Improving Zigbee Light Link with Public key cryptography

Rocco Schaffland
Technical University Munich
Munich, Germany
rocco.schaffland@tum.de

***Abstract*—The number of insecure devices connected to the internet is growing exponentialy. One of the major sources of this growth are smart home and IoT devices. People are buying these devices in huge numbers, often multiple devices at once. Installing them into theire apartments and with this opening up major IT security vulnerabilities. Even after years of development some of these technologies are still in experimental stages. Some manufacturers even knowingly risk the stability of digital infrastructure by selling insecure devices. The Zigbee standard once was pronounced to be a secure solution for this problem. Still there are flaws both in the implementation as well as design of the protocol. One major weakness of ZLL is the reliance on symmetric cryptography. Two of the major attack types on ZLL networks could have been prevented if ZLL software and hardware would have been designed and implemented with public key cryptography in mind.**

I. Introduction

The term ”Internet of things” (IoT) describes devices that are connected to local networks or the global internet with one dedicated functionality. This in stark contrast to devices like smartphones or desktop computers which implement diverse functionality. Classical examples in the consumer marked would be smart thermostats, fridges and lighting systems.

The smart home industry is thriving. In 2020 9.76 billion ”things” were connected to the internet, 2030 this number is predicted to be around 29.42 [1]. This shows the grade number of IoT device on the marked as well as its predicted growth. This surpasses the even the number of smartphones worldwide.

Those devices try to fullfill conflicting goals and meet different requirements. Many of them are concerns of usability and feasibility of the system. Those are direct factors affecting buying decisions concerning the product. The list of factors contain, but is not limited to, the price point, functionality and the convenience of a product. Those goals can be seen when comparing products representing a sector by comparing them with the technologies it tries to replace. One example for such a comparison are smart lamps and classical lighting systems. A smart system that wants to attract users has many requirements it must fulfill.

First it has to be at least as good as the product it wants to replace in the core technology it represents. In the case of smart lights this would bee the actual lighting it brings to a home as well as reliability and responsiveness. This functionality is often improved in IoT devices. In the case of smart lamps this would be represented with adjustable brightness or multicolor support.

It is also important that a user can conveniently switch to a smart system. This requires an easy setup process. For smart lamps this would require that the experience should be as close to plug and play as possible. Ideally a customer has to just screw a light bulb into a socket and start to use it.

The third factor is a reason the system has to be ”smart” and connected to the internet. For Lighting systems this is the possibility to customize the system to the wishes of the user. Part of this is the ability to remotely control the system via the internet. This can often be archived using a smartphone.

All of this needs to be done on a budget. This requires products to use simple software. Additionally This software is often implemented on cheap hardware. This adds some constraints to the system. This is often hardly compatible with the desire of a high security standard.

The sheer number of devices online call for a secure implementation of this technology. This need for security often collides with the aforementioned goals for usability. especially the need desire for low cost systems and prioritising convenience over security can cause problems.

Many different type of attacks can be conducted on, or using compromised IoT devices. Theses attacks can be clustered into 4 major groups [2]. Two of this princibles try to wrongly configure the attacked device. This can either be done by limiting or destroying the functionality. One example for destroying functionality is to turn of a smart fridge. Alternativly the functionality of a device could be used in a wrong way. An example for this would be to turn a thermosthath to the wrong temperature or open a smart door for a break in. One diffrent way of attack is the aim to extend the functionality of a system. This can be done for example by using the capability of a device to connect to the internet to capture messages in the correct frequency and spy on the attacked user. Finally there are attacks that ignore the functionality of a smart device. These attacks are very similar to classic computer network attacks. One example would be a distributed denial of service (ddos) attack. These attacks are often very effective by benefitting from the big number of device online. such attacks using bot-nets build out of IoT devices have already happend.

One protocol trying to combine both, the usability aspect as well as appropraite security measures is the Zigbee protocol.

II. Background

A. encryption and decryption

Encryption describes the procedure of encoding a messages into a cipher text only readable by authorized entities. The procedure of translating a given cipher text back to plain text is called decryption. encryption and decryption are implemented to guarantee the confidentiality of data. Today IT-security divides between two main approaches for encryption, symmetric and asymmetric cryptography.

1) Symmetric Cryptography: Communication with symmetric cryptography uses one key that is shared between every participant. This key is called the secret Key and is used for both encrypting and decrypting a message. If this key ever gets leaked all communication can be decrypted and read by everybody in possession of the key. One widely implemented and therefor thoroughly tested Algorithm for symmetric cryptography is the Advanced Encryption Standard (AES). This Algorithm is considered highly secure as well as fairly efficient.

