Secure Firmware Updates in the Internet of Things: A survey

Njabulo S. Mtetwa
Department of Computer Science
University of Zululand
KwaDlangezwa 3886, South Africa
mthethwansm@gmail.com

Paul Tarwireyi
Department of Computer Science
University of Zululand
KwaDlangezwa 3886, South Africa
tarwireyip@unizulu.ac.za

Adnan M. Abu-Mahfouz
Council for Scientific and Industrial Research (CSIR)
Pretoria 0184, South Africa
a.abumahfouz@ieee.org

Matthew O. Adigun
Department of Computer Science
University of Zululand
KwaDlangezwa, South Africa
adigunm@pan.unizulu.ac.za

***Abstract*— The Internet of things is an infrastructure of connected things (sensors, refrigerators, smart phones, etc.) which is changing the way we live, play, work, communicate and conduct business. Even though these things have such great impact in our lives, they are susceptible to security issues and vulnerabilities, which often result in negative consequences such as loss of confidentiality, integrity and availability. Hence, some users are still skeptical about the use of IoT. This paper explores the security problems that arise due to IoT device firmware updatability. New vulnerabilities are uncovered all of the time. If a device is non-upgradeable, then the vulnerability will exist for the rest of the device's lifetime. As a countermeasure, it is imperative to design security measures to enable automatic update of IoT devices. This paper focuses on current research work based on IoT firmware updates with the aim of highlighting issues related with security of firmware updates. It specifically focuses on the security challenges that face low-end IoT devices and Low-power Wide Area Networks.**

Keywords— Firmware, Internet of Things, Security, LPWAN, Low-end devices.

I. INTRODUCTION

Internet of Things (IoT) has experienced exponential growth both in research and in industry; however, privacy and security remain a challenge [1]. Several types of malicious activities exist that attempt to compromise the security and expose the privacy of the IoT devices[2]. These various malicious activities are motivated by known vulnerabilities that exist in the IoT therefore, it is important that after the initial deployment, the IoT devices need to stay updated and well patched to mitigate subsequent security vulnerabilities which may lead to various attacks[3].

Internet of Things devices can be updated using wireless communication technologies categorized as Short Range Networks and Low Power Wide Area Networks (LPWANs). Short Range Networks include communication protocols such as Bluetooth, Wi-Fi, WiMax, ZigBee[4]. These communication protocols could not address the needs for IoT devices, which include long battery life, long range data transmission and low power consumption. To accommodate these needs the Low-Power Wide Area Networks communication protocols were developed. This includes LoRa, NB-IoT, Sigfox, IQRF[5]. Both of these communication technologies present their own security challenges. For instance in LPWANs offers long-range connectivity therefore it is more exposed to various attacks.

One of the top hack took place in 2015 called the jeep hack[6]. Two researchers took advantages of many vulnerabilities including the firmware update vulnerability, where reverse engineering was performed on the firmware. As the result, the researchers were able to take control of a jeep using the vehicle's Controller Area Network (CAN) bus, which enables communication between different elements on vehicle such as steering wheel, breaks, heaters, locks, headlights etc. They were able to send CAN messages taking control of various elements of the vehicle to make it speed up, slow down and even veer of the road. The lack of security on firmware update made this attack possible therefore, strong encryption mechanisms are required to ensure security during the firmware updates.

Firmware updates is a challenge in IoT due the various reasons such as resource constraints devices since most of the present technologies is not suitable for the IoT devices due to their nature and limitations such as storage and processing power. These limitations make it difficult to secure the updates for the IoT devices. Open Web Application Security Project (OWASP) has listed vulnerabilities based on IoT that attackers use to compromise IoT devices. This includes insufficient authentication/authorization, lack of transport encryption, privacy concerns, insufficient security configuration, poor physical security and insecure software firmware updates [7].As attacker, take advantage of these vulnerabilities one of the recovery mechanism would be to initiate a secure firmware update procedure . The purpose of the firmware update procedure is to fix bugs and improve device functionality[8]. If the initiated firmware update mechanism is insecure it could lead to compromise of the user data, enable unauthorized control over the device, which can lead to launch more attacks against other devices.

