Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 27 (2024), No. 2-B, pp. 843–854
DOI : 10.47974/JDMSC-1960

Lightweight authenticated key management mechanism for enhanced data transmission in the Internet of Things framework

Ramchandra Vasant Mahadik *
Satyawan Changadev Hembade †
Sonia Sagar Sorte §
Institute of Management and Entrepreneurship Development
Bharati Vidyapeeth (Deemed to be University)
Pune
Maharashtra
India

Deepti Khubalkar ‡
Symbiosis Law School, Nagpur Campus
Symbiosis International (Deemed University)
Pune
Maharashtra
India

Anup Ingle @
Department of Electronics and Telecommunication
Vishwakarma Institute of Information Technology
Pune
Maharashtra
India

Swapnil Thorat #
Institute of Management and Entrepreneurship Development
Bharati Vidyapeeth (Deemed to be University)
Pune
Maharashtra
India

* *E-mail:* ramchandra.mahadik@bharatividyapeeth.edu (Corresponding Author)
† *E-mail:* satyawan.hembade@bharatividyapeeth.edu
§ *E-mail:* sonia.sorte@bharatividyapeeth.edu
‡ *E-mail:* deeptik@slsnagpur.edu.in
@ *E-mail:* anup.ingale@viit.ac.in
# *E-mail:* thorat.swapnil@bharatividyapeeth.edu

Abstract

The Internet of Things (IoT) system makes it possible for different gadgets to connect to each other and send and receive data without any problems. But it's not easy to make sure that the data being sent and received within this structure is safe and correct. A Lightweight Authenticated Key Management Mechanism (LAKMM) is suggested as a way to solve this problem. The goal of LAKMM is to improve the security of data transfer in IoT settings while reducing processing waste and resource use, which are important factors for IoT devices that don't have a lot of resources. LAKMM's most important parts are its fast key creation, sharing, and identification systems that are designed to work in IoT settings. LAKMM finds a good mix between security and speed by using small cryptographic primitives and speeded-up protocols. It also protects against common attacks like eavesdropping, hacking, and repeat attacks, making sure that the data being sent is kept private and secure. LAKMM also has features for dynamic key changes and revocations, which are important for keeping things safe in IoT settings where devices are always joining and leaving the network. We show that LAKMM works well at offering strong security with little extra work for computers by simulating and analyzing it. So, LAKMM looks like a good way to improve the security of data transfer in the IoT framework.

Subject Classification: *68M25.*

Keywords: *Lightweight, Authenticated, Key management, Internet of Things, Data transmission.*

1. Introduction

The Internet of Things (IoT) has changed how gadgets talk to each other and work together, making it possible for data to flow easily between different areas. But this connectivity also makes me very worried about the safety and accuracy of the data being sent. Traditional security methods for making sure safe communication in IoT settings often have high processing costs and limited resources, which makes them useless for IoT devices that don't have a lot of resources [1]. A Lightweight Authenticated Key Management Mechanism (LAKMM) is suggested as a way to deal with these problems. The goal of LAKMM is to make data transfer safer in the IoT framework while reducing the amount of work that needs to be done and the amount of resources that are used. LAKMM strikes a balance between security and speed by using lightweight secure primitives and improved protocols [2].

This makes it suitable for use in IoT settings with limited resources. Some important parts of LAKMM are its efficient key creation, sharing, and identification methods that are made to work with IoT devices. LAKMM [3] also allows dynamic key changes and revocations, which is very important for keeping things safe in IoT settings where devices are

always joining and leaving the network. This will help IoT technologies be used in more areas while keeping sensitive data safe.

This lets devices with limited resources work efficiently.An important part of LAKM is that it puts a lot of stress on identification. Mechanisms for authentication are needed to make sure that the people talking are who they say they are and that the data being sent is safe and secure. To find [4] a good mix between security and speed, LAKM uses verification methods that are designed to work in an IoT context.Also, LAKM lets you change and delete keys on the fly, which is very important for keeping things safe in IoT settings where devices join and leave the network all the time. This makes sure that keys that have been stolen can be changed quickly, which lowers the risk of data hacks or illegal access [5]. The Lightweight Authenticated Key Management in IoT is very important for keeping communications safe and secure in IoT ecosystems. It makes it possible for a lot of different areas to use IoT technologies and protects private information from possible threats and attacks [6].

