious attacks. The Internet of Things (IoT) has the potential to change a variety of industries, including healthcare, agriculture, transportation, and manufacturing.

Keywords – Internet of Things, Machine-to-Machine, Security, IoT Models, Hybrid algorithms.

Date of Submission: 18-03-2023 Date of acceptance: 03-04-2023

I. INTRODUCTION

IoT communication refers to the process of exchanging data between devices and sensors in an Internet of Things (IoT) network. The data exchanged can be used to monitor, control, and optimize various aspects of the physical world.

IoT communication technologies consist of Wi-Fi, Bluetooth, Zigbee, cellular networks, satellite communication, and more. The choice of communication technology depends on various factors such as range, bandwidth, power consumption, and data security. IoT communication is a crucial aspect of IoT systems, enabling devices to exchange data and work together seamlessly to make intelligent decisions and optimize the physical world. IoT communication can occur in several ways, including:

a) Device-to-Device Communication: Devices in an IoT network can communicate directly with each other without the need for a central hub or gateway. This type of communication is often used in peer-to-peer (P2P) networks, where devices share information and resources directly.

b) Device-to-Cloud Communication: Devices in an IoT network can also communicate with cloud-based servers or services. This type of communication is often used to store and analyze data collected from devices in the network.

c) Device to Gateway Communication: Devices in an IoT network can communicate with a gateway device that acts as a central hub for the network. The gateway device can then transmit data to other devices in the network or to cloud-based servers.

d) Machine to Machine Communication: Without human interaction, machines or other devices exchange data through machine-to-machine (M2M) communication. In order for devices to gather, process, and interact with one another in order to make decisions and take actions based on the data, this form of communication is essential for IoT networks.

M2M communication between machines is a crucial component of the Internet of Things (IoT). Without human interaction, gadgets or machines exchange data in this scenario. M2M communication allows

machines to collect data, process it, and communicate with other devices to make decisions and take actions based on the data.

In an IoT network, M2M communication occurs through various sensors, devices, and gateways that collect data from the physical world. This data is then transmitted to other machines and devices for further processing and analysis. For example, a smart home system may use M2M communication to connect various sensors, such as motion sensors and temperature sensors, to a central hub. The hub can then use this data to control the temperature and lighting in different rooms of the house automatically.

Several communication technologies, including Wi-Fi, Bluetooth, cellular networks, and satellite communication, can enable M2M communication in an IoT network. Each technology has its advantages and disadvantages and can be appropriate for various IoT applications.

Overall, M2M communication is a critical component of IoT systems, enabling devices and machines to work together seamlessly and make intelligent decisions based on data.

II. Architecture of M2M

Machine-to-Machine (M2M) architecture is a communication system that allows devices or machines to exchange data and interact with each other without human intervention. The architecture of M2M involves various components and layers that work together to facilitate seamless communication between machines. Here is a brief overview of the M2M architecture:

I. Devices: The devices or machines are the primary components of the M2M architecture. They can be sensors, actuators, machines, or any other type of device that can transmit and receive data.

II. Connectivity: The connectivity layer provides the infrastructure for connecting the devices. This layer can include wired or wireless networks, such as cellular networks, Wi-Fi, Bluetooth, or ZigBee.

III. Communication protocols: The communication protocols layer defines the rules and procedures for transmitting data between devices. Examples of communication protocols include MQTT, CoAP, HTTP, and WebSocket.

IV. Application enablement: The application enablement layer provides the tools and resources for developers to build M2M applications. It includes APIs, software development kits (SDKs), and other resources that simplify the process of developing M2M applications.

V. Data management: The storing, processing, and administration of data produced by the devices are the responsibilities of the data management layer. Data can be stored, processed, and analysed using databases, cloud platforms, and analytics tools, which are all part of this layer.

VI. Security: The security layer offers safeguards for the M2M system against unapproved access, data breaches, and other security risks. In addition to additional security measures, it has systems for authentication and authorization.

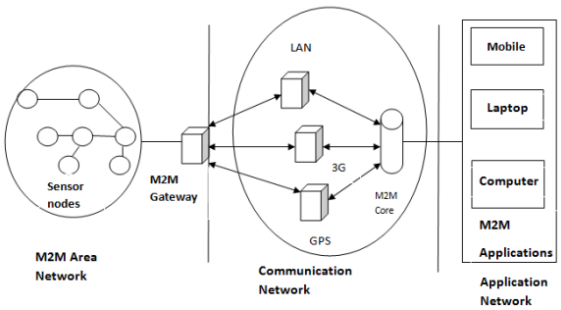

Fig.1 Architecture of M2M

III. Security in M2M

Data sharing between two or more devices without human interaction is referred to as M2M (Machine-to-Machine) communication. Because it makes transactions and trading quicker and more efficient, this sort of communication is becoming more and more significant in the securities business.

