2nd International Conference on Recent Innovations in Science, Engineering and Management
JNU Convention Center, Jawaharlal Nehru University, New Delhi (ICRISEM-15)
22 November 2015 www.conferenceworld.in ISBN: 978-81-931039-9-9

SECURITY AND PRIVACY ISSUES OF INTERNET OF THINGS: CHALLENGES AND THREATS

Shalabh Agarwal[1], Sayantan Majumdar[2], Abhisek Maiti[3], Asoke Nath[4]

[1] Associate Professor and H.O.D, [2,3]PG Scholar, [4]Associate Professor,
Department of Computer Science, St. Xavier's College Kolkata, (India)

ABSTRACT

According to the latest research of emerging technologies in the next few decades, an era of fully integrated Future Internet is coming. The Internet of Things (IoT) paradigm will be the one of the leading technologies in the transformation from nowadays Internet into the Future Internet. IoT market has already begun to take off rapidly. Consumers can buy connected versions of nearly every household appliance available. However, despite its increasing acceptance by consumers, recent studies of IoT devices seem to agree that "security" is not a word that gets associated with this category of devices, leaving consumers potentially exposed. A couple of security issues on a single device such as a smart phone can quickly turn to multiple issues when considering several IoT devices in an interconnected home or business. In light of the importance of what IoT devices have access to, it's important to understand their security risk.

Keywords: IoT, security, cyber-attack, authentication, Denial of Service

I. INTRODUCTION

Recent Gartner research [1] predicts that there will be 25 billion connected IoT devices in consumer smart home environments by 2020. These connected devices could provide a much larger surface for attackers to target home networks.

Currently, most proposed IoT attacks are proof-of-concepts and have yet to generate any profit for attackers.This does not imply that attackers won't target IoT devices in future, even if it is just to misuse the technology or have a persistent anchor in a home network.

All of the potential weaknesses that could afflict IoT systems, such as authentication and traffic encryption, are already well known to the security industry, but despite this, known mitigation techniques are often neglected on these devices. IoT vendors need to do a better job on security before their devices become ubiquitous in every home, leaving millions of people at risk of cyberattacks.

The IP Security protocols already take care of many of the concerns. But the nature of IoT devices and IoT architecture creates its own challenges in securing every IoT Solution.

II. SECURITY CHALLENGES AND THREATS FOR IOT

The use of weak passwords is a security issue that has repeatedly been seen in IoT devices. These devices often do not have a keyboard, so configuration has to be done remotely. Unfortunately, not all vendors force the user

to change the device's' default passwords and many have unnecessary restrictions which make the implementation of long, complex passwords impossible.

The Open Web Application Security Project's (OWASP) [2] List of Top Ten Internet of Things Vulnerabilities sums up most of the concerns and attack vectors surrounding this category of devices:

- Insecure web interface
- Insufficient authentication/authorization
- Insecure network services
- Lack of transport encryption
- Privacy concerns
- Insecure cloud interface
- Insecure mobile interface
- Insufficient security configurability
- Insecure software/firmware
- Poor physical security

III. ATTACK SURFACE AND THREAT SCENARIOS

Attackers can intercept or change the behavior of IoT devices in many ways. Some methods require physical access to the device, making an attack more difficult to conduct. Other attacks can be carried out over the internet from a remote location. The following sections list the different attack scenarios based on the access level that the attacker may have.

3.1 Physical Access

An attacker can gain the highest level of access to the smart home device if they get physical access to it. Although this might seem like an improbable attack vector, it is still a plausible threat. For example a friend of a friend can gain access to his/her friend's network and gain access to sensitive information. Another plausible physical access attack scenario takes advantage of the market for second-hand IoT devices. Some users might buy a used device off the internet in order to save some money, but could end up with a device that has been compromised to spy on people.

3.2 Local Attacks Over the WiFi/Internet

An attacker with access to the local home network, either wirelessly or through an Ethernet connection, is able to perform various attacks against smart home devices. There are generally two common modes of for smart home devices:

3.1.1 Cloud Polling: In this case, the smart home device is in constant communication with the cloud.The device checks the cloud server to see if there are any commands to be executed and then uploads its current status. This implementation is susceptible to Man in the Middle Attacks(MITM).