One central Problem of Symmetric Cryptography is the mechanism used to agree on the used secret key. The initial solution for this problem would be to pick one key and share it between every peer in the conversation. This approach needs a secure way of sharing the key. Without this, the key can be extracted and is therefor useless. Default methods used for this are Public Key cryptography, Diffie Hellman Key exchange and sharing the key out of band. sharing the key out of band means on a different medium then used for communication, one example would be to manually input the key on every device. This is probably one of the most secure but least convenient methods for a key exchange.

2) Public Key Cryptography: One widely used way of ensuring the confidentiality of a message is Public Key Cryptography, also called asymmetric cryptography this approach eliminates the need of a secure key exchange between different parties. Asymmetric cryptography relies on one or more key pairs for every entity, this is in stark contrast to symmetric cryptography with one key per network or connection. This key pair consists of one public and one private key. Messages encrypted with the public key can only be decrypted with the coresponding private key and vise versa. Knowledge of the public key can not be used to generate the private key. Having two different key types therefore enables parties to freely share their public key without risking the confidentiality of encrypted messages.

public key cryptography Algorithms are typically more expensive in regard to energy consumption, key size as well as computation power. This is one reason why public key cryptography is mostly used to securely share a secret key for further use in symmetric encryption. One widely used example for a public key cryptography Algorithm is RSA.

B. authentication

Authentication is the process used to guarantee integrity of data, this is done to proof both the origin of the message as well as the fact that it hasn't been changed by a third party mid traffic.

1) Message Authentication Codes: Message authentication code (MAC) is an authentication mechanism based on a shared secret between participating actors. One version of this is the hash-based message authentication code. The core approach is to prepend the shared secret to the send message and generate a hash value for this concatenation. This value is the MAC for the corresponding message. The Mac will be appended to the original message. The recipient of the data can now also hash the secret in combination with the original message. The resulting value will be compared to the appended MAC, only if they are equivalent the authenticity is proven.

The MAC mechanism contains equivalent problems compared to symmetric cryptography, working with one shared value that must stay secret for them to work.

2) digital signatures: The concept of digital signatures relies on public key cryptography instead of one shared secret. It utilizes the properties of the key pair used for encryption and decryption. The first step of creating a digital signature is, again, hashing the original message. The generated hash value can now be encrypted with the private key. The resulting value is called signature. This signature can now be appended to the original message. Authenticity of the data can be verified by decrypting the value with the freely available public key and comparing it to the correct hash value of the received message.

This mechanism inherits the benefits of public key cryptography and enables the source of the data to uniquely sign information with its private key. Giving everybody the ability to verify the authenticity of the message.

III. The Zigbee Protocol

Based on the IEEE 802.15.4 protocol for wireless personal area networks (WPAN), Zigbee is an industry standard for low power close ranged communication.

It is developed and maintained by a group of companies called the Connectivity Standards Alliance (CSA), formerly known as Zigbee Alliance.

A. Zigbee Light Link

Zigbee Light Link (ZLL) is a protocol based on Zigbee and used for home automation of lighting systems. One popular brand of products supporting ZLL is the Philips Hue lighting series. A ZLL network consists of multiple components. The three main components of such a network are light switches, light bulbs and a bridge.

B. ZLL networks

The Bridge represents the brain and central unit of the Network. The bridge is resposible for controlling, managing the system and representing an interface to the outside. The bridge is also responsible for joining new devices to the system.

Light bulbs can either be controlled via a light switch or a smartphone. For the use with a smartphone the bridge connects with the internet forming an access point for mobile controls.

A control request send by a phone is therefore relate via the bridge to the addressed light bulb. The communication within one ZLL network is encrypted using AES and authenticated by a MAC. Using symmetric models for encryption as well as authentication is a superior to a asymmetric public key approach. The efficiency benefits prevail while in this case MAC and especially AES add sufficient security to the network.

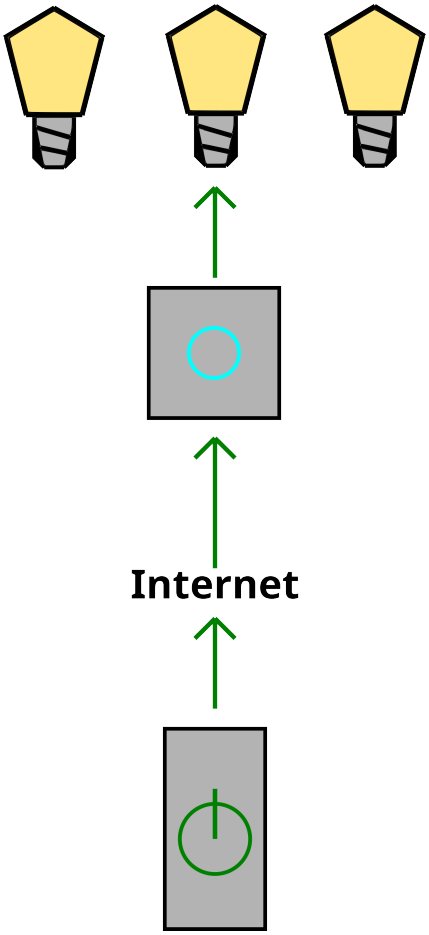

Fig. 1. typical zll network topologie

C. key exchange

bridge=hue?