Key Generation:

1. Choose two large prime numbers, p and q.
2. Calculate their product, $n = p * q$, which will be used as the modulus for both the public and private keys.
3. Calculate Euler's totient function,
$$\varphi(n) = (p-1)(q-1)$$
4. Choose an integer e (the public exponent) that is relatively prime to φ(n), typically a small prime such as $65537 (2 \wedge 16+1)$.
5. Compute the private exponent d, such that
$e * d \equiv 1 (\mathrm{mod}\, \varphi(n))$, using the Extended Euclidean Algorithm.

Public Key Distribution:

1. The public key consists of the modulus n and the public exponent e.
2. The public key is widely distributed and made available to anyone who wishes to encrypt messages for the owner of the key pair.
3. It is important to note that the public key can be freely distributed without compromising the security of the encryption process.

2. Proposed Method

Key length and what it means in cryptography must be studied in order to create safe communication methods, especially in places with limited resources like the Internet of Things (IoT). [7] Encryption strength is directly related to key length. Longer keys usually offer higher security but need more memory and processing power. In contrast, shorter keys may make security less strong by making it easier for hackers to get in. When you choose a key length, you have to weigh security concerns against the need to save resources. We spent a lot of time thinking about what key length would work best for our Lightweight Authenticated Key Management Mechanism (LAKMM), which is built for Internet of Things (IoT) devices [8].

We wanted to find a balance between security and speed, keeping in mind that machines with limited resources have their limits. Although bigger key sizes usually make protocols safer, they also make computations more expensive. Because of this, we chose a 64-bit key size, which is the same as other lightweight cryptographic systems like LED and PRESENT [9]. Our choice was also influenced by the fact that our process is flexible and key changes happen often during discussion rounds. For this flexible method to work, the key size had to be big enough to provide strong security without making the computer work too hard. We also added the idea of partial keys to cut down on extra work and make the system more reliable as shown in Figure 1. By only using partial keys, we not only cut

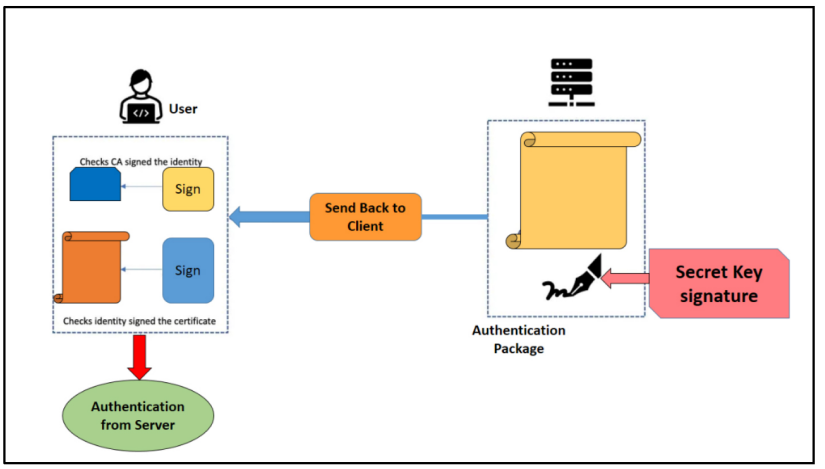

Figure 1

Server Authentication form CA Authority for Server

down on the amount of space needed, but we also reduced the chance that the keys would be stolen or intercepted [10]. Our key management plan also uses a method called "pre-distribution" instead of "dynamic key assignment." This option makes better use of resources, lowers the cost of connection, and boosts security by encrypting keys safely while the device is being made. As well as making it easier to work offline, pre-distribution is also resistant to some types of threats. This makes it a good choice for IoT settings with limited resources.