The contribution of this survey are:

- We discuss the overview of over-the-air (OTA) updates and the important components that need security during the update process.
- The paper provide the security challenges that exists when updating low-end IoT devices in LPWANs.

- This paper gives the current state of the art based of the firmware update solutions presented by different researchers. The focus is based on the specific category of IoT devices (Low-end IoT devices – battery powered), which uses LPWANs for communication. Classify the existing studies based on what type of device the existing mechanism are targeting (Low-end IoT devices and Medium/High-end IoT devices).

This paper is structured as follows. Section II provides with the overview of firmware updates and common threats involved as the firmware is distributed from the manufacturer to the IoT devices. Section III provides the challenges that exist when applying the firmware updates in IoT devices with mainly focus in low-end IoT devices. This section looks at what makes difficulties when the firmware updates are applied in LPWANs. Section IV presents approaches for firmware update mechanisms only in the context of IoT. Section V consist of the discussion of literature visited on section IV. Section VI consists of the conclusion.

II. Overview of Firmware Security Threats

This section provides with the overview of firmware updates and common threats involved as the firmware is distributed from the manufacturer to the IoT devices. It looks at the three main entities that needs to be secure namely firmware repository, communication channel and IoT device.

There are different attacks that can happen at different levels as the firmware is transmitted to the IoT devices and after the firmware has securely delivered to the IoT. Figure 1 depicts the possible threats in IoT.

1) Firmware manufacturer is the entity that produces the new version of the firmware image and distributes it over the untrusted network.

2) The untrusted network communication channel may be eavesdropped by an attacker. An attacker can take hold of the firmware image and extract sensitive data from it and the file can be modified and returned for a distribution.

3) Customers get the firmware then distributes it into the IoT devices.

4) The same attacks of insecure channel may take place as the firmware is distributed from customer to the IoT device. Additional risks maybe involved such as loading unauthorized firmware onto unauthorized devices or to completely abort the update procedure.

5) An attacker can extract sensitive information from the device such as the keys and even do attacks directly to the firmware such as system-safety patch vulnerabilities, Firmware bricking[9].

Figure 1 depicts the possible attacks at the communication level and physical layer. Therefore, the firmware needs to be secure both in transit and at rest. Figure 1, also shows three main elements that can be compromised during the update process namely firmware repository, communication path and IoT device.

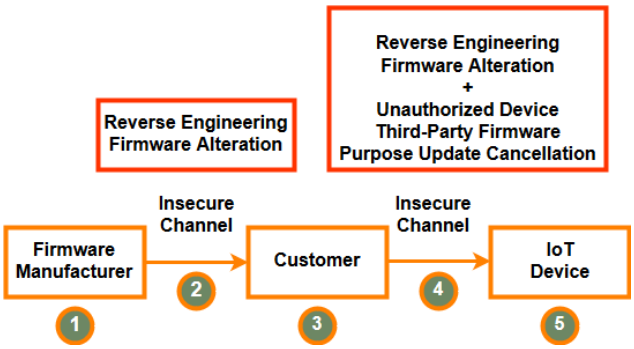

Fig 1. Possible attacks associated with the firmware update[10]

Figure 1 depicts the possible attacks at the communication level and physical layer. Therefore, the firmware needs to be secure both in transit and at rest. Figure 1, also shows three main elements that can be compromised during the update process namely firmware repository, communication path and IoT device:

A. Firmware Repository

The firmware must be securely store on the repository. This is the initial step the attackers use to make attack possible. Attackers can get hold of the firmware in different ways such as obtaining it from the vendor's website, google support and community forums, reversing the mobile application, sniffing the OTA update mechanism and dumping it from the device[11]. Once the firmware is obtained, then reverse engineering can be performed to extract the sensitive information such as Application Programming Interface (API) and encryption keys, Access token, encryption algorithms, hard-coded credentials, sensitive URLs and more[12]. Therefore, it is important to store the firmware on the repository securely. The firmware must be encrypted and signed before it gets stored to the repository. Usually Advanced Encryption (AES) is used to ensure confidentiality and integrity of the firmware at rest. Apart from AES the XOR encryption[13] can be used to encrypt the firmware.