All devices in one Zigbee network get assigned a secret network key issued by the bridge for encrypting and decrypting messages. The Zigbee specification describes three ways to establish this shared key [3].

1) preconfigured key: Lightbulbs, switches and a bridge can be sold as a set with the network key preinstalled on every device. The benefit of this is that there is no keyexchange that can be exploited. This is a very secure but inflexible way for establishing the secret key.

2) out of band key transfer: In this case the key is exchanged via a different medium, for example by manually configuring the secret key. It has many security benefits but may be impractical. This is especially the case in the example of a light bulb or ZLL devices in general.

3) Key establishement: Devices can join a network by exchanging the secret key with the central organ of the WPAN in question. This mechanism enables both flexibility as well as interoperability. It enables new devices to easily join a network. This mechanism is realised by the Touchlink Commission Protocol

always?

D. Touchlink Commission Protocol

The Touchlink Commission Protocol is a way to securely join a new device, e.g. a light bulb, to an already existing Zigbee network. This is initiated by the bridge sending a request to join to the device in question. To ensure this request to join a new network is from a valid source the recipient uses the strength of the signal to analyse the spacial proximity. The concrete values may change between device models. For philips hue products this lies between 45 and 75 centimeters [3].

This is supposed to prevent a unkowing neighbor from accidently hijacking an already established device. Additionally this can stop an attacker from joining a lightbulb to a malicious network. The final step of the Touchlink commission protocol is the key exchange. In this step the bridge shares the network key with the new device. The network key gets encrypted via AES using a preconfigured master key. This master key is provided only to manufactures of Zigbee certified products [4]. The master key got leaked in 2015 [4].

E. Over The Air Update

The Over the air (OTA) update mechanism enables a manufacturer to update a device after shipment. The Zigbee OTA Upgrading Cluster specification [5] defines certain security measures to be required or recommended for Zigbee certified products. Apart from image encryption and image transport it also describes image Verification. It recommends to establish a mechanism to verify the authenticity of a given update image. This can be accomplished either through a digital signature generated with public key cryptography or using a MAC. For the use of a digital signature the recipient of the image needs the public key belonging to the private key used by a manufacturer to create the signature, as well as well as the technology necessary to calculate this verification. Alternatively authenticity can be ensured via a MAC based on a secret shared between the device and the creator of the update image. Using MACs be less secure but more feasible because the devices affected certainly support symmetric encryption already.

F. implementation

Manufacturers of Zigbee certified products have to consider different aspects of an implementation for the the ZLL standard. They frequently opt for goals conflicting with the highest possible security standards.

As a leading manufacturer for Zigbee certified products Signify and its supplier Microchip implement the defacto standard used with ZLL products. Therefore this implementation can

source??

be used as an case study to analys security issues across the industry.

Signify (formerly Philips lighting) is the Manufacturer of the Philips hue product series and a former subsidery of Philips. Microchip is the owner of Atmel, the company designing and producing the SoC used on Philips hue products.

This implementation has some major security flaws.

1) Key transfer:* For key transfer, Philips opted to use the key exchange mechanism described in before including the Touchlink Commission Protocol.

2) Touchlink Commission Protocol: The Touchlink Commision Protocol is implemented in its basic form. This is with out any additional security measures to prohibit malicious or wrong join requests. This works fine for avoiding accidental hijacking from a neighbor but is easily circumvented for an attacker. With the means to forcefully get into the system and dedicated hardware It is still possible to yank a device from its corresponding network. One possible addition to the Touchlink Commission Protocol would be the need to press a button on the said device. How would that help?

3) OTA image authentication: Another decision made was to only use a MAC based verification process instead of an asymmetric model based on signing the update image.

There are many different reasons like ease of use, production cost or technological constraints to opt for one implementation over the other. These are just three cases describing the defacto ZLL standard implementation exposing home networks to additional risk.