3.1.2 Direct Connection: Some devices use direct connections to communicate with a hub or application in the same network. For example, a mobile app may be able to scan the local network for new devices and locate them by probing every IP address for a specific port. Another method is to use the Simple Service Discovery

Protocol/Universal Plug and Play (SSDP/UPNP) protocol to discover the devices. This means that any attacker could do the same to easily find these devices.

OWASP's List[] of the Top Ten Web Vulnerabilities includes:

- Infection flaws
- Broken authentication
- Cross-site scripting (XSS)
- Insecure direct object references
- Security misconfiguration
- Sensitive data exposure
- Missing function-level access control
- Cross-site request forgery (CSRF)
- Use of components with known vulnerabilities
- Invalidated redirects and forwards

3.1.3 Cloud Infrastructure Attacks: A smart home device may include a back-end cloud service, depending on the category of the device. Many IoT devices offer cloud services.The cloud infrastructures are prone to MITM attacks.

3.1.4 Malware: Malicious software installed on any device connected to the home network could have the ability to interact with smart home devices and let the attacker perform the attacks as previously described. Most likely, a compromised smartphone or computer could be used to attack other devices.

Attacks on Wireless Sensor Network(WSN): WSNs are used to monitor physical or environmental conditions like temperature, sound, pressure etc. and to co-operatively pass their data through the network to a main location. Most of the WSN's routing protocols are easy and straightforward and hence are vulnerable to attacks like Denial of Service (DOS) [3]and Distributed DOS attacks.

IV. MITIGATION

Unfortunately, it is difficult for a user to secure their IoT devices themselves, as most devices do not provide a secure mode of operation. Nonetheless, users should adhere to the following advice to ensure that they reduce the risk of these attacks:

- Use strong passwords for device accounts and Wi-Fi networks
- Change default passwords
- Use a stronger encryption method when setting up Wi-Fi networks such as WPA2
- Disable or protect remote access to IoT devices when not needed
- Use wired connections instead of wireless where possible
- Be careful when buying used IoT devices, as they could have been tampered with
- Research the vendor's device security measures
- Modify the privacy and security settings of the device to your needs
- Disable features that are not being used
- Install updates when they become available
- Use devices on separate home network when possible

- Ensure that an outage, for example due to jamming or a network failure, does not result in an insecure state of the installation
- Verify if the smart features are really required or if a normal device would be sufficient Manufacturers of smart home devices should ensure that they implement basic security standards at the very least:
- Use SSL/TLS-encrypted connections for communication
- Mutually check the SSL certificate and the certificate revocation list
- Use strong firewall for the network access
- Deep packet inspection should be enabled
- Allow and encourage the use of strong passwords
- Require the user to change default passwords
- Do not use hard-coded passwords
- Provide a simple and secure update process with a chain of trust
- Provide a standalone option that works without internet and cloud connections
- Prevent brute-force attacks at the login stage through account lockout measures
- Secure any web interface and API from bugs listed in the OWASP List of Top Ten Web vulnerabilities
- Implement a smart fail-safe mechanism when connection or power is lost or jammed
- Where possible, lock the devices down to prevent attacks from succeeding
- Remove unused tools and use whitelisting to only allow trusted applications to run
- Use secure boot chain to verify all software that is executed on the device
- Where applicable, security analytics features should be provided in the device management strategy

V. SECURITY MEASURE

In case of IoT devices, mainly there are two potential points of attack.

- Attack during data transfer
- Unauthorized access to the device and sensitive information stored in it

5.1 Securing Data Transfer

The data in a public network passes through a number of untrusted intermediate points. Therefore the secure data must be scrambled in such a way that the data will be useless or unintelligible for anyone who is having unauthorized access to the secure data. This can be achieved with the help of cryptographic methods explained in the following sections.