One of the main securities issues related to M2M communication is the potential for security breaches. Since M2M communication involves multiple devices communicating with each other, it creates more potential entry points for hackers to exploit. This is especially concerning in the securities industry, where sensitive financial data is exchanged.

Another issue is the need for standardization. With so many different devices and systems communicating with each other, it can be difficult to ensure that data is being transmitted and interpreted correctly. This can lead to errors or inconsistencies in data, which can have serious consequences in the securities industry.

The issue of interoperability is the last one. It may be challenging for various devices and systems to connect with one another because they use various communication protocols. This can be a significant barrier to the adoption of M2M communication in the securities industry, as it requires a high degree of coordination and standardization across multiple systems and devices.
There are various types of M2M threats that can affect the security of machine-to-machine communication. Here are some of the most common ones:

a. Data interception and eavesdropping: Hackers may try to intercept and eavesdrop on M2M communication to steal sensitive information such as financial transactions, authentication credentials, or other valuable data.

b. Denial-of-service attacks: Attackers may try to overload M2M systems with a flood of requests or traffic, causing them to crash or become unavailable, disrupting critical business processes.

c. Malware and viruses: Malicious code can be injected into M2M devices or systems, which can cause damage, data theft or manipulation, or unauthorized access.

d. Device spoofing: Attackers can create fake devices or manipulate the identification of devices in the M2M network to gain access to sensitive information or disrupt legitimate transactions.

e. Physical tampering: M2M devices may be physically vulnerable to tampering or theft, leading to security breaches, data theft or destruction, or other types of malicious activities.

f. Insider threats: Employees or third-party contractors with authorized access to M2M networks may deliberately or accidentally misuse or compromise the system, leading to data breaches or other security incidents.

g. Lack of security updates: M2M devices may have vulnerabilities that can be exploited if they are not updated with the latest security patches and firmware updates, leaving them open to attacks.

To combat these and other possible M2M risks and guarantee the security and integrity of M2M communication, extensive security measures must be put in place.

IV. TYPES OF MODELS

There are several modeling techniques that can be used to secure machine-to-machine communication in an IoT network. Here are a few:

a) Access control models: Based on user identification, role, or other factors, access to resources can be restricted using access control models. Based on the identity of the devices, users, or applications, access control models can be used in the context of IoT to limit access to devices, data, and network resources. Role-based access control (RBAC), attribute-based access control (ABAC), and other access control mechanisms can be used to construct access control models.

b) Encryption: A method for securing both data at rest and in transit is encryption. Encryption can be used in an IoT network to safeguard data stored on the devices as well as communication between them. Hashing, symmetric encryption,

and asymmetric encryption are a few of the different encryption algorithms and methods that can be utilized.

c) Authentication and authorization models: Authentication and authorization models are used to verify the identity of devices, users, or applications and to authorize access to resources. In the context of IoT, authentication and authorization models can be used to ensure that only authorized devices, users, or applications can access the network and its resources.

d) Firewalls and intrusion detection systems: IoT networks may be monitored and secured from unauthorized access and attacks using firewalls and intrusion detection systems (IDS). IDS can be used to identify suspicious network activity and take appropriate action, whereas firewalls can be used to limit access to the network based on source IP addresses, ports, and protocols.

e) Virtual private networks (VPNs): VPNs can be used to create secure connections between devices and networks over the internet. In an IoT network, VPNs can be used to provide secure communication between devices located in different geographical locations.

f) Trust models: Trust models can be used to establish and maintain trust between devices, users, or applications in an IoT network. Trust models can be based on a variety of factors, such as device reputation, user behavior, and security posture.

Overall, a combination of these modeling techniques can be used to provide robust security for machine-to-machine communication in an IoT network.

V. Hybrid Algorithms

A hybrid algorithm for machine-to-machine (M2M) communication is a combination of different cryptographic algorithms that work together to ensure secure and efficient communication between machines. Here are some common hybrid algorithms used in M2M communication:

1. **RSA and AES:** This hybrid algorithm uses the RSA algorithm for key exchange and AES for data encryption. RSA is a public-key cryptographic algorithm that uses a pair of keys (public and private) for secure key exchange between machines. AES is a symmetric key encryption algorithm that encrypts the data using the key generated by the RSA algorithm.

2. **ECC and AES:** Elliptic Curve Cryptography (ECC) and AES are both used in this hybrid technique to exchange keys and encrypt data. ECC is a computationally efficient public-key cryptography technique that offers the same level of security as RSA with reduced key sizes.

