International Journal of Research in Engineering, Science and Management
Volume-3, Issue-5, May-2020
www.ijresm.com | ISSN (Online): 2581-5792

A Review on FOTA Update Mechanisms Available for In-Vehicle Embedded Devices

M. S. Tejaswini[1*], Neeta B. Malvi[2]
[1]*Scholar, Dept. of Electronics and Communication Engineering, R. V. College of Engineering, Bangalore, India*
[2]*Assistant Professor, Dept. of Electronics and Communication Engineering, R. V. College of Engineering, Bangalore, India*
*Corresponding author: tejums24@gmail.com

***Abstract*: Global technological advancement in the field of Internet-of-Things (IoT) has made the connectivity ubiquitous. The IoT is so powerful that it is expanding its application to every field and thus it is predicted to connect one to three trillion devices by the end of 2025. Connected vehicle is now an emerging trend in automobile sector. A major feature of this connected vehicle technology is the capability to upgrade the in-vehicle embedded devices through firmware updates over the air (FOTA). The security of these firmware updates is very crucial because the firmware flashed onto the in-vehicle devices control various operations of the vehicle. In this regard several protocols are developed. These security layer protocols protect the binary image of the firmware only during transmission from trusted portal to end device. However, there exists a possibility that attackers can modify the downloaded binary before installation. Therefore, there is a need for secure FOTA object. In IoT context, privacy and security are the major concerns when large scale deployments are considered. Open Mobile Alliance (OMA) has set standards for FOTA. This paper presents a review of such existing standard protocols and schemes for FOTA. The short comings in existing techniques are identified and the issues are addressed.**

***Keywords*: Connected vehicle, Firmware-Over-The-Air, Internet-of-Things, Open mobile alliance.**

1. Introduction

IoT has set stage for smart global advancement and connectivity is ubiquitous. Things present in our everyday life, ranging from smart TVs to garden equipment and electronic toys to vehicles, are now getting connected through Internet. Millions of devices with sensing, actuating and controlling capabilities are now connected to Internet. In such atmosphere, automobile industry is also advancing technically by incorporating intense knowledge in the field of diagnosing automobile systems with the amazing potential of connectivity. IoT enables the vehicles to connect to the network, communicate and interact with each other and with the infrastructure. Today, a typical in-vehicle network will have more than 50 embedded devices called as Electronic Control Units (ECUs). As technology advances with time, the software updating of these embedded devices is necessary. This upgradation process helps to enhance the existing features or to fix any bugs found in the ECUs. To carry out this firmware update, vehicle manufacturers are following a mechanism known as firmware update over the air (FOTA). FOTA update has become an integral part of IoT. Communication protocols based on IP are used in establishing the communication between device management server and clients. IoT based device management platforms perform deployments on heterogeneous target devices. This is so because of the ever-changing requirements and updates of existing features. Therefore, FOTA has a huge scope in managing large scale IoT devices.

Though FOTA is widely adopted in latest electronic gadgets such as smart phones, tablets and other connected devices, there is no standard procedure or approach with the existing schemes. Device manufacturers have designed and developed their own proprietary mechanisms to carry out FOTA update. As the number of constrained edge devices such as vehicle ECUs, embedded devices and machine to machine devices getting connected through IoT is going to rise exponentially in future, there extremely exists a need for software and device management. Existing mechanisms suffer from restricted bandwidth to deploy updates and distribute latest software /firmware modules, insufficient memory resources and insufficient RAM to store FOTA packages and execute FOTA updates. The update process is expensive, prone to errors and is time consuming. Therefore, devices with such limitations will not be able to optimally update themselves with security [9].

Device manufacturers are in a potent need of a well-defined structure and standard approach. They are commonly referred as Objects in IoT context. Though several objects are suggested by Internet Protocols for Smart Objects (IPSO) [10], standardized models of FOTA still do not exist. Hence, it is proposed in this paper that there is a huge scope in enhancing the security of FOTA objects by incorporating standardized security protocols and object-oriented notations such as Java Script Object Notation.