To ensure authentication, confidentiality and integrity the firmware manufacturer must sign the firmware image using the private key, which is held secret. When a firmware has this signature attached to it, a device with the feature enabled will validate the firmware before accepting to install it. The process of signing firmware is initiated through the computation of a cryptographic hash value. The value is then signed with the private key of a private/public key pair before the signature is attached to the firmware image figure 2 shows this process.

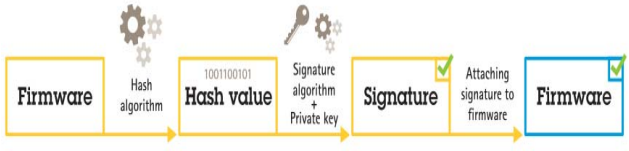

Fig 2: The process of signing firmware[14].

B. Communication Path

After the firmware has been encrypted and signed it is then ready to be transmitted over the communication channel. To ensure the securely delivery of the firmware over the channel, the SSL can be used to provide transport

encryption to an already encrypted firmware. SSL utilizes the digital signatures to provide data integrity, confidentiality and authentication[15] to mitigate the man-in-the-middle attacks such as spoofing attacks, reverse engineering.

C. IoT Device

After the firmware has been securely delivered to the IoT device then the firmware needs to be flashed. Before it gets flashed possible attacks could happen such as time-of-check-to-time-of-use attack (TOCTOU). The security needs to be applied to ensure consistent integrity of the firmware [16], meaning the integrity of the firmware must be also checked after the firmware has been flashed to memory. The physical security of IoT devices is also required. Unnecessary ports such as JTAG and UART should be blocked to avoid any interruption in update process by adversaries[16].

There are two firmware update strategies can be taken to updates the IoT devices. The popular one is through the client-server model where the device manufacturers use servers (possibly using cloud providers) to distribute firmware updates to IoT client devices. This centralized approach exhibits a single point of failure for both the availability and the integrity of the firmware update [17]. The second strategy is blockchain-based. It has more advantages over the client-server model by able to keep track of all events[18] (stored in the immutable ledger) associated with the firmware update. It provides manufacturers with the ability of using smart contracts to enforce the firmware updates conditions in a flexible manner. The smart contract (logic) is securely stored and decentralized on every peer on the network, and also consider to be permanently tamperproof [19], unlike the client-server logic, which is centralized and exposed to a single point of failure. The distributed nature of the blockchain frameworks makes blockchain ledger and smart contract to be resilient to network failures and cyber-attacks.

III. Privacy and Security Challenges

In this section, we focus on the security challenges and privacy needed specific for the Low-End IoT device. There are two main aspects explained which includes IoT devices, and the challenges faced with IoT communication/connectivity specific in LPWANs.

A. IoT Devices

IoT devices can range from smartphones to RFID readers, wearable devices to tablets, gadgets. These devices are categorize into three categories, the Low-end IoT devices, Middle-end IoT devices and High-end IoT devices[20]. The main difference between these devices is hardware. Some of these devices are battery-powered and consist of different sizes of Random Access Memory (RAM) and ROM, which make it difficult to distribute firmware. The most challenging categories of IoT devices to update firmware are the low-end IoT devices. This is because of very limited resources compare with other IoT devices. For example, a RFID tag consists of a single 16-bit processor, with 6-12 MHz in an energy saving mode, with a RAM of 512 bytes and 16 Kbytes of flash storage. The RFID tag devices are unable to provide authentication and integrity due to such constraints.

