Key Management architecture for Zigbee Light Link Protocol

Xiaoyao Guan, Maitrey Talware, Xiaoxuan Liu, Bhavnesh Gupta, Jay Kim

Abstract: Zigbee is one of the most popular protocols for connecting IoT devices. However, ZigBee has security weaknesses.
We find ZLL as emerging communication technology in an increasing number of smart lighting devices. With the rapidly growing number of IoT devices, ZLL can fix some security issues which occur in the Zigbee network. However, ZLL has some of its own security issues. Thus, we build a ZLL threat model to get its vulnerability and then introduce a way to improve the ZLL protocol's key management to solve some security issues. The method we used to improved ZLL is by using the properties of the hash chain, which will ensure the transmission security.
In the proposed redesign of ZLL, the initiator and the target devices will send each other hashed seeds to verify each other's identity when communicating with other devices.
The evaluation part of this paper concludes both in security analysis and performance analysis of using the hash function to improve ZLL.

Keywords: ZigBee Light Link, hash function, key management

I. Introduction

Smart devices, also known as Internet of Things(IoT), are growing at a rapid rate. Currently, there are 9 billion IoT devices in use and in 2021, and it is expected to reach 28 billion devices in use[7]. Zigbee is the most commonly used protocol in IoT devices as a short range wireless protocol and it is a standard based on IEEE 802.15.4. ZigBee works by creating a mesh network from each of the devices communicating with each other. Zigbee comes in small sizes and only requires low power consumption and does not need too much bandwidth to communicate with other devices.

In this paper, we describe the shortcomings and the security issues in Zigbee protocol which is used as smart lock in the network. ZigBee provides a packet integrity check function based on cyclic redundancy check (CRC), basically ensure the authenticity of the transmission. The AES-128 encryption algorithm is adopted, and each application can flexibly determine its security attributes.

There are so many threads we may face when using Zigbee, so we have ZigBee Light Link protocol which fixes some of Zigbee security issues. ZigBee Light Link uses ZLL master key to define the active network key which can fix the Zigbee issues. There is a pre-installed link key which is secret shared by all certified ZLL device to product the security issues.

However, ZLL protocol still has some security issues which we will discuss in this paper, to describe the threats and risks in ZLL network and how to improve the ZLL network. And we will introduce a method to improve the ZLL protocol.

II. Background

The most important way to improve Zigbee protocol is pre-installed key.

The ZLL pre-installed link key is the key shared by all authenticated ZLL devices. It will only be distributed to certified manufacturers and will be subject to a custody contract. Additionally, if the decryption of the APS message fails with the above key, the ZLL device will attempt to decode the APS message using the known default trust center link key. This also leads to the same fragile initial key exchange.

The security issue is that even if the manufacturer implements secure key exchange and distributes the appropriate keying material, an external attacker could use selective interference to interfere with the network connection and then wait for the insecure connection to access the exchanged keying material.

Since each ZLL device joining the ZLL network should use the ZLL master key to define the active network key, understanding the ZLL master key allows the attacker to intercept the key exchange and obtain the current active network key. This will allow an attacker to control all the devices in the ZigBee network. Since the ZLL master key is said to have been compromised on the Internet, the security of the ZLL device must be considered compromised.

In addition to the leaked keys, the ZLL device also supports a feature called "Touchlink Commissioning" that allows the device to be paired with the controller. The device may be "steal" because of the default and well-known TC link key[9].

III. Threat Model

In the previous assignment, we drew the ZigBee attack tree as Fig 1.

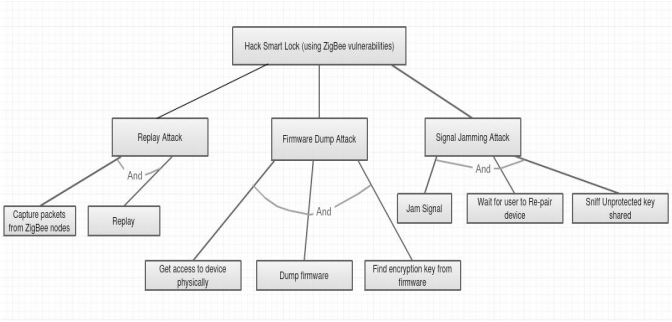

Fig 1: ZigBee attack tree

Let's look into how ZLL protocol addresses these attacks.