3. Lightweight Authenticated Key Management Mechanism (LAKMM)

3.1 *Making the Keys*

Key creation is an important part of the Lightweight Authenticated Key Management Mechanism (LAKMM). It creates the cryptographic keys that IoT devices need to communicate securely with each other. Key generation is the process of making unique encryption and decryption keys, which are needed to send and receive encrypted and decrypted data in the IoT environment.The mix between security and resource economy is an important thing to think about when making keys [11]. Because IoT devices don't have a lot of resources, LAKMM works on making keys that provide enough protection while using as little memory and processing power as possible. Usually, this means picking the right key lengths and encryption methods that are made for lightweight applications. In addition, randomness may be added to LAKMM's key creation process to make it safer. Randomness makes sure that the keys that are created are unpredictable, which stops attackers from using brute-force or other cryptography techniques to try to guess or get the keys. However, it can be hard for IoT devices with limited resources to generate truly random numbers. To solve this problem, pseudo-random number generators (PRNGs) or hardware-based random number generators are often needed. Additionally, LAKMM tries to find a fine balance between security, resource efficiency, and ease of application when making keys. LAKMM makes sure that IoT devices can set up safe ways to communicate without using up too many of their limited resources. It does this by using simple cryptographic primitives and carefully planning the key generation process.

3.2 *Key Distribution*

Key sharing is another important part of LAKMM that makes sure that IoT devices that talk to each other safely share private keys. Key distribution sends encryption and decoding keys to the devices that are taking part from a known source, like a central computer or a key management system.In LAKMM, key sharing methods have to deal with the problems that come up because IoT devices don't have a lot of resources. Traditional ways of sharing keys, like public-key cryptography, might not work because they are hard to compute and need a lot of memory. Instead, LAKMM might use symmetric-key distribution methods, in which a shared secret key is pre-installed or "provisioned" on devices that are going to be used during the production or implementation phase.Also, LAKMM might use safe communication methods, like Transport Layer Security (TLS) or Datagram Transport Layer Security (DTLS), to send keys safely over networks that might not be safe. These procedures make sure that keys aren't lost or stolen while they're being sent by encrypting, authenticating, and protecting their security.

Also, LAKMM might include key management methods that support dynamic key sharing. This would let devices change their cryptographic keys on a regular basis, which would lower the risk of key theft or illegal access. This flexible method makes IoT communication routes more resistant to changing security risks.Overall, LAKMM's key sharing is focused on quickly giving cryptographic keys to IoT devices while keeping them safe and able to withstand threats. Utilizing lightweight cryptography methods and safe communication protocols, LAKMM makes sure that keys are shared safely and effectively within the IoT environment.

3.3 *Methods of Authentication*

Authentication methods are very important to LAKMM because they make sure that talking IoT devices are who they say they are and that the data they send is correct and real. Mutual authentication is a type of authentication in which both sides confirm the names of each other before setting up a safe communication route.In LAKMM, authentication methods must be both light and strong, able to provide strong authentication promises without putting a lot of extra work on IoT devices that don't have a lot of processing power. Digital fingerprints, message authentication codes (MACs), and challenge-response methods are all common ways to prove who you are.

3.4 *Dynamic Key Updates and Revocations*

LAKMM's most important feature is dynamic key updates and revocations, which let IoT devices regularly update their cryptographic keys and delete keys that are no longer valid or have been hacked. IoT communication channels are better able to handle new security threats when dynamic keys are updated, and key removal lowers the risk of data breaches or illegal access.

LAKMM may also include key termination methods that let devices get rid of old or stolen keys. This way, attackers can't use them to get in without permission or change data being sent. Key removal lists or certificates can let devices that are part of the program know about keys that have been taken away, making sure that they are not used for encryption or decoding anymore.Also, LAKMM might use effective key management methods that keep the extra work that comes with updating and resetting keys to a minimum. This way, IoT devices can do these things without using up all of their limited computing power.