TABLE I. Classes of Low-End IoT devices [20]

Specifications	Class 0	Class 1	Class 2
RAM	<< 10kB	**≈ 10kB**	**≈ 50kB**
Flash	<< 100kB	**≈ 100kB**	**≈ 250kB**
Communication Protocols	No protocol stacked embedded, use gateways for communication.	Communication via lightweight protocols such as CoAP.	Communication protocols such as HTTP are supported

During the firmware update process the device is expected to verify whether the firmware image is coming from the right source. This is possible with the high-end IoT devices through the verification of the manufacturer's certificate. However, this this is not possible with the low-end devices due to resource constraints. Table I shows the classes of low-end devices with their memory storage and protocols supported. These supported protocols such as HTTP do not provide security to the devices, most of these protocols need to be integrated with others to provide security. There are some works done around this context for instance, [15] shows that it is possible to implement traditional cryptographic technologies on low-end IoT devices also demonstrated that is it possible to implement the firmware update solution without exceeding threshold of 32kB of RAM and 128 of flash memory.

Even though the signature's/asymmetric encryption solves the trust challenge during the firmware update. They have disadvantages in terms of processing time and in terms of data since, they are sent along with the packets and result to the increase of the packet size [21].

B. Connectivity and IoT Communication

There are several options available for the IoT devices to connect to the internet this include cellular, satellite, Wi-Fi, Bluetooth, NFC, LPWAN and Ethernet. The LPWAN focuses to provide long-range communication for low power devices. The nature of LPWAN make firmware updates to be challenging for low powered devices, this is true since some of these networks (Sigfox, LoRa) operate in the unlicensed spectrum (ISM band) and they cannot offer the same QoS which is offered by other networks i.e. NB-IoT [22]. This enables the messages that are sent over LPWAN not to be received by the gateway due to the interference of the signal sent (packet loss). When it comes to each network/communication protocol of LPWANs the security is offered differently. For example in LoRa, AES-128 is used for encryption as a method that provides multiple layers of encryption in LoRaWAN. The network keys and application keys used to provide security to the packets over the network [23]. Even though the LoRa have such support of security but LoRa devices are still susceptible to replay attack, jamming attacks, wormhole attack.

The communication over the networks is bi-directional which make it possible to provide the firmware update OTA. Even though the communication is bi-directional but the channel is not always open (downlink) for most of

these networks. For instance, LoRaWAN has three type of devices namely class A, B and C. Class A downlink transmission is allowed after a successful uplink transmission which opens through RX windows and these windows uses the channel with the low data rate[24]. This benefits the battery life of the device since it is not always listening and receiving the packets however, it has disadvantages when it comes to firmware updates. The technologies with low data rates (Sigfox, LoRaWAN) are extremely affected with the increase in packet size [25] and affects the battery life. For instance, in LoRaWAN to send a 100kB of firmware image, the exchange of 891 messages is required[26].

Table I shows the classes of low-end devices with their specifications. The table indicates that some of the supported protocols for the devices is Constrained Application Protocol (CoAP). CoAP is an application protocol that sends the information in unsecure manner. However, there are developed protocols such Datagram Transport Layer Security (DTLS) to integrate with CoAP [27]. DTLS provides end-to-end encryption just like TLS. However, the DTLS handshake procedure give rise to computation overheads and excessive message signaling, which are not clearly suited for the LoRaWAN network[28].

There are other developed protocols that provide updates to resource-constrained devices with the aim of ensuring authentication, confidentiality, and data integrity. For example, Lightweight Machine-to-Machine (LwM2M) that provides an API for constrained devices to manage firmware updates. LwM2M specifically provides a firmware object that enable the installing of firmware package, updating firmware, and performing actions after updating the firmware[29]. However, issues such as packet loss that results in time consumption of firmware update still exist. The connectivity problem has bad effect on the performance of the battery-powered device. If there is loss of packets, it is required to resume the firmware updates to resend the packets. This requires more of downlink and make the device to loss power.

IV. Firmware Update Mechanism

This section looks at the existing firmware updates mechanism available for the IoT. The literature is categorized based on what type of IoT devices the update mechanisms are targeting i.e. low-end IoT devices, Medium-end IoT devices/high-end IoT devices. On each of these categories the LWM2M/server-based, blockchain-based mechanism can be found.