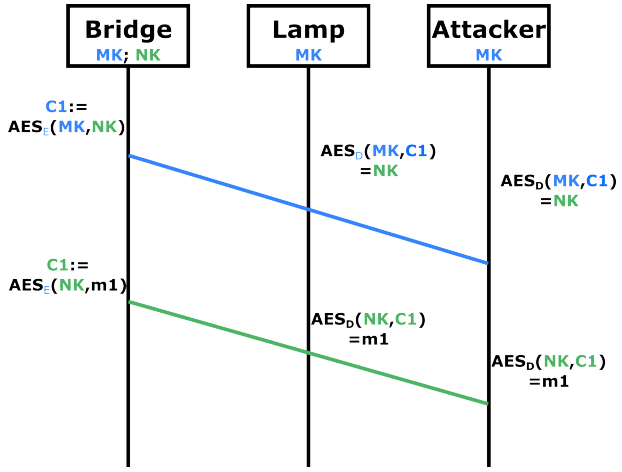

Fig. 2. Chronology of a key extraction attack.

IV. Attacks

Design as well as implementation decisions enable several different attacks on ZLL networks. In the following we will describe two major vulnerabilities and attacks discovered implemented and published in several papers.

The first vulnerability enables the malicious actor to extract the secret network key used for encryption. This attack is performed by eavesdropping on the key exchange. As a result the intruder can capture information from the network. The second attack described below is based on a weakness in the Touchlink commission protocol. This weakness can be used by an aggressor to hijack individual devices and link them to a new, malicious, network. This can further be used to install malicious firmware. This vulnerability got abused in an experimental setting creating a network worm. This worm had the ability to infect light bulbs and spread between lamps in one network as well as between multiple networks [3].

A. extracting the network key

One Problem arising from the implementation decisions made by Signify and other manufacturers is the possibility to spy on the key exchange and extracting the secret key. This is enabled by the design of the Touchlink comission protocol relying only on signal strength to prove proximity. This is used in combination with the insecure decision to encrypt the network key with a secret master key. Figure 1 shows the course of events of this attack. The proximity check is omitted for reasons of simplicity. The graph begins with the exchange of the Network key. MK represents the master key and NK the network key.

1) Sharing the network key: As mentioned above a light bulb that wants to join a new Zigbee network needs access to the corresponding network key shared by the bridge. The Network key is encrypted using AES with the preconfigured master key. In Figure 1, this encrypted essage is C1. This message containing an encrypted version of the secret key is send to the new device. This message can be captured by a malicious actor. With the master key available, everybody can decrypt the cipher and extract the secret network key.

2) using the network key: Messages send after establishment of the network key will be encrypted using this key. In figure 1, m1 represents arbitrary messages send after establishing the network key. C1 is the cipher generated by encrypting m1 using the NK. After gaining access to the network key used, an attacker can decrypt messages send between the corresponding devices in the network.

3) Trigger the join event: For this attack to work the attacker needs to observe a key exchange. This requires a new device joining the network. Alternatively an attacker has the option to jam the connection between existing elements of the system, this can cause the owner or administrator to re pair the device.

B. hijacking lightbulbs

A Different vulnerability is embedded into the Touchlink commission protocol. The only validation happening is based on signal strength. This implies that not just anyone in close proximity can yank a device from its linked network but also everybody with the means to pretend or fake this proximity. Faking this proximity typically needs dedicated hardware with a very high signal strength.

Hijacking a light bulb is a prerequisite for installing new firmware. This attack uses fake firmware images to overwrite the programming of a device. The need for dedicated hardware restricts the use of this vulnerability to a rather limited scale.

”IoT Goes Nuclear: Creating a ZigBee Chain Reaction” is a Paper documenting a bug in Atmels Zigbee stack was published in 2016 [2]. This bug circumvented the proximity check all together and therefore eradicated the need for dedicated hardware. Using this bug, the hardware used on a Philips hue light bulb could be used to perform the attack. Today this bug is fixed

C. The infectious worm scenario

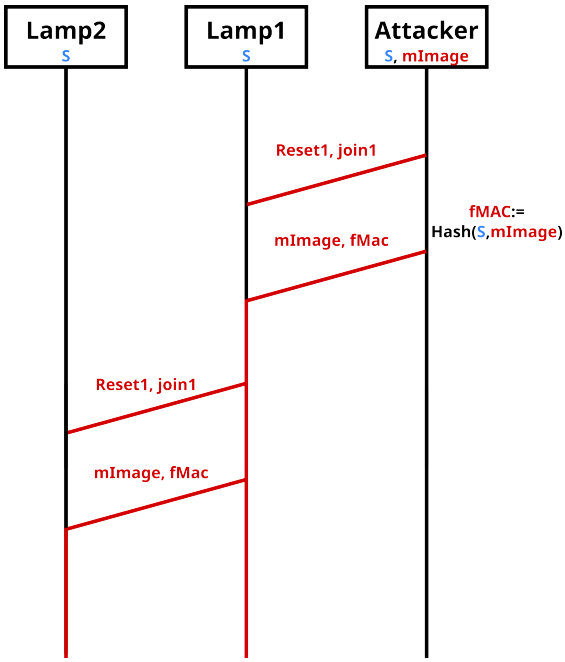

Fig. 3. Chronology of a worm attack.