5.1.1 Data Encryption: Encryption is the process of scrambling/encrypting any amount of data using a (secret) key so that only the recipient, who is having access to the key, will be able to descramble/decrypt the data. The algorithm used for the encryption can be any publicly available algorithm like DES[4], AES[5] or any algorithm proprietary to the device manufacturer. The key is known only between the communicating devices and is typically 100s of bits in length. If publicly available algorithms are used, the security of the transferred data totally depends on the secrecy of the keys used for the encryption. Sharing and maintaining the secret key between the communicating devices without any unauthorized entity getting access to the keys is important for

foolproof secure data communication. These keys can be embedded in the device prior to the communication, exchanged offline in a secure manner or established online using any key agreement algorithm

5.1.2 Public-Key Key Agreement Algorithm: Key agreement algorithm is a public-key cryptography algorithm. For devices that use Key agreement algorithm will have a private-key and an associated public-key. The private-key is generally a random number of hundreds or few thousands of bits and the public-keys are derived from the private-key using the one-way hash function specified by the key agreement algorithm. One-way functions are mathematical function that follows the Secured Hash Standard(SHS)[6] and in which the forward operation can be done easily but the reverse operation is so difficult that it is practically impossible. The public-key is derived using private-key on the forward operation of the one-way function making the reverse operation of obtaining the private-key from the public-key practically impossible.

For e.g. let P be the private-key of a device and U(P, C) to be the public-key of the device, the representation U(P, C) is to show that the public-key of a device is derived from the private-key P of that device and some shared constants C known by all the device taking part in the communication.

Consider two devices A and B. Let P_A and U_A (P_A , C) are the private-key and public-key of device A, and P_B and U_B (P_B , C) are the private-key and public-key of device B respectively. Both device exchanges their public-keys.

Device A, having got the public-key of B, calculates key, K_A = Generate_Key(P_A , U_B (P_B , C))

Device B, having got the public-key of A, calculates key, K_B = Generate_Key(P_B , U_A (P_A , C))

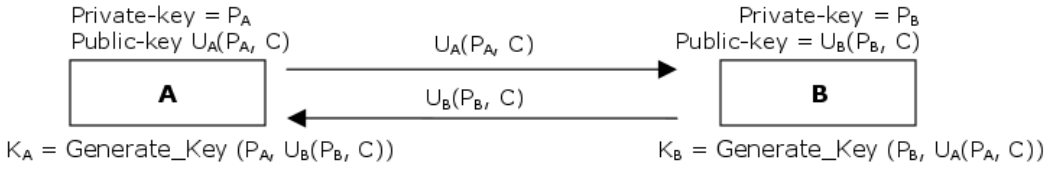

Fig. 1: Key Agreement

The key generation algorithm 'Generate_Key' will be such that the generated keys at the device A and B will be the same, that is shared secret KA=KB=K(PA, PB, C). It is impossible for any middleman, who only have access to the public-keys, UA(PA, C) and UB(PB, C), to obtain the shared secret K unless he gains access to the private-key, PA or PB of any of the communicating device. Examples of key generation algorithms are DH[7] and ECDH[7].

5.1.3 Digital Signature: The device in a network may be communicating with the unknown or less familiar device. The communication may also require routing through many intermediate points. During Key Agreement process, for establishing a secret key, any middlemen can substitute a device's public-key to its public-key and thus results in establishing a shared secret with the device. Therefore, for establishing shared secret using the key agreement algorithm, it is important for device to receive an authenticated public-key from the peer. For authenticated exchange of public-key, Digital Signature is used.