A. Firmware Targeting Low-End IoT Devices

In [10], authors present the secure delivery of firmware updates to the internet of things devices as well as a design of safe and secure bootloader for radio-frequency identification reader. The main goal of authors was to find out whether it would be possible to integrate security features such as AES, which is used for encryption, as well as other security features into the existing IoT devices. Authors developed an application in order to encrypt the firmware image file and be able to flash the firmware in the devices. The application uses AES to encrypt the firmware file where the encryption key is required together with the initialization vector, which is an arbitrary number that can be used along with the encryption key. Authors concluded that it is possible to integrate such kind of encryption and it leaves more space to integrate other security techniques. The results show that minimum flash memory of 49.7 kB and RAM of 10 kB are required.

The authors of [30] presented a Firmware Over The Air(FOTA) procedure for the IoT devices and introduced a new secure object. The work tries to improve the issues that is faced with the LwM2M protocol. Currently, with the protocols like LwM2M cannot handle loss of packets which is due to network leakage, note the firmware update process maybe interrupted due to the network leakage. This work proposes the new secure object to save power and to provide longevity on IoT devices. The authors of [31] also presented the architecture of firmware software update infrastructure that utilizes a centralized server to distribute the firmware to large number of embedded devices. The work aims provide update to resource-constrained device where the microprocessor ATMega128 was used to test the prototype. ATMega128 consists of ARV Core with better processing capabilities and consist of 128Kb of flash memory.

B. Firmware Targeting Medium/High-End IoT Devices

In [32], the authors represent a protocol for securing the firmware updates over the air in intelligent vehicles. This protocol ensures the data integrity, data confidentially, and data freshness. Moreover its identifies an attack model where it assume that the portal which communicates with the vehicles over the wireless communication is well protected and not considered a target for intrusion and DoS attacks.

This is assumed for the purpose of focusing the security in transmission of the firmware. It is assumed that the vehicle cannot be subjected to intrusion and DoS attacks that means only the attacker targets the communication link. The new firmware is processed at the portal before it reaches the vehicle and it is divided into data fragments where each fragment is hashed. Each fragment contains hash of a previous fragment therefore; the whole firmware forms a hash-chain. Note the hash-chain provide the integrity and all the packet in the chains are integrity protected excluding the first packet. The integrity of the first packet is achieved with asymmetric keys. As the first packet is sent from the portal to the vehicle, it is signed with the private key of the portal to provide integrity like other packets. The authenticity of the firmware is achieved through the hash-chain and though the first signed packet as the vehicles uses portal public key to confirm the origin of the firmware. The data confidentiality of the firmware is obtained through symmetric key, which encrypt and decrypt the packet.

Finally, the data freshness is obtained since the first signed packet consists of the firmware information such as the firmware version. The protocol also avoids the replay attack by using hash-chain in the communication channel. This technique maybe used for avoiding replay attacks against LPWAN.

[33] Proposed a solution on how to secure the firmware updates on the IoT gateway devices, which aims to assure the proof of origin, integrity and confidentiality in transit of the firmware image. Furthermore, it defeats the most

relevant external security threats. Finally, it measures the network overhead and energy consumption after that the comparison with the related work is done. Let see how the security is applied in the entire solution. There are four components used which are:

- Development tool - used by the manufacturers to generate the images and to upload the images to the signing server
- The signing server - receives the firmware update images and include keys, certificates and configuration files in them. The IoT devices to authenticate images and the identity of the update server during TLS use these keys and certificates. Update server calculates the SHA512 hash of the firmware and sign it with its RSA private key then uploads the firmware package to the update server.
- The update server – is responsible for alerting the IoT devices about the new updates.
- The device daemon. – is a long running process, which periodically sends message to the update server to query new updates.

Shortly, these entities utilize TLS to secure the firmware image over the channel and the firmware image is digital signed using the private key. A checksum algorithm SHA256 is used to ensure the integrity of the firmware image.

The authors of [34] proposed and implement a decentralized firmware update framework called Código network, which is implemented on top of the Ethereum blockchain, and the IPFS network. This target to achieve a framework, which will allow no single point of failure, scalability, transparency of firmware updates, equivalent security code with code signing. The code signing which achieved through the use of digital signature and also through the use of Ethereum smart contract to ensure whether the firmware has been corrupted by comparing the hash stored in the smart contract with the hash of the firmware. The proposed solution was experimented with 10MB of the firmware image. The firmware was distributed in three different storages include the server, BitTorrent and IPFS and the time it took to distribute the firmware was measured on each storage. To multicast/distribute a single file to 100 devices it took 2.491 seconds for client-server. In IPFS, it took 57.053 seconds and for BitTorrent it took 331.489 seconds.