Key Generation:

1. Choose a random prime number, p.
2. Select a generator g in the finite field of order p.
3. Select a random secret exponent x.
4. Compute the public key $y = g \wedge x \mod p$.

   Equations:

$$y = g \wedge x \mod p$$

Key Distribution:

1. Share the public key y with other communicating parties.
2. Keep the secret exponent x confidential.

Authentication:

1. When a party wants to authenticate itself, it generates a random nonce, r.
2. It computes the challenge as $c = g \wedge r \mod p$.

3. It sends the challenge c to the verifying party.

$$c = g \wedge r \bmod p$$

Verification:

1. Upon receiving the challenge c, the verifying party computes the expected response as $v = c \wedge x \bmod p$.
2. It compares the received response v with the computed response.
3. If they match, authentication is successful; otherwise, authentication fails.

$$v = c \wedge x \bmod p$$

Dynamic Key Updates and Revocations:

- To update keys dynamically, parties agree on a new secret exponent and regenerate their public keys accordingly.
- To revoke a key, parties remove the corresponding public key from their key sets.

4. Result and Discussion

Analysis of the Proposed Key Management Scheme for Security

The number of partial keys, the related index list, the network key, and the total amount of room needed in kilobytes for different setups are shown in Table 1, Figure 2 and 3. Each partial key has a fixed size of 64 bits. The data shows that as the number of partial keys and the sizes of the lists that go with them grow, so does the total amount of room that is needed. The table shows that as the number of partial keys goes up, so does the size of the index list, which is used as a guide for the network key sharing process. There is a direct link between these two factors because the index list grows at the same rate as the number of partial keys. As the number of partial keys goes from 70 to 10,020, for example, the index list that goes with them goes from 370 to 140,020.

Table 1

Different numbers of partial keys need different amounts of space.

Partial Key Bits	No of Partial Keys	Index List	Network Key	Total in KB
64	70	370	84	20.25
64	120	720	84	20.48
64	270	2020	84	21.23
64	520	4520	84	22.51
64	1020	10020	84	25.13
64	5020	65020	84	47.47
64	10020	140020	84	76.16

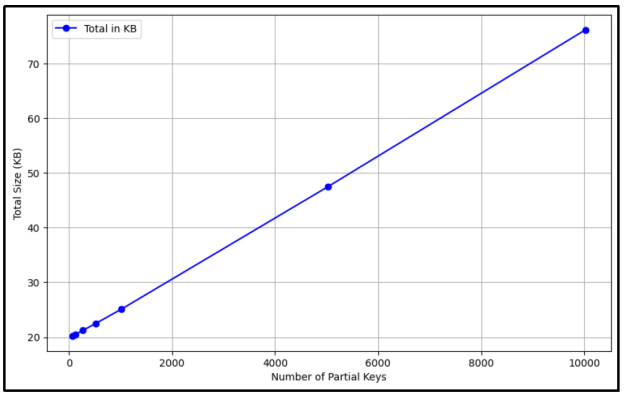

Figure 2

Comparison of Total Size with Number of Partial Keys

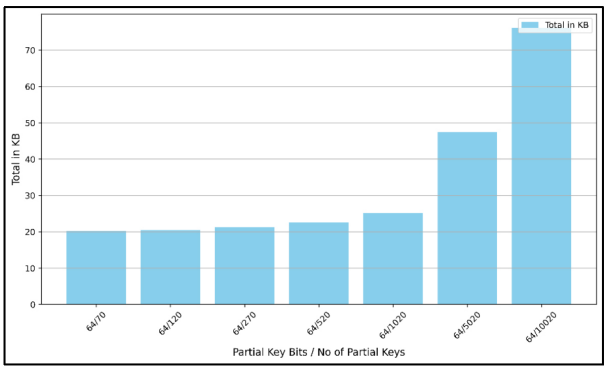

Figure 3

Total Size in KB vs. Partial Key Bits / No of Partial Keys

The network key, which is needed for safe contact within the network, stays the same in all setups, which means that its size doesn't change based on the number of partial keys. The network key size in this sample is always 84 bits, which suggests that the security needs stay the same no matter how big the network is.