The remainder of the paper is organized into four sections. Section 2 describes about FOTA, FOTA PUSH and PULL mechanisms, the existing protocols and strategies prevailing in the area of security such as CoAP, in addition OMA-DM and LWM2M DM are also discussed. Section 3 discusses about

identification of shortcomings in the existing methods. Section 4 discusses about the possible solutions to the identified problems and scope for research. Section 5 concludes the paper.

2. Study of Existing FOTA Systems

The firmware update over the air also known as FOTA, is now an emerging trend in automobile sector. It enables the vehicle manufacturers to rollout software/ firmware updates which will be used to upgrade the existing features and services or fix the discovered bugs in a location independent manner.

Most of the existing systems for performing FOTA are fragmented and some of them use unique proprietary solutions which are not compatible, which others and are not inter-operable. There exists a Device Management server, which consists of all the data related to target devices to perform new firmware update. The device management server launches FOTA process using either PUSH or PULL schemes.

FOTA PUSH technique is suitable for constrained edge limited resource devices with restricted protocol support. A typical PUSH method used in FOTA is as shown in Figure 1. It includes the following steps.

- The new firmware is sent in the form of packets of binary image through a secure transport layer using the device management server. The protocol can be CoAP [11].
- The device management client receives, verifies and saves the firmware image.
- Once the last packet of the binary is downloaded successfully, device management client indicates to the server that the download is finished successfully, or if there was an error that will be reported.
- If the device management server does not receive any acknowledgement from the client then it sends an update request and waits for the response.
- The device management client flashes the binary image of new firmware and acknowledges to server either a successful update or error.
- Any error during download or flashing will lead to re-starting the above process.

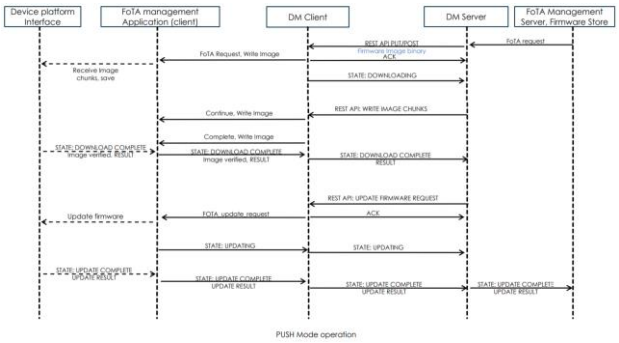

Fig. 1. PUSH mechanism in a typical FOTA process [12]

FOTA PULL technique is suitable for devices like gateways with enormous protocol stack, sophisticated logical management support and devices which are powered and connected to internet always. A typical PUSH method used in FOTA is as shown in figure 2. It includes the following steps.

- The device management server gives a URL of the new firmware update to the device management client and it waits for the completion of download. Device management client acknowledges the server.
- The device management client begins to download of the firmware using other methods like HTTP GET request.
- After a successful download or after a failure of download, the device management client will notify the device management server.
- If the device management server does not receive any acknowledgement from the client then it sends an update request and waits for the response.
- The device management client flashes the binary image of new firmware and acknowledges to server either a successful update or error.
- Any error during download or flashing will lead to re-starting the above process.

The downside of the omnipresent Internet Protocol is security. Numerous algorithms are there in cryptography to secure the data. [1] AES and IDEA are a few such algorithms to mention. Integrating cryptography with IoT has a lot of scope [2].

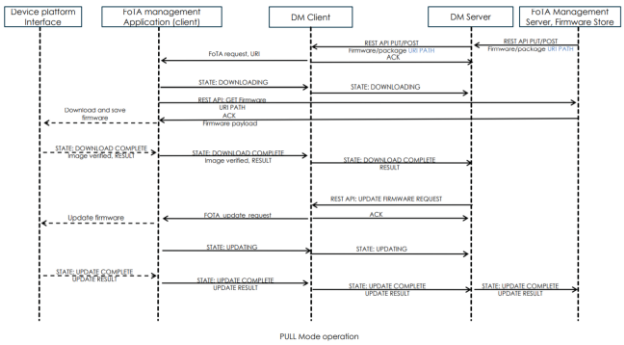

Fig. 2. PULL mechanism in a typical FOTA process [12]