[35] Proposed a firmware update mechanism utilizes hyperledger fabric (blockchain) and chaincode (smart contract) that ensures the integrity of the firmware during the update process. The proposed mechanism is scalable and support for heterogeneity of IoT devices in the smart cities. The proof-of-concept is implemented with the use of D1 MIN board which is ESP8266 based IoT device consisting of 4MB of flash memory. The firmware is processed in fragments since the device consist of RAM approximately to 50KB. The integrity of the firmware is achieved with the use of SHA256 algorithm than using built in MD5 algorithm for ESP8266. In [36] authors presented a framework for self-verification of firmware update over-the-air, which achieves the security of firmware binary after it has been downloaded. The portal, which communicates with the vehicle ECU, generates a nonce, divide firmware into blocks and individual block has a hash. The following block includes the hash of the previous block hence it creates the hash-chain of memory contents. Moreover, the final block hash is used as verification code. Now the nonce, verification code and firmware binary is sent to the device using a secure firmware update that guarantees the authenticity and integrity of the downloaded data. This information is sent for the later use to determine the integrity of memory contents of the flashed firmware.

[37] proposed a scheme that utilizes a blockchain technology to securely check the firmware version, validate the correctness of the firmware and download the latest firmware for the embedded devices. In the proposed scheme every IoT device represents a node in the blockchain, which means they are required to store the blockchain ledger in their local storage. The challenge with this, is that most of the IoT devices have limited resources such as energy, computation and storage capacity. This mechanism might be difficult to be implemented in the real world IoT environment.

[38] proposed to use the blockchain technology to securely update software and firmware of the IoT devices. The firmware update solution proposed focuses on the resource constrained IoT devices such as the Wi-Fi smart plug and sensors. This work only provides the integrity verification of the firmware therefore more security is needed beyond integrity such as determining whether the firmware is coming from the legitimate source; ensure confidentiality, non-repudiation and data freshness.

V. DISCUSSION

This section provides with a discussion of the literature and represent the Table II, which describe the important features of firmware update in IoT.

From the literature it is observed that many recent studies [22],[23],[24],[25],[26],[27],[28] are focusing on the middle/high-end IoT devices while few are done on low-end IoT devices. All of the firmware update solutions provided on literature may all work for high-end devices but not for the low-end devices therefore, it is will be wise to provide a solution that will work in both cases. The main challenge that is faced by LPWANs and low-power devices is the connectivity issue. [20] was able to show how to handle firmware update process if the connection break or if there is loss of packets. All of the viewed literature based on the blockchain do not provide with the solution on how the connectivity issues with loss of packets can be solved. In fact, none blockchain solutions have tried to solve the firmware updates in the context of LPWAN, a solution which will be able to take account of battery-powered devices and connectivity issues of LPWANs.

Another important aspect that need attention is availability of the firmware. [24],[25] used the decentralized storage which is IPFS and BitTorrent to distribute the firmware. Both of these papers utilizes blockchain however, [28] utilized a server to distribute the firmware while using blockchain.

TABLE II. COMPARISON BETWEEN THE STUDIES.

	Client-Server Based					Blockchain-Based				
Addressed features	**Kvarda [12]**	**Doddapeni [30]**	**Jurković [31]**	**Oka et.al [32]**	**Alexandre[33]**	**nanopoulos et.al [34]**	**He et al [35]**	**Nillson [36]**	**Lee [37]**	**Yohan [38]**
Target Low-End Device	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Target Low-High Device	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
heterogeneity	✗	-	✗	✗	✗	✗	✓	✗	✗	✓
Availability	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗
Authentication	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗
Integrity	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Confidentiality	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗
Data Freshness	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗
Handle Connectivity	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Performance analysis	✓	✗	✓	✓	✓	✓	✓	✗	✗	✗

✓: Covered Feature ✗: Uncovered Feature

Therefore, the firmware image is centralized even though blockchain is a distributed network. This enable a single point of failure unless the firmware image is kept on different locations.