One dangerous and wide-ranging attack enabled by circumvention of the proximity check is a Computer worm. This attack describes a virus with the ability to spread between computers. Attacks like this are known for a long time on traditional computer networks. In combination with the widespread adoption of IoT devices those attacks increase in potential.

One example of a spreading ZLL worm was created and published as part of the aforementioned paper [2]. This implementation was based on the novel bug in the Touchlink Commission Protocol. This bug enabled the use of a compromised Philips Hue smart light as an initiator of a hijacking attack on other ZLL devices in a distance up to 100m . In combination with the OTA update mechanism this was used to create a worm capable of spreading between networks as well as infecting other, spatially close, ZLL networks. Figure 3. shows the course of events corresponding to this attack. S is the secret necessary to generate the MAC used for authenticating an OTA update image.

1) factory reset and repairing the ligh bulb: The first step of this attack abuses the lackluster implementation of the Touchlink commission protocol on ZLL devices.

The process of hijacking a light bulb already connected to a different network consists of resetting the lamp to factory new and then reconnecting it to a new malicious network. This can be done using two messages build in to the bridge for a legitimate repairing of lamps. An attacker would need to first send a reset to factory new request, resetting the targeted lamp, removing all stored information related to its former WPAN. After resetting the lamp it is possible to send a join request for the new, compromised network. This is were the bug circumventing the proximity check is used to pair the Lamp to the malicious network. These two messages are represented by reset1 and join1 in Figure 3.

2) OTA update: To infect the yanked device the OTA upgrade protocol is used to overwrite the firmware. This can be used to transform the affected device into a malicious actor, deactivating the OTA feature so it can’t be repaired or just brick the lamp entirely.

One factor facilitating this is the use of MACs as the authentication method. For authenticating a malicious update image only the shared secret is needed. This secret can be extracted from a ZLL device. All lamps of the same product series share the secret [3]. This way it is sufficient to gain access to the secret stored on one device belonging to a given product series to validate an update image that will be accepted by all devices of belonging to this group. Reproducing this secret can be done by a power analysis attack. An sidechannel attack abusing the simple hardware used for many IoT devices. In Figure 3, this malicious image is represented by mImage. The faked MAC by fMac. This image is send together with the generated MAC to the yanked light bulb.

Due to the correct S used for generating the MAC Lamp1 will falsely verify and install the image. After the installation of the malicious firmware Lamp0 will turn into a malicious actor and start spreading the faulty image.

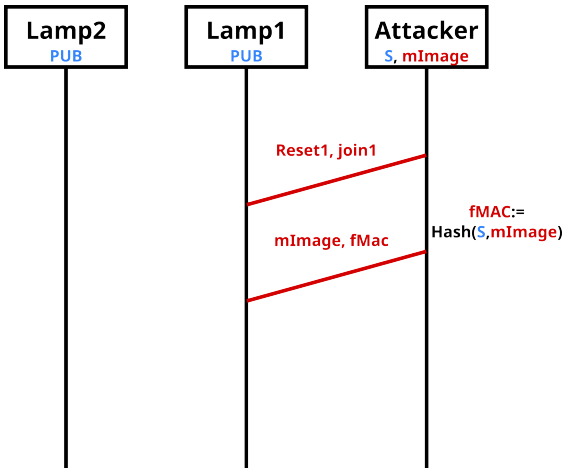

Fig. 4. typical zll network topologie

3) worm:

V. PUBLIC KEY CRYPTOGRAPHY IN ZLL

A. Symmetric cryptograpyh

Lamp2 in Figure 3 represents an arbitrary lamp in close proximity to an infected device. The attack will be repeated as described before. First Lamp1 will yank Lamp2 from its associated network and joint it to the malicious network. This is done by sending a factory reset (reset2) and join (join2) request to Lamp2. The next step is sending the same, malicious update image (mImage) and the corresponding MAC (fMAC). Again Lamp2 will verify and install the image. This process will repeat and infect more devices. The vulnerabilities enabling the attacks described in the previous chapter are enabled by three factors.

1) shared secret key: The central weakness for all symmetric cryptography is the reliance on a secret, shared between all associated parties. This implies that the sensitive data has to be stored on all devices, also including the less secure IoT systems. Using symmetric approaches forces the security to be defined by the weakest link. It is sufficient to retrieve the key material from any device to break encryption as well as authentication mechanisms.