3. **ChaCha20-Poly1305**: This hybrid stream cypher combines the Poly1305 algorithm for data authentication with the ChaCha20 method for data encryption. Whereas the Poly1305 algorithm creates a message authentication code (MAC) to guarantee data integrity, the ChaCha20 algorithm symmetric key encryption scheme employs a 256-bit key to encrypt the data.

4. **HMAC-SHA256**: To provide secure and effective data authentication, this hybrid technique combines the Secure Hash Algorithm (SHA) and the Hash-based Message Authentication Code (HMAC) algorithm. To maintain data integrity, the HMAC-SHA256 algorithm creates a MAC using a secret key and the SHA-256 hash function.

Overall, hybrid algorithms provide a higher level of security and efficiency compared to single algorithms, and they can be tailored to specific use cases and security requirements in M2M communication. However, the selection of a hybrid algorithm must consider factors such as performance, computational overhead, and compatibility with the M2M communication protocol used.

VI. Comparison of Hybrid Algorithm

There are various hybrid algorithms available for machine-to-machine (M2M) communication that combine different cryptographic techniques to

provide secure and efficient communication between devices. Here's a comparison of popular hybrid M2M algorithms:

a) **RSA-AES vs ECC-AES:** In contrast to ECC-AES, which utilises the Elliptic Curve Cryptography (ECC) technique for key exchange and AES for data encryption, RSA-AES is a hybrid approach that uses the RSA algorithm for key exchange and AES for data encryption. Both methods offer high levels of security, but ECC-AES has a smaller key size than RSA-AES and is therefore more computationally effective and suitable for low-power M2M devices.

b) **ChaCha20-Poly1305 vs AES-GCM:** ChaCha20-Poly1305 is a stream cipher that combines the ChaCha20 algorithm for data encryption and the Poly1305 algorithm for data authentication, while AES-GCM is a block cipher that combines the AES algorithm for data encryption and the Galois/Counter Mode (GCM) algorithm for data authentication. Both algorithms provide secure and efficient communication, but ChaCha20-Poly1305 has better performance on low-power M2M devices due to its lower computational overhead.

c) **HMAC-SHA256 vs CMAC-AES**: HMAC-SHA256 is a hybrid algorithm that combines the Hash-based Message Authentication Code (HMAC) algorithm with the Secure Hash Algorithm (SHA) for data authentication, while CMAC-AES combines the Cipher-based Message Authentication Code (CMAC) algorithm with the AES algorithm for data authentication. Both algorithms provide secure and efficient authentication, but HMAC-SHA256 has better performance on low-power M2M devices due to its lower computational overhead.

In general, the choice of a hybrid M2M algorithm depends on factors such as security requirements, computational overhead, and compatibility with the M2M communication protocol used. It is essential to select an algorithm that provides the required security while also being efficient enough to work on low-power M2M devices.

VII. IoT Security Requirements

To address the security challenges in IoT, there are several security requirements that must be considered:

a) Authentication and Authorization: IoT devices should have strong authentication mechanisms to ensure that only authorized users or devices can access them. This can include password protection, two-factor authentication, and certificate-based authentication.

b) Encryption: Strong encryption should be used by IoT devices to safeguard data both in transit and at rest. This may involve the use of secure protocols like HTTPS and SSL/TLS as well as the encryption of data using powerful algorithms like AES.

c) Secure Boot and Firmware Updates: IoT devices should have secure boot processes that ensure the device's software has not been tampered with or compromised. Devices should also be able to receive firmware updates securely to address any security vulnerabilities.

d) Access Control: IoT devices should have access controls that restrict what actions can be performed by different users or devices. This can include role-based access control and device-to-device communication authentication.

e) Monitoring and Logging: IoT devices should be able to log security events and generate alerts for unusual or suspicious activity. This can include monitoring network traffic, device performance, and security events.

f) Privacy: IoT devices should be designed with privacy in mind, and should limit the

collection, storage, and transmission of personal data. They should also have mechanisms in place to enable users to control their own data.

g) Compliance: IoT devices must adhere to all applicable security and privacy laws and regulations, including the General Data Protection Regulation (GDPR) and the Payment Card Industry Data Security Standard (PCI DSS).

Overall, IoT security requirements must be considered at every stage of the device lifecycle, from design to deployment and beyond. By prioritizing security requirements, IoT devices can be developed and deployed in a way that ensures their security, privacy, and usability.

VIII. Conclusion

Several industries, including healthcare, agriculture, and transportation, stand to benefit from the Internet of Things (IoT). IoT can provide new insights and efficiencies that were previously not achievable by connecting physical items to the internet and enabling them to gather and exchange data. However, IoT also presents a number of security and privacy challenges. To ensure the widespread adoption and success of IoT, it is crucial to prioritize security requirements at every stage of the device lifecycle, from design to deployment and beyond.