VI. CONCLUSION

The Internet of Things is growing exponential with lot of the devices being deployed and connected to the internet. These devices are resource constrained and for this reason, it is hard to integrate the existing cryptographic techniques since most of these require more resources. This has more effect it comes to firmware update. In this paper, we investigated the security issues and challenges faced with the resource constrained devices with more focus on low-end IoT devices. We first provided with the basic overview of the existing threats in firmware update. The further discussed the security issues and challenges faced in firmware updates with the resource-constrained devices in LPWANs. The current state-of-the-art was presented and categorized based on what type of the device the mechanisms are targeting and the security approaches took by researchers were discussed.

ACKNOWLEDGMENT

We acknowledge the support from the Department of Computer Science at University of Zululand and we are also grateful for research funding and support from CSIR.

REFERENCES

[1] K. R. Özyılmaz and A. Yurdakul, "Designing a blockchain-based IoT infrastructure with Ethereum, Swarm and LoRa," pp. 1–6, 2018.

[2] Z. Alansari *et al.*, "Internet of Things: Infrastructure, Architecture, Security and Privacy," in *IEEE International Conference on Computing, Electronics & Communications Engineering*, 2018, vol. 2018, no. August, pp. 211–238.

[3] A. Boudguiga *et al.*, "Towards better availability and accountability for IoT updates by means of a blockchain," *Proc. - 2nd IEEE Eur. Symp. Secur. Priv. Work. EuroS PW 2017*, pp. 50–58, 2017.

[4] D. Ismail, M. Rahman, and A. Saifullah, "Low-power wide-area networks," pp. 1–6, 2018.

[5] Leverege LCC, *An Introduction to the Internet of Things*. .

[6] Z. Tyree, R. A. Bridges, F. L. Combs, and M. R. Moore, "Exploiting the Shape of CAN Data for In-Vehicle Intrusion Detection," *IEEE Veh. Technol. Conf.*, vol. 2018-Augus, pp. 1–5, 2019.

[7] OWASP, "OWASP Top 10 Internet of Things," *Salem Press Encycl. Sci.*, pp. 5–7, 2018.

[8] M. Kameswarao & P. Bhavya Sree, "a secured firmware update procedure to prevent cross channel scripting attack in embedded devices," *Int. J. Electron. Commun. Eng.*, vol. 2, no. 2, pp. 161–168, 2013.

[9] D. J. Chris, M. H. Saleem, M. Evanglopoulou, M. Cook, and R. Harkness, "Defending Against Firmware Cyber Attacks on Safety-Critical Systems."

[10] L. Kvarda, P. Hnyk, L. Vojtech, Z. Lokaj, M. Neruda, and T. Zitta, "Software implementation of a secure firmware update solution in an IOT context," *Adv. Electr. Electron. Eng.*, vol. 14, no. 4Special Issue, pp. 389–396, 2016.

[11] R. A. Grimes, "IoT Hacking," *Hacking the Hacker*, pp. 189–191, 2017.

[12] A. Gupta, *The IoT hacker's handbook [electronic resource]: A practical guide to hacking the internet of things / Aditya Gupta*. 2019.

[13] A. A. Tamimi, A. M. Abdalla, and P. O. Box, "An Image Encryption Algorithm with XOR and S - box," pp. 166–169.

[14] A. Axis Communications, "Signed firmware, secure boot, and TPM key storage in Axis products," no. November, 2018.

[15] J. Clark and P. C. Van Oorschot, "SoK: SSL and HTTPS: extended version," *Proc. - IEEE Symp. Secur. Priv.*, pp. 511–525, 2013.

[16] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Journal of Network and Computer Applications Blockchain ' s adoption in IoT : The challenges , and a way forward," *J. Netw. Comput. Appl.*, vol. 125, no. November 2018, pp. 251–279, 2019.