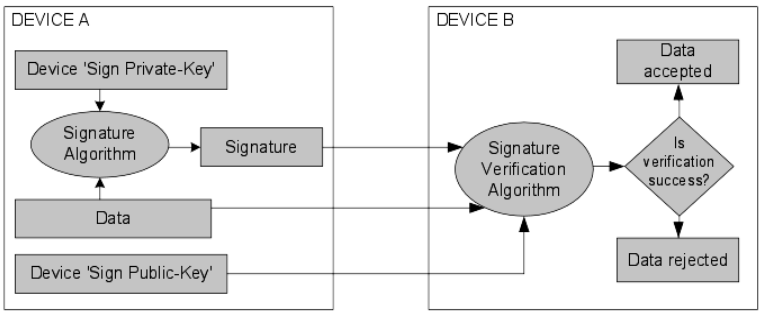

Fig. 2: Digital Signature Scheme

A device signs the message using a signatures algorithm with its sign private-key to generate a signature and any device that has got the access to the sign public-key of the signed device can verify the data with the signature using the signature verification algorithm. If any third party modifies the data or signature, the verification fails. Since only the signed device knows its sign private-key, it will be impossible for any other device to forge the signature. Examples of Digital Signature algorithms are RSA[8][9], DSA[9][10], ECDSA[9][10].

5.1.4 Digital Certificate: Even while using the digital signature algorithm, the 'sign public-key' from a peer device has to be obtained by an authenticated way to ensure the authenticity of a received message. For key agreement or digital signature the authenticated transfer of public-key in a large network is difficult or even not possible without a centralized trusted authority. This centralized authority is trusted by all the devices in the network. This authority is generally known as trusted Certificate Authority or CA. The Certificate Authority (CA) signs the public-keys of devices along with the device ID using the CA's private-key to generate the signature. These CA signed data of a device (public-key, IDs etc.) along with the signature arranged in a standard format is called as a certificate. The certificate is issued by CA to all devices taking part in the communication. Any device, having the CA's public-key installed, can verify the authenticity of the received certificate and thus the public-key of the peer device. One popular certificate format is X.509[9][11].

For obtaining the certificate, a device requests the certificate to the CA. The device sends its public-key, unique IDs and other information as required by the CA. The CA usually does some background check to ensure the device is not hostile before issuing the certificate. The certificate obtained by a device is rarely changed and the process of obtaining a digital certificate for an embedded device is usually done offline.

Once communicating devices obtains its certificate from the CA, the devices exchange their respective certificate before establishing a shared secret between them. Any device on receiving the peer certificate verifies it using the CA's public-key. Since the CA public-key common to all the devices taking part in the communication and is never changed, it is pre installed on all the devices in network generally through any offline trusted methods. Once a peer authenticates the device certificate, the device can use the public-key in the certificate to sign any message sent by the device to the peer to prove the message's authenticity.

As the number of devices taking part in the communication increases and the location of these devices is distributed over different parts of the world, a single certificate authority may not suffice to issue and maintain certificates for all the devices. Certificate Hierarchy is the solution here.

In Certificate Hierarchy there will be a Trusted Root CA who will give permission other CA to issue certificate to the communicating devices. The Root CA will issue the certificate for the respective intermediate certificate

authority. The intermediate CA will issue certificates to the device. In addition to issuing certificates to the devices, the intermediate CA will also give its certificate, issued by root CA, to the devices. There can be multiple level of certificate hierarchy in which the intermediate CA's will give permission to other CA to issue certificate to the communicating devices. During a secure communication a device may ask all the intermediate certificates from its peer for successful verification of the received device certificate.