The connected devices possess low computational strength and limited memory, therefore there is a need to optimize the existing schemes to implement secure firmware update [3]-[5]. An end-to-end secure data transfer strategy is very important to protect and device and data and thereby assuring CIA to the user [4]. The most important elements of security are confidentiality, integrity and availability. CIA can be used for IoT applications. It has evolved over time and now include privacy and authenticity. Data objects need additional security in IoT along with application layer security. IP security, DTLS are of the prevailing schemes and in some cases cannot provide complete security [6]. Channel based security protocols fail to secure the messages from intermediaries. The FOTA update binary image packets have specific target devices.

Fig. 3. CIA Triad diagram
(Image Source: https://www.ansible.com/blog/an-introduction-to-windows-security-with-ansible)

After the update package is designed and developed, the software module will be transmitted to the target device. This will be done using communication protocol. The device management system takes care of firmware/software repository management, monitoring target devices, and manages the FOTA update tasks. Open Mobile Alliance – Device Management (OMA-DM) is a device management protocol. It is used for device management of standard interfaces, mobile devices and interoperability of devices with different protocols. It is used for device management system to communicate with OMA-DM client. Firmware update management is defined by OMA-DM [7]. It offers a high-end interface between client and server to make the process of FOTA update smooth by offering schemes for download, install and status update. It offers FOTA PUSH and PULL schemes. Device provisioning is available. The target device can perform alternate download. Whereas, the standardized process for FOTA update, download and installation schemes, software package design and definition, error handling, pause and resume status, data recovery etc., are not well-defined.

Lightweight Machine to Machine Device Management (LWM2M DM), is developed for supporting constrained edge IoT devices and it is appropriate for end-to-end solutions. It is same as standard OMA-DM unlike this has lightweight interfaces. Client-server interfaces are defined under IEFT open standards, such as CoAP and TLS which are connected to SMS and UDP. Whereas, LWM2M DM offers a high-level interface between client and server. It offers rich interface for FOTA download, update, action status and reporting. The FOTA package is provided through PUSH technique and location is provided through PULL technique. The packages can be downloaded by target devices in alternate mechanisms. In addition, LW-OMA does not support a well-designed structure. In addition to the channel bindings, the LWM2M uses DTLS to achieve data security and integrity, authenticity and confidentiality [8]. This protocol offers three various credentials. Certificates, Raw public keys and Pre-shared secret keys are those three types.

Thus, from the study of existing schemes in this field, it can be noted that there is a requirement for a secure FOTA object. Moreover, platform independent and protocol independent meta data in hashed format helps in resuming a paused FOTA update to deliver the objects with security.

3. Identification of Limitations in Existing Schemes

The following action steps are carried out in sequence while executing a typical FOTA update.

- New firmware is designed and developed to perform functional upgrade of system/device.
- The software bundle which has to be transmitted over-the-air by the device management server, is converted into appropriate binary image and its meta data is tagged.
- The binary image of the firmware (including meta data) is fragmented into packets of data called "chunks" and then these chunks are transferred over-the-air to the target devices. Here, the data is protected only by application layer security protocol such as CoAP.
- In the client end, received chunks are rearranged and integrity is verified.
- After passing the integrity verification, new firmware if flashed on to the target device in the client end.

Upon studying current literature about the existing FOTA mechanisms, the following limitations were noticed:

- There is no option for partial update of firmware image on a target device.
- If connectivity is lost during the FOTA update process due to network interruption or power failure in target device, then there is no provision for resuming the FOTA update process from the point of interruption.
- Existing systems do not possess security isolation; thus, one error can crash the entire system/ device.
- The ECUs are in untrusted environment and the main challenges noticed here are how the ECUs will verify and validate the integrity of the downloaded firmware image followed by flashing.

The common methods followed to address the first two issues is resending the entire firmware image or repeated restarting of the update process over the air. By adopting these repetitive mechanisms, the time duration to successfully complete the firmware update will extend, although there was only a need for resending partial chunk of binary image. Such extended update durations will cause ill-effects on the target device, the device may consume more energy due to longer active state of radio and this in-turn degrades its performance with respect to battery usage. Long battery life is one of the prominent features of IoT devices [10].