By implementing strong authentication mechanisms, encryption, secure boot processes, access controls, monitoring and logging, privacy protections, and compliance with regulations and standards, IoT devices can be developed and deployed in a way that ensures their security, privacy, and usability.

Overall, IoT represents a major opportunity for innovation and growth, but it must be approached with a focus on security and privacy in order to realize its full potential.

Reference

[1]. Yisroel Mirsky, Tomer Golomb, Yuval Elovici, "Lightweight collaborative anomaly detection for the IoT using blockchain“ Journal of Parallel and Distributed Computing, vol. 145, pp. 75-97, Nov 2020.

[2]. Syed Rizvi, RJ Orr, Austin Cox, Prithvee Ashokkumar, Mohammad R. Rizvi, "Identifying the attack surface for IoT network" Internet of Things, vol. 9, Art.no. 100162, March 2020

[3]. N. Ravi and S. M. Shalinie, "Learning-Driven Detection and Mitigation of DDoS Attack in IoT via SDN-Cloud Architecture,“ IEEE Internet of Things Journal, vol. 7, no. 4, pp. 3559-3570, April 2020, doi: 10.1109/JIOT.2020.2973176.

[4]. A. Tandon, T. J. Lim and U. Tefek, "Sentinel based malicious relay detection in wireless IoT networks," Journal of Communications and Networks, vol. 21, no. 5, pp. 458-468, Oct. 2019, doi: 10.1109/JCN.2019.000049.

[5]. V.Tejaswini and Dr.D.Susitra, "Hybrid PSO-WOA for Solving ORPD Problem under Unbalanced Conditions", Journal of Computational Mechanics, Power System and Control, Vol.2,No.2, pp.10-20,2019.

[6]. S. Latif, Z. Zou, Z. Idrees and J. Ahmad, "A Novel Attack Detection Scheme for the Industrial Internet of Things Using a Lightweight Random Neural Network," IEEE Access, vol. 8, pp. 89337-89350, 2020, doi: 10.1109/ACCESS.2020.2994079

[7]. Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, Yanghui Rao, “Attentional Encoder Network for Targeted Sentiment Classification” , arXiv preprint arXiv:1902.09314, 2019

[8]. Jialong Tang, Ziyao Lu, Jinsong Su, Yubin Ge, Linfeng Song, Le Sun, Jiebo Luo, “Progressive Self-Supervised Attention Learning for Aspect-Level Sentiment Analysis”, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages: 557–566, 2019

[9]. Binxuan Huang, Yanglan Ou and Kathleen M. Carley, “Aspect Level Sentiment Classification with Attention-over-Attention Neural Networks”, International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation, Springer, Pages: 197-206, 2018

[10]. Xin Li, Lidong Bing, Wai Lam, Bei Shi, “Transformation Networks for Target-Oriented Sentiment Classification”, arXiv preprint arXiv:1805.01086, 2018

[11]. Yuxiao Chen, Jianbo Yuan, Quanzeng You, Jiebo Luo, “Twitter Sentiment Analysis via Bi-sense Emoji Embedding and Attention-based LSTM”, MM '18: Proceedings of the 26th ACM international conference on Multimedia Pages 117–125, 2018

[12]. Li X, Rao Y, Xie H, Lau RYK, Yin J, Wang FL, "Bootstrapping social emotion classification with semantically rich hybrid neural networks", IEEE Trans Affect Comput 8(4):428–442,2017.
[13]. Shailendra Rathore, Jong Hyuk Park, "Semi-supervised learning based distributed attack detection framework for IoT" Applied Soft Computing,vol. 72, pp. 79-89, Nov 2018
[14]. Mahmudul Hasan, Md. Milon Islam, Md Ishrak Islam Zarif, M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches" Internet of Things, vol. 7, Art.no. 100059, Sep 2019.
[15]. A. Y. Khan, R. Latif, S. Latif, S. Tahir, G. Batool and T. Saba, "Malicious Insider Attack Detection in IoTs Using Data Analytics," IEEE Access, vol. 8, pp. 11743-11753, 2020, doi: 10.1109/ACCESS.2019.2959047.
[16]. Yisroel Mirsky, Tomer Golomb, Yuval Elovici, "Lightweight collaborative anomaly detection for the IoT using blockchain" Journal of Parallel and Distributed Computing, vol. 145, pp. 75- 97, Nov 2020.
[17]. Syed Rizvi, RJ Orr, Austin Cox, Prithvee Ashokkumar, Mohammad R. Rizvi, "Identifying the attack surface for IoT network" Internet of Things, vol. 9, Art.no. 100162, March 2020