- Replay attack: As an improvement of ZigBee, there are counters in ALL devices. Thus, replay attack is avoided.
- Firmware Dump Attack: Once the ZLL master key is exposed, the network will be severely compromised because the network key is encrypted by ZLL master key for storage and all the communications are encrypted by network key.
- Signal Jamming Attack: There is no Trust Center or identity authentication to authenticate the devices asking for joining the network. So untrusted devices can join the network and lead information disclosure.

In addition to these attacks, ZLL encounters some other threats:

- Identity spoofing: There is no identity authentication between target and initiator. Thus, attacker can pretend to be the target and then communicate with the initiator.
- Man-in-the-Middle: As described above, it is also hard to recognize the Man-in-the-Middle attack.
- Eavesdropping: As stated earlier, once ZLL master key is exposed, the network key is not safe, and attackers can easily eavesdrop on the communications.
- According to the analysis above, we can get the ZLL attack tree as Fig 2.

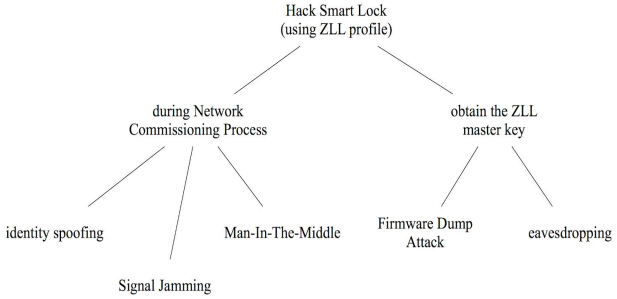

Fig 2: ZLL attack tree

In order to mitigate these attacks, we can use the hash function and hash chain to enhance ZLL key management. In the following part, we will explain how we will use hash to improve ZLL.

IV. Design

The proposed solution should improve ZLL's key management adding by identity authentication procedures in our design. Our proposed solution will have following properties

- Authenticate identity between source and target
- Improved Security against replay attack
- Improved security against Eavesdropping attack
- Improved security against Deception attack
- Improved security against Man in Middle attack
- Reduce Communication/Computing Cost

A Typical ZLL's protocol involves the following phases:

1. **Broadcast for device discovery**- Phase where initiator will broadcast scan request to start a network.

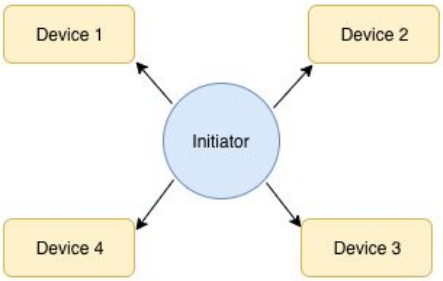

Fig 3: Broadcast

2. **Identity Authentication -** Phase where identity of source and target is authenticated.

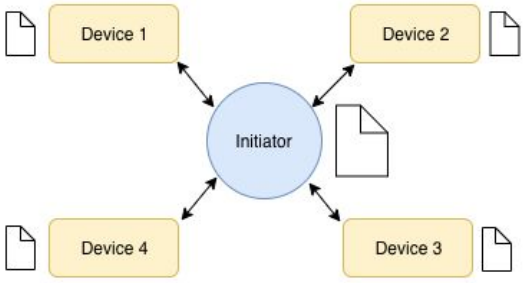

Fig 4: Flow of identity authentication

3. **Key Transmission** - Phase where initiator generated random bits and encrypt it with AES to form network key.

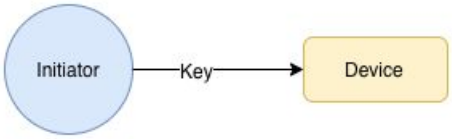

Fig 5: Flow of key transmission

V. Implementation

Liu et al proposes a new workflow for ZLL which will make ZigBee more secure and this section will explain the proposed redesign from their paper [1]. The main improvements are in 3 phases that are mentioned in the design section. We assume that the target device and the initiator both have an identical hash function.