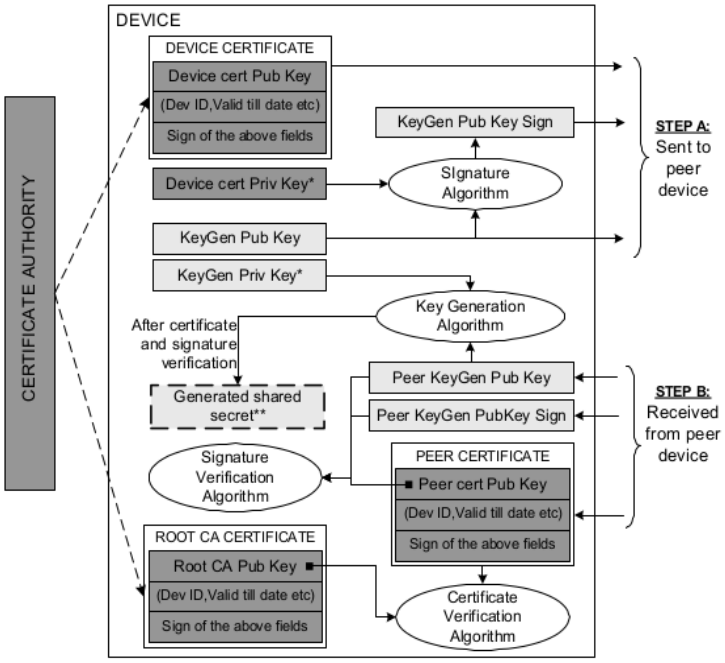

Fig. 3: Authenticity Verification with the Help of Digital Certificate

5.2 Prevention of Unauthorized Access

We must ensure that, secret key is known between the communicating devices. It must also be ensured that the private-key is not disclosed from the device. The safe storage of the private-key of key agreement algorithm on the device is thus important.

5.2.1 Secure SoC: The Secure SoC(System on Chip) provides physical protection to secret keys by protecting the components like ROM, which is handling the secret keys, inside the Secure SoC. During execution time, the protected secure keys from the Secure ROM has to be loaded to the RAM in clear text and during that time the bus from the Secure ROM to the RAM can be monitored to access the secret keys. This can be prevented by allocating buffers for secret keys or intermediate values of cryptographic operations involving secret keys in the Internal RAM of the Secure SoC. This prevents the protected keys being available to any bus outside the Secure SoC

5.2.2 Secure ROM: One method for storing the device secret keys securely in the persistent storage of a device is to encrypt the secret keys before storing. Thus even if anyone managed to get the data out of the persistent storage he/she will never be able to understand the secret keys. Encryption algorithm is used that is only known to the device manufacturer.

5.2.3 Internal RAM and Secure Process: The buffers for secret keys or intermediate values of cryptographic operations involving secret keys are allocated in the Internal RAM of the Secure SoC to prevent the secret keys being available to any bus outside the Secure SoC. Let this memory area in the internal RAM be called as

Secure Memory Area. Not every process should access this memory area. Only the processes with special OS privilege, Secure Process, should be able to access the Secure Memory Area. This is analogous to process with administrative privilege or root privilege in an operating system.

5.2.4 Secure Bootloader: On startup before loading the firmware code, the Secure Bootloader checks whether the firmware is genuine or not and prevents the device from booting up if the device firmware is modified or replaced.

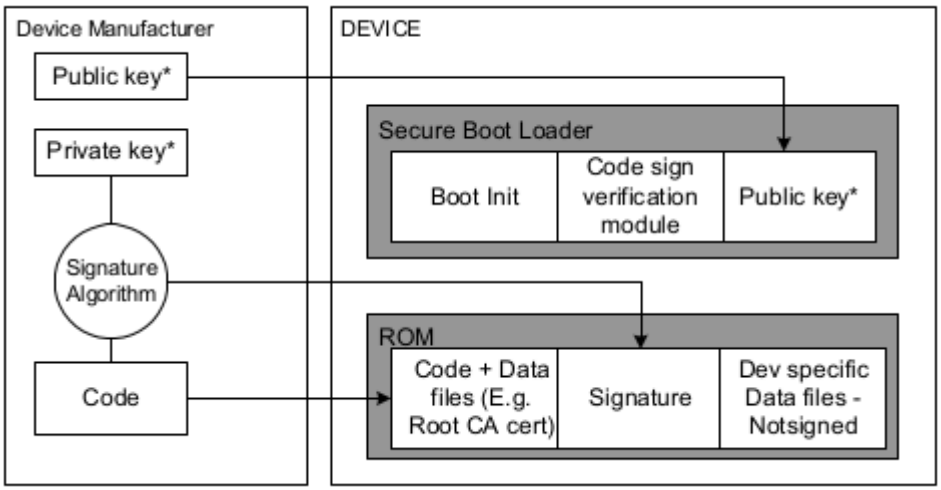

Fig. 4: Signature Verification During Boot

If the firmware of the device is non-upgradeable then the level of security in the boot loader can be enforced in much simpler method without the presence of a Secure Bootloader. In such case, writing the entire firmware in a read-only memory and configuring the bootloader to boot only from the given read-only memory area can prevent any unauthorized firmware component to run on the device. But in many cases, non-upgradeable firmware brings too much limitations on a product.