[17] L. Hang and D. H. Kim, "Design and implementation of an integrated iot blockchain platform for sensing data integrity," *Sensors (Switzerland)*, vol. 19, no. 10, 2019.

[18] B. Pichon, O. Kahl, B. Hammer, and J. S. Gray, "Blockchain - an introduction," vol. 6, no. 4, pp. 382–387, 2006.

[19] K. Sultan, U. Ruhi, and R. Lakhani, "Conceptualizing Blockchains : Characteristics & Applications," *11th IADIS Int. Conf. Inf. Syst. 2018*, pp. 49–57, 2018.

[20] M. O. Ojo, S. Giordano, G. Procissi, and I. N. Seitanidis, "A Review of Low-End, Middle-End, and High-End Iot Devices," *IEEE Access*, vol. 6, no. November, pp. 70528–70554, 2018.

[21] B. Benjamin and B. Brown, "Over-the-Air (OTA) Updates in Embedded Microcontroller Applications : Design Trade- Offs and Lessons Learned," no. November, pp. 1–7, 2018.

[22] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019.

[23] S. Chacko and M. D. Job, "Security mechanisms and Vulnerabilities in LPWAN," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 396, no. 1, 2018.

[24] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the Limits of LoRaWAN," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, 2017.

[25] T. G. Durand, "by," no. December, 2018.

[26] J. Jongboom and J. Stokking, "Enabling firmware updates over LPWANs," *Embed. World Conf.*, 2018.

[27] S. Arvind and V. A. Narayanan, "An Overview of Security in CoAP: Attack and Analysis," *2019 5th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2019*, pp. 655–660, 2019.

[28] I. You, S. Kwon, G. Choudhary, V. Sharma, and J. T. Seo, "An enhanced LoRaWAN security protocol for privacy preservation in IoT with a case study on a smart factory-enabled parking system," *Sensors (Switzerland)*, vol. 18, no. 6, pp. 1–32, 2018.

[29] S. Rao, D. Chendanda, C. Deshpande, and V. Lakkundi, "Implementing LWM2M in constrained IoT devices," *2015 IEEE Conf. Wirel. Sensors, ICWiSE 2015*, no. September, pp. 52–57, 2016.

[30] K. Doddapaneni, R. Lakkundi, S. Rao, S. G. Kulkarni, and B. Bhat, "Secure FoTA Object for IoT," *Proc. - 2017 IEEE 42nd Conf. Local Comput. Networks Work. LCN Work. 2017*, pp. 154–159, 2017.

[31] G. Jurković and V. Sruk, "Remote firmware update for constrained embedded systems," *2014 37th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2014 - Proc.*, no. May, pp. 1019–1023, 2014.

[32] D. K. Oka, E. K. K. Escrypt, D. K. Nilsson, and U. E. Larson, "Secure Firmware Updates over the Air in Intelligent Vehicles Secure Firmware Updates over the Air in Intelligent Vehicles," no. June 2008, pp. 380–384, 2014.

[33] T. Alexandre, "UpdaThing : A secure and open firmware update system for Internet of Things devices," no. October, 2016.

[34] S. A. Nanopoulos, "Código Network : a Decentralized Firmware Update Framework for IoT Devices," 2018.

[35] X. He, S. Alqahtani, R. Gamble, and M. Papa, "Securing Over-The-Air IoT Firmware Updates using Blockchain," no. May, pp. 164–171, 2019.

[36] D. K. Nilsson, L. Sun, and T. Nakajima, "A framework for self-verification of firmwareupdates over the air in vehicle ecus," *2008 IEEE Globecom Work. GLOBECOM 2008*, pp. 1–5, 2008.

[37] B. Lee and J. H. Lee, "Blockchain-based secure firmware update for embedded devices in an Internet of Things environment," *J. Supercomput.*, vol. 73, no. 3, pp. 1152–1167, 2017.

[38] A. Yohan and N. W. Lo, "An Over-The-Blockchain Firmware Update Framework for IoT Devices," *DSC 2018 - 2018 IEEE Conf. Dependable Secur. Comput.*, pp. 1–8, 2019.