Broadcast for device discovery - traditionally, the initiating device sends out transaction ID to the targets. The target devices reply if they wish to communicate with the initiating device. The proposed improvement by Liu et al makes the device discovery phase more secure by transmitting the ID of the device and a hashed seed. The seed that is being hashed and the number of times hash function is being run will be kept secret. Both the initiating and target devices will store the received hashed seed.

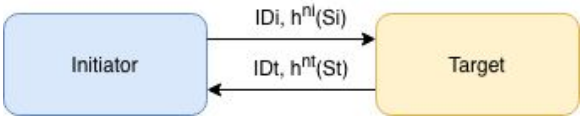

Fig 6: Step 1- device discovery phase

IDi	Identifier of the Initiator
IDt	Identifier of the Target
h	Hash Function
ni	Number of times the initiator will hash the secret seed
nt	Number of times the target will hash the secret seed
Si	Secret seed generated by the initiator
St	Secret seed generated by the Target

Table 1: variable using during device discovery phase

Identity Authentication - typical ZLL workflow does not include the identity authentication phase. Instead, typical ZLL workflow goes directly to the identification phase. Identification phase allows the user of the Zigbee network to be able to identify a target device that is added to the network. Liu et al proposes to add another phase which will allow the target device to ensure the identity of the initiating device and determine whether or not to continue with the transaction. Firstly, the initiating device will send concatenated message consisting of hashed seed(but ni-1 times this time, where ni represents the number of times the seed was hashed by the initiating device in the device discovery stage), and the timestamp that is encrypted with the network key that every ZLL device has. The target device will firstly check the timestamp to ensure that the message is fresh in case of a replay attack. Then the target device will hash the seed that was hashed n-1 times and check whether or not it matches up with the hashed seed received during the device discovery stage. This allows the target device to determine whether or not to trust this transaction and continue. The target device will reply with the same message format which allows the initiating device to confirm the authenticity of the target device as well. The initiating and target devices will update the stored hashed-seeds during the device discovery phase to the hashed seeds that they received during the identity authentication phase. Both the initiator and the target devices will store the timestamp of the initiator that was sent during this phase as it will be used during the key transmission phase.

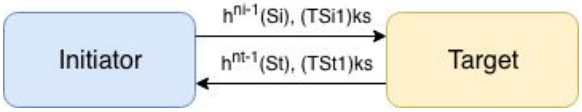

Fig 7: Step 2- identity authentication phase

TSin	N-th timestamp generated by the initiating device
TStn	N-th timestamp generated by the target device
ks	ZLL key stored in the devices. (message)ks means encrypting message with the key

Table 2: variables used during identity authentication phase

Key transmission - this phase consists of the initiating device sending out the encrypted network key which will allow the devices to decrypt the data being transmitted within the network.

The first part of the message being sent by the initiating device will consist of hashed seed, but hashed ni - 2 times this time so that the target device

can hash the received hashed seed to see if it's identical to the stored hashed seed which was hashed ni - 1 times(because a message hashed n-2 times should be equal to h(same message hash n-1 times)). Next part of the message will be the new timestamp that is encrypted with the ZLL key. In other words, the first two parts of the message will be similar to the identity authentication phase in which allows the targets to determine whether or not the source of the message is legitimate and the message is fresh.

Lastly, the initiating device will create a 32-bit long network key. The network key will then be encrypted with AES. Once the target device receives the message from the initiator, it firstly checks the authenticity and freshness of the message similar to the identity authentication phase. Then the target device can decrypt the network key to figure out the network key. The target device will then generate a response. The first part of the message will be seed that is hashed nt-2 times. Once again, this will allow the initiator to verify the authenticity of the message. Then, the target device will load up the timestamp it received from the initiator during the identity authentication phase then encrypt it with the network key that was just received. This gets added to the message.

The initiating device will firstly use the seed and timestamp to check the freshness and authenticity of the message. It will then recall the timestamp it sent to the target device during the identity authentication phase and encrypt it using the network key. If the encrypted timestamp is identical to the one that it just received from the target device, initiator will send a response notifying that there has been a successful exchange of network key. After this point, the initiator and the target can securely communicate with each other using the network key that has been shared.