5.3 System Time

System Time plays an important role in device device security. Device must be designed maintain and sync time properly and periodically.

The digital certificate of a device generally comes with validity period. The validity periods varies across different protocol implementation. Some protocols like SSL specify a fixed validity period of few years or decades whereas other protocols like DTCP specifies infinite validly for a certificate.

The system time in an embedded device will generally have interfaces to user to set or modify the system time. For certificate verification process the device should maintain a system time that is different from the system time modified by the user so that the users are not able to modify the system time and make the device accept an expired certificate. It is also important that the timer keeps counting even after when the device is in the switched off state.

V. CONCLUSION

Despite an almost constant stream of media reports of cyber attacks and hacking incidents, there are still many devices that do not use encrypted communications or proper authentication. It is crucial that smart home devices, or any IoT devices for that matter, use mutual authentication and encryption. Generally IoT devices come with embedded software. The problem with the embedded software is that, it is device specific software so generally it is hard to update. So it is the responsibility of the device manufacturers to take extreme care in developing, testing these softwares and making them bug free as far as possible. They should issue updates regularly too. Also IoT devices often have less memory and slower CPUs, so they may be unable to use the same encryption methods as a traditional computer does, but that is no excuse for the lack of strong encryption. There are efficient cryptographic methods designed for small scale devices, such as Elliptic Curve Cryptography (ECC), which can be used. With all of these issues affecting the devices on different levels, it is currently not easy to deploy multiple smart devices in a secure fashion. Fortunately, there are ways to improve the overall security, as we highlighted above.

Although the more the hardware and software security measures implemented in a device to protect its secret keys and other secure data, the more costly the device will be. It should be noted software based security measures are cheaper compared to hardware protections. But without some standard hardware protection, software based cryptography alone can not ensure sufficient security. However the security measures implemented in the device are a trade off between the cost of implementation and the cost of the data protected. Achieving a cost effective yet foolproof method to protect the secret keys and secure data within the device will be a boon to the owner of the contents that needs security.

REFERENCES

[1] http://www.gartner.com/newsroom/id/2905717

[2] https://www.owasp.org/index.php/OWASP_Internet_of_Things_Top_Ten_Project

[3] Doddapaneni.Krishna Chaitanya, Ghosh.Arindam,"Analysis of Denial-of-Service attacks on Wireless Sensor Networks Using Simulation", 2011.

[4] DATA ENCRYPTION STANDARD (DES), FIPS PUB 46-3

[5] "Announcing the ADVANCED ENCRYPTION STANDARD (AES)", Federal Information Processing Standards Publication 197, November 26, 2001

[6] Secure Hash Standard (SHS), FIPS PUB 180-4

[7] Elaine Barker, Don Johnson, and Miles Smid, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)", NIST Special Publication 800-56A March, 2007

[8] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems"

[9] Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid, "Recommendation for Key Management- Part 1: General (Revision 3)", NIST Special Publication 800-57

[10] Digital Signature Standard (DSS), FIPS PUB 186-4

[11] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", NIST May 2008

[12] Mario Ballano Barcena, Candid Wueest, "Insecurity in the Internet of Things",Version 1.0 – March 12, 2015

[13] Sayantan Majumdar, Abhisek Maiti, Biswarup Bhattacharyya, Asoke Nath, "A new encrypted Data hiding algorithm inside a QRCode™ implemented for an Android Smartphone system:S_QR algorithm", International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163 Issue 4, Volume 2 (April 2015)

[14] Sayantan Majumdar, Abhisek Maiti, Biswarup Bhattacharyya, Asoke Nath, "Advanced Security Algorithm Using QRCode™ Implemented for an Android Smartphone System: A_QR", International Journal of Advance Research in Computer Science and Management Studies(IJARCSMS) ISSN: 2321-7782 (Online) Volume 3, Issue 5, May 2015