2) preshared secret key: The need for a shared secret key is esspecialy problematic if this key is the same for multiple devices of a series or manufacturer. This is the case for both, the master key used for encrypting the network key as well as the secret used for generating and verifying the MAC. The master key is the same on every ZLL device. This is done for better interoperability. Using the same master key on every certified ZLL product enables products from different manufactures to share network key with each other. The secret corresponding to the MAC is the same on devices of the same product series. This allows to only create and authenticate one update image for a product instead of individually authenticating one image for every device.

The disadvantage of sharing a secret between so many devices is the increased attack surface. Gaining access to one device and the corresponding key material is sufficient to build an attack for many ZLL devices at the same time.

3) simple hardware: For practical reasons as well as production costs IoT devices are designed to use simple chips without complicated security measures. This facilitates attacks that try to extract key material.

B. public key cryptography use cases

For adoption of new technologie like IoT devices a low price point is a relevant factor. Therefore simple, fairly cheap and insecure chips will be part of the market. Using different keys for every device on the marked does not just hinder interoperability but is also unfeasible. This is due to the big quantities of devices sold. Public key cryptography is a solution to all three of the core problems described.

Public key cryptography has no need for a shared secret and can securely be implemented without a huge amount of keys or restricting the interoperability of ZLL products. The two attacks described, can each be mitigated in a different way by using pubic key cryptography

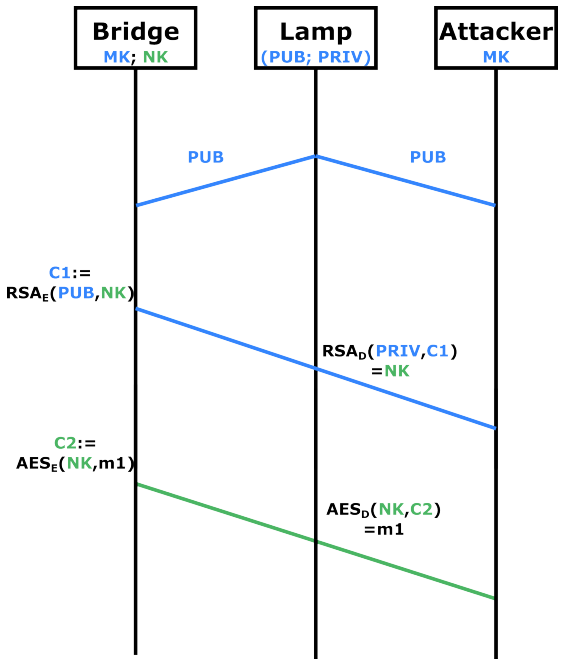

Fig. 5. Prevented key extraction attack

1) encryption for key exchange: The underlying weakness used in key extraction attack is the leaked master key used to encrypt the network key. Using public key encryption for this step can prevent the attack and further eavesdropping on messages within the network.

For this, one key pair consisting of public key and private key has to be installed onto every ZLL device. This key pair can replace the master key and has to differ between every device. Guessing a key pair from an already retrieved key pair must not be possible. This key pair can be preconfigured when manufacturing the device.

The key exchange between the bridge and a lightbulb joining the system can be encrypted using the public key corresponding to the light bulb. Encrypting the network key with the lamps public key ensures, only the lamp that message was addressed to can decrypt the network key.

For this to work the public key must be shared with the bridge. This can be done by just sending the public key as part of the join request. Capturing this exchange does not empower the attack. This is due to the nature of public key cryptography. Exchanging the public key this way ensures the interoperability of ZLL products.

Recovering the private key stored on one device would enable an attacker to compromise communication including the affected device. Knowledge about this key does not facilitated the attacker to gain access to other private keys used in other devices. This way an attacker can only carry out a key extraction attack if he has physical access to the attacked device. This is unfeasible. One additional attack surface would arise if all installed key pairs were logged to a list and this list leaked to the public. Logging key pairs must not happen in a retrievable way.

Figure 4 shows a chronology of an prevented attack. In the begining The bridge only knows the network key NK while the Lamp has only its key pair consisting of public key (pub) and private key (priv) installed. This depicts the events after the inital join request from the Bridge to the Lamp. First the Lamp shares its public key. This might be unencrypted and can be read by both the Bridge as well as the attacker. Next the Bridge encrypts the network key (NK) it wants to share with the joining lamp. This encryption is done using RSA with the public key received from the lamp. This cipher (C1) is send to the Lamp, it can be decrypted at its destination. The attacker can not decrypt the cipher. Further messages can be encrypted with the secret network key (NK)

2) authentication for OTA: Digital signatures are generally more secure then MAC authentication. This is not implemented with most ZLL devices today. Using public key cryptography for signing the OTA images would eliminate the need for shared secret and replace it with a key pair consisting of public key and private key. Only the public key is installed on the consumer device. The private key is only stored on secure machines used by the manufacturers for signing the update images.