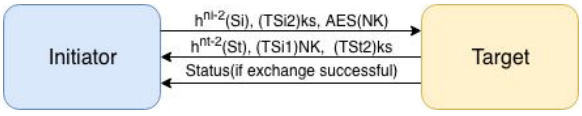

Fig 8: Step 3- key transmission phase

NK	32-bit long network key generated by the initiator
AES(NK)	Network key encrypted with AES
Status	Initiator notifying the target if the exchange was successful

Table 3: variable used in sending response notifying

VI. Evaluation

According to Liu et al [1], we evaluate improved ZLL in security aspects and compare it to the original ZLL, and we also evaluate improved ZLL in performance.

A. Security analysis

- Data integrity

In each message in the improved ZLL, there is a MAC for data validity. If the calculated MAC' is different from the received MAC, the message is identified as being tampered and the transaction is stopped. Therefore, the improved ZLL can recognize **tamper attack**.

However, the original ZLL does not recognize the tampered messages and does not guarantee data integrity.

- Identity Authentication

During device discovery phase, suppose that all the targets are trusted and all the packets are untampered. Thus, at this stage, both the initiator and the target store h^{nt}(St) and h^{ni}(Si) respectively, which will be used in the following phases. Then, during identity authentication and key transmission, each message contains St or Si encrypted by the target's partial hash chain, and the target or initiator will use the information stored earlier to identify the sender. Therefore, the improved ZLL protocol can resist the **identity forge attack**.

However, in the original ZLL protocol, the initiator verifies the target only when the target identifies itself in an application way. In key transmission phase, the initiator does not recognize the target and is vulnerable to identity spoofing.

- Data Freshness

Liu et al adds timestamps to prevent **replay attack** and **Man-in-the-Middle attack**.[1] Also, the message will have their own life cycles. Based on the timestamp, the receiver can identify whether the message is a valid or invalid message.

The ZLL protocol uses counters against replay attack but it does not recognize if the message is fresh or catch by other attackers.

- Keys Confidentiality

As mentioned earlier, during key transmission phase, the target and initiator identify each other's identity and the validity of each message. The network key is acknowledged twice, including comparison with MAC and being sent back to the

initiator. Therefore, the validity of network key is guaranteed.

In contrast, in the key transmission of the original ZLL, it is impossible to identify whether network key is valid. If the network key is tampered, the target will restart all phases. However, we check the network key in improved ZLL.

B. Performance analysis

- Communication cost

Compared to the original ZLL, the improved ZLL adds two messages during Identity Authentication and key transmission. Thus, the communication cost is only slightly increased.

- Computational cost

The improved ZLL requests much more calculations for identity authentication and data integrity. In other words, the improved ZLL sacrifices computational overhead for security.

VII. Related Solutions

There are many contributions addressing ZigBee security flaws. In this section, we will be presenting solutions that concentrate on key management of ZigBee

In [10], authors have proposed a system that combines AES algorithm with ECC (Elliptic curves cryptography) for ZigBee networks. They performed a multiple key protocol that ensures the protection of cryptographic keys. This system protects keys from disclosure but there are too many keys and exchanged packets generating a significant overhead in the network.

In [8], authors proposed a secure routing protocol that uses aggregated MAC for authentication code for ZigBee networks. Although it permits an end to end authentication, it uses only two keys, which does not ensure the secrecy of communication. In addition, these keys are not well protected against disclosure. A Man-In-The-Middle attack can gain access to the key. Also, the key is shared with every node in the network consuming large memory as the big number of keys are stored unnecessarily.

In [11], authors conducted a comparative study of Localized Encryption and Authentication Protocol (LEAP+) and current ZigBee, the experimental result showed that a distributed key management scheme such as LEAP+ provides improved security and offers good scalability. But, using a unique trust center increases cost. As it is designed to support multiple sorts of keys depending on the type of message.

In [12] authors proposed a system which improves the security of zigbee by using static keys. Which is also the weakness of the system as attacker with good extracting skills can get access to the network.