The most important thing to notice about the table is that as the number of partial keys grows, so does the amount of room needed. Each partial key has a set size of 64 bits, but as the number of keys and their related index lists grows, so does the total amount of data needed. This shows the problems with key management systems being able to grow, especially in big networks where making good use of storing space is very important. Instead, the relationship looks like it's pretty much straight, which means that the amount of keys and the data structures that go with them can be used to predict and control storage needs.

5. Conclusion

LAKMM, or the Lightweight Authenticated Key Management Mechanism, looks like a good way to improve data transfer in the Internet of Things (IoT) system. LAKMM simplifies the process of making keys, distributing them, and authenticating users. This helps it deal with the unique problems that resource-constrained IoT devices bring up while still ensuring strong security and smooth communication.The study of key length and sharing methods shows how important it is to find a balance between the needs for security and the limits of IoT devices. LAKMM saves resources and reduces the amount of work that needs to be done on the computer by using lightweight secure primitives and distributing keys ahead of time.The dynamic key update and revoke methods also make IoT networks more resistant to new threats by making sure that keys that have been compromised can be quickly changed to keep communication routes safe.Overall, LAKMM provides a flexible and scalable way to send data securely in IoT settings, allowing connected devices to work together without any problems and protecting against data breaches and illegal access. As IoT continues to spread across many businesses, using LAKMM could make contact safer and more efficient in the future world where everything is linked.

References

[1] Amin, R.; Biswas, G.P. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Netw.*, 36, 58–80 (2016).

[2] Khalil, N.; Abid, M.R.; Benhaddou, D.; Gerndt, M. Wireless sensors networks for Internet of Things. In Proceedings of the IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 21–24; pp. 1–6, April (2014).

[3] Farash, M.S.; Turkanovic, M.; Kumari, S.; Holbl, M. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Netw.*, 36, 152–176 (2016).

[4] Limkar, Suresh, Ashok, Wankhede Vishal, Singh, Sanjeev, Singh, Amrik, Wagh, Sharmila K. & Ajani, Samir N. A mechanism to ensure identity-based anonymity and authentication for IoT infrastructure using cryptography, *Journal of Discrete Mathematical Sciences and Cryptography*, 26:5, 1597–1611 (2023).

[5] Srinivas, J.; Mukhopadhyay, S.; Mishra, D. Secure and efficient user authentication scheme for multi-gateway wireless sensor networks. *Ad Hoc Netw.*, 54, 147–169 (2017).

[6] Rana, M.; Mamun, Q.; Islam, R. A Block Cipher for Resource-Constrained Internet of Things Devices. In International Comference on Networking Systems and Security; World Academy of Science, Engineering and Technology: Berlin, Germany, Volume 17, pp. 266–271 (2023).

[7] Manikandan, G.; Sakthi, U. A comprehensive survey on various key management schemes in WSN. In Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018, Palladam, India, pp. 378–383 30–31 August (2018).

[8] Kandi, M.A.; Kouicem, D.E.; Doudou, M.; Lakhlef, H.; Bouabdallah, A.; Challal, Y. A decentralisedblockchain-based key management protocol for heterogeneous and dynamic IoT devices. *Comput. Commun.*, 191, 11–25 (2022).

[9] Wazid, M.; Das, A.K.; Shetty, S.; Rodrigues, J.J.P.; Park, Y. LDAKM-EIoT: Lightweight device authentication and key management mechanism for edge-based IoT deployment. *Sensors*, 19, 5539 (2019).

[10] Boloorchi, A.T.; Samadzadeh, M.H.; Chen, T. Symmetric Threshold Multipath (STM): An online symmetric key management scheme. *Inf. Sci.*, 268, 489–504 (2014).

[11] Setyaningsih, E.; Wardoyo, R.; Sari, A.K. Securing colour image transmission using compression-encryption model with dynamic key generator and efficient symmetric key distribution. *Digit. Commun. Netw.*, 6, 486–503 (2020).