With public key cryptography an OTA image would be signed with the private key only known to the manufacturer generating the image. The authenticity of this image can then be verified by the addressed devices. Gaining access to the used public key device does not jeopardize the security of this single device or the system as a whole. This enables the manufacturer to still use one key pair for all devices belonging to a given group.

C. feasability

Public key cryptography can, as mentioned, improve a products security. But the implementation of public key cryptography encounters some challenges. Especially in the combination with ZLL devices in particular and IoT in general, this can pose a problem. Trying to get this technology to cheap, small and low power devices has some difficulties. Besides the added security feature this can degrade the user experience or be completely impossible under given circumstances. Here are three of the classical questions about feasibility that have to be answered before considering an implementation of public key cryptography on ZLL devices.

1) latency: A common concern regarding public key cryptography is the latency arising from the additional computing effort needed in comparison to symmetric methods. In the case of a zigbee implementation as proposed this is not a big problem. This is due to the fact that the public key cryptography is used very rarely and only on occasions were the responsiveness can be neglected i.e. pairing a new device and installing a firmware update. For this the computation of public key cryptography is fast enough.

2) power consumption: All IoT devices try to maintain a small power consumption footprint. This is done for multiple reasons. Many devices are always online and responsive. Some of them have limited power at their disposal. Most ZLL devices e.g. light bulbs and the bridge are always connected to a power source. This, as well as the rare use of public key cryptography, limits the additional power usage to an negligible amount. Only for some devices this can be a problem. One example are light switches. Many switches work only with the energy generated by pressing the buttons. This of cause could lead to problems when using a more power intensive scheme like RSA.

3) productioncost: One of the main factor for the acceptance of smart devices in the public is the acquisition cost of the products available. This directly drives manufacturers to lower production costs to uphold certain profit margins. Today this is one of the main reasons hindering the switch to public key cryptography.

D. Chip marked

The Microchip ATmega2564RFR2 (formarly Atmel ATmega2564RFR2) is one of the chips used in a verity of Philips Hue products manufactured by Signify. This SoC is built for use with the ZLL protocol. The chip incorporates an AES hardware accelerator for efficient computation of cryptographic operations using the AES scheme. It does not contain a comparable component for RSA or other public key cryptograpyh methods. This is the case for virtually every board within the field of commercial ZLL products. This is due to a obstructive cycle where chips not implementing RSA are the industrie standard. This in turn makes it harder to adopt RSA into a product and therefore reinforces purely symmetric devices as the sole industry standard.

Adding hardware acceleration for a RSA, can be done in three ways.

1) Use a Chip on the market: As described above, the marked for ZLL capable SoCs using RSA hardware acelleration is small. Chips meeting these criteria are considerably more expansive and often only available in smaller quantities.

2) Adding a component: The simplest Idea is to extend a SoC already used for ZLL, e.g. Atmel ATmega2564RFR2, and expand it with an additional component. This way more widely available dedicated hardware acceleration chips could be used to add the desired functionality to the system. This approach adds new complexity to the system and the manufacturing process. The existing SoC must be changed to work with the additional hardware. Further this would add an additional step in production, costs and increase the size of the device.

3) Designing a dedicated chip: The final approach is the redesign of the SoC, adding the hardware acceleration directly to the used chip. This has the benefit that the new unit can be exactly tailored to the needs of the corresponding device. Also this might lower the cost per unit over time in comparison to other approaches. However the expanses required for researching and designing the new SoC might be substantial.

VI. Evaluation and Discussion

Public key cryptography could improve the security of the ZLL standard. To some extent, this is even recommended by the Zigbee specification. But it is not implemented today.

A. improvements

As described before, public key cryptography can improve a ZLL system by preventing some of the major attacks used on the system. Additionally the implementation does not degrade the user experience. The increased power usage and latency is not tangible. On the surface the systems using symmetric and public key cryptography can not be distinguished. This is contrary to many other approaches like out of band key transfare.

we prob still need that !

B. Problems

While the implementation of public key cryptography seems to be a good thing, there are also good reasons for going solely with semetric cryptograpyh.

1) production cost: One fundamental argument against implementing public key cryptography is the increased production cost of such a system. A Increase in production cost leads to an increased cost for the end consumer. Especially in the beginning of such a technology this can be detrimental to its success and sale numbers. This could also lead consumers to choose cheaper and even more insecure alternatives.