In [6], KAAZ was another system proposed based on ECC (Elliptic Curves Cryptography). This system allows the network communication to be encrypted. It utilizes small keys that are equally secure as long keys for RSA. There is a requirement to preload a custom master key.

VIII. Conclusions

With more IoT devices, our homes are more vulnerable. To combat this, protocols, such as Zigbee, is one such solution to this issue. Although Zigbee is useful for our daily life, it still has some security issues, and some scientists developed ZigBee Light Link protocol for ensure the security. However, in daily use, we can find the security issues in ZLL protocol. With a few modifications, we can improve ZLL to be more secure.

In this paper, we discussed the improved secure ZigBee Light Link protocol that aims to improve the security of ZigBee Light Link.

To make sure each process in the network which uses the ZLL protocol can be more safety than the previous one. This solution adds hash functions and hash chains for data integrity and identity authentication to prevent tamper attack and identity forge attack, even at the expense of computational overhead.

By adding timestamps, the improved protocol ensures the message is fresh and mitigates replay attack as well as Man-in-the-Middle attack. The network key will be checked before use to avoid extra communicational cost that may exist when the key is tampered. After adding those two methods, the improved ZLL can be more safety to use.

REFERENCES

[1] Liu, Ruiqing, et al. “Improved Secure ZigBee Light Link Touchlink Commissioning Protocol Design.” 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2018, doi:10.1109/waina.2018.00138. https://ieeexplore.ieee.org/document/8418123

[2] ZLL FUNDAMENTALS(UG103.9) https://www.silabs.com/documents/public/user-guides/ug103-09-fundamentals-zll.pdf

[3] ZigBee Light Link User Guide http://www.nxp.com/docs/en/user-guide/JN-UG-3091.pdf

[4] Nguyen, Son Thanh, and Chunming Rong. “ZigBee Security Using Identity-Based Cryptography.” Lecture Notes in Computer Science Autonomic and Trusted Computing, pp. 3–12., doi:10.1007/978-3-540-73547-2_3.

[5] ZigBee fundamental by Silicon Labs https://www.silabs.com/documents/public/user-guides/ug103-09-fundamentals-zll.pdf

[6] Zigbee Key management https://pdfs.semanticscholar.org/54a5/3acb4cd874fb8e2b37a09a4831155eed086b.pdf

[7] Chellappan, V., and K.m. Sivalingam. “Security and Privacy in the Internet of Things.” *Internet of Things*, 2016, pp. 183–200., doi:10.1016/b978-0-12-805395-9.00010-1.

[8] Suhas Kulkarni, Uttam Ghosh, Haribabu Pasupuleti, “Considering Security For ZigBee Protocol Using Message Authentication Code”, IEEE INDICON 2015, Pages: 1 - 6, DOI: 10.1109/INDICON.2015.7443625- https://ieeexplore.ieee.org/document/7443625

[9] Zillner, Tobias, and Sebastian Strobl. “ZIGBEE EXPLOITED - The Good, the Bad and the Ugly.” *BlackHat Presentation*, Black Hat, www.blackhat.com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly.pdf

[10] Saif Al-alak, Zuriati Ahmed, Azizol Abdullah and Shamala Subramiam, “AES and ECC Mixed for ZigBee Wireless Sensor Security “, World Academy of Science, Engineering and Technology International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, Vol:5, No:9, 2011- https://waset.org/publications/2076/aes-and-ecc-mixed-for-zigbee-wireless-sensor-security

[11] Mohammad Rezaeirad, Sahar Mazloom, Muhammad Aamir Iqbal, Dmitri Perkins, Magdy Bayoumi“ Investigating the Feasibility of LEAP+ in ZigBee Specification”, IEEE IRI 2014, August 13-15, 2014, San Francisco, California, USA DOI:10.1109/IRI.2014.7051918 https://ieeexplore.ieee.org/document/7051918

[12] A. Melaragno, D. Bandara, D. Wijesekera, "Securing the ZigBee Protocol in the Smart Grid", IEEE Computer Society, DOI: 10.1109/MC.2012.146, Page: 92 – 94, 05 April 2012, ISSN: 0018-9162 https://www.computer.org/csdl/mags/co/2012/04/mco2012040092-abs.html