2) backwards compatibility: A significant reason against introducing public key cryptography today is the the need for backwards compatibility. Implementing such a major change to the security system of ZLL certified products will divide the marked into camps. This would destroy the interoperability not just between different manufacturers but even different generations of products within the same brand. This is one of the main reasons the implementation of public key cryptography is almost impossible for ZLL products today.

3) its not perfect and not needed: The ZLL implementation described using only symmetric aproaches has flaws and serious security issues as well as visible vulnerabilities embedded into its desing. But it has to mentioned that the worst attacks like the worm attack were be fixed by finding a bug within the implementation rather then design. This way the problem got corrected without redesigning the whole system. Public key cryptography on the other hand can mitigate some of the problems arising and could have stopped some attacks from happening, but it is still not perfect. Some vulnerabilities are woven so deep into the protocol that even redesigning chips using RSA will not fix them. One of this issues is the described possibility of yanking a lamp from the system by using dedicated hardware.

no proper reason

VII. Related Work

Today very little work has been published on public key cryptography in Consumer Zigbee products. One of the central papers for ZLL security is ”ZIGBEE EXPLOITED: The good, the bad and the ugly” by Tobias Zillner [4]. This paper does not mention public key cryptography. RSA in industrial grade Zigbee products is discussed by Hamza Kadhum in ”Enhancing Zigbee Security for Industrial Implementation” [**?**]. This does not deal with ZLL.

VIII. Conclusion

The amount of IoT devices on the marked and connected to the internet is immense. This number is growing rapidly. The sheer number of devices online makes attacks against devices using Zigbee in general or ZLL in particular interesting and very effective. Especially self-replicating worms will be a grade danger to interconnected infrastructure in the future. Therefore high security standards for smart devices are needed. One protocol trying to archive this while also presenting a good user experience is the popular Zigbee protocol. Zigbee Light Link (ZLL) is a version of Zigbee used for smart lighting systems. It is supposed to bring a secure implementation of a smart lamps into consumer homes. With the industry standard implementation of ZLL this is not the case. The decision to only use symmetric cryptography both for encryption as well as authentication on ZLL devices opens up many vulnerabilites. This is facilitated by the simple and rather cheap hardware used. There are two major attacks that can be carried out to a ZLL network that could be mitigated with public key cryptography.

The first attack is a key extraction attack where by a leaked master key is used to retrieve the network key used for further communication. The attack works because the encryption of the network key exchanged is done with the same master key on every device. Public key cryptography can prevent this attack from happening. This can be done by encrypting the network key using a public key with a corresponding private key only known to the device receiving the message. For this to work every device must have a individual key pair installed. The public key used for encryption can be shared in the join request.

Another, more dangerous attack enabled by symmetric cryptography is the installation of malicious firmware via an update image. This attacks abuses the symmetric MAC authentication mechanism used for most implementations. This attack consists of two steps. retrieving the secret used for generating and verifying the MAC and using this MAC to authenticate a hostile update image. Extracting the secret can be done using a power analysis attack on a chip used in the targeted product. All devices belonging to the same product series share the same secret. public key cryptography can metigate this attack by using digital signatures instead of MACs. This can prevent the attack by only installing the public key needed for verification on the vulnerable smart device while storing the private key only on secure systems owned by the manufacturers. This is the recommended approach mentioned int the the Zigbee Upgrade cluster specification [5]. There are reasons against using public key cryptography on ZLL devices. The major reasons are production costs and backward compatibility. Also it can be argued that public key cryptography is still not prefect and leaves vulnerabilities to be improved. There are many alternative concepts in the space of cyber security including the diffie hellman method replacing the classical key exchange. There are also new protocols on the marked trying to replace Zigbee. One of them

is even developed by the group maintaining ZLL. Today the implementation of public key cryptography on ZLL certified products is way out of reach. Still this shows again how certain aspects of Informational security have to be planed ahead.

REFERENCES

[1] https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[2] E. Ronen and A. Shamir, "Extended Functionality Attacks on IoT Devices: The Case of Smart Lights," 2016 IEEE European Symposium on Security and Privacy (EuroSP), Saarbruecken, Germany, 2016, pp. 3-12, doi: 10.1109/EuroSP.2016.13.

[3] E. Ronen, A. Shamir, A. -O. Weingarten and C. O'Flynn, "IoT Goes Nuclear: Creating a ZigBee Chain Reaction," 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2017, pp. 195-212, doi: 10.1109/SP.2017.14.

[4] T. Zillner, "Zigbee exploited - the good, the bad and the ugly," in Black Hat USA, 2015.

[5] ZigBee Document 095264r18; ZigBee Over-the-Air Upgrading Cluster