Similar to majority of the data traffic on the Internet, FOTA is also not fully secured. Attack on IoT devices is a potent threat. There are no universal standards for FOTA containers and meta data formats. As there are no standard models for integrity verification in end device, there are chances of attackers tampering the FOTA update process by introducing malicious firmware along with the downloaded image. After flashing this malicious code, the attacker can execute arbitrary actions and virtually control the end device. Therefore, there is a need for trusted computation within the end device.

4. Scope for Research

There is need for research in the area of developing a secure object which can overcome the drawbacks of huge packet loss due network interruption, which in-turn extends update time. Extensive work is required in the field of standardizing the FOTA procedure. This helps to avoid the repeated restarting of FOTA procedure for constrained edge IoT devices. Thus, the task can resume from the point of interruption, which in-turn reduces time and consumes less energy.

Research can be done in the direction of developing a trusted and self-verifying computing base can be designed within the ECUs to verify and validate the integrity of the downloaded binary image. Mechanisms to perform runtime verification of system functionality. To enhance the reliability of the in-vehicle ECUs, virtualization techniques can be incorporated to support a control system, that can check the installation. Local verification within the ECU for newly flashed memory can be developed. Research can be extended to develop recovery actions in case of unsuccessful FOTA updates. Rollback strategy is a solution to this issue.

5. Conclusion

According to the current literature, most of the existing schemes for FOTA update lack with a standardized format or structure. This has led to less secure and unoptimized procedures. In most of the cases, there is no provision for delta update of firmware and the FOTA task fail to resume from the point of interruption. And the existing solutions to these issues are retransmission of entire image and repeated restarting of the update procedure. Instead, research can be carried out to develop a standard secure FOTA framework, which can offer partial update of firmware image and resuming a paused/ abruptly stopped FOTA task due to loss of connectivity.

It was also identified that there is a need to secure the end device itself. FOTA protocols have only IP-based application layer security. Whereas, after the image is downloaded there is no guarantee about the integrity and credibility of the downloaded firmware image, as there are chances of attackers introducing malicious firmware chunks into the ECU itself. Therefore, an extensive research is required in the field of developing a trusted integrity verification and memory verification systems within the ECU itself. Thus, the connected vehicle manufacturers are enabled to verify whether the firmware is securely downloaded and also verify whether correct firmware image is flashed onto the end device.

References

[1] A. Sachdev and M. Bhansali, “Article: Enhancing Cloud Computing Security Using AES Algorithm”, *International Journal of Computer Applications*, vol. 67, no. 9, pp. 19-23, April 2013.

[2] N. Sklavos and I. D. Zaharakis, "Cryptography and Security in Internet of Things (IoT): Models, Schemes and Implementations", 8th I*FIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2016, pp.1-2.

[3] S. S. Basu, S. Tripathy and A. R. Chowdhury, "Design Challenges and Security Issues in the Internet of Things", IEEE *Region 10 Symposium*, May 2015, pp. 90-93.

[4] S. Cirani, G. Ferrari and L. Veltri, "Enforcing Security Mechanisms in the IP-based Internet of Things: An Algorithmic Overview", *Algorithms*, vol. 6, no. 2, pp. 197-226, 2013.

[5] K. Doddapaneni, E. Ever, O. Gemikonakli and L. Mostarda, "Effects of IDSS on the WSNs Lifetime: Evidence for the Need of the New Approaches", TRUSTCOM *‘12, IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, June 2012, pp. 907-912.

[6] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC *6347*, Jan 2012.

[7] Open Mobile Alliance Device Management, *[Online: http//openmobilealliance.org/iot/]*, 2017.

[8] C. A. L. Putera and F. J. Lin, "Incorporating PMA Lightweight M2M Protocol in IoT/M2M Standard Architecture", *IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Dec 2015, pp. 559-564.

[9] Efficient Over-the-Air Software and Firmware Updates for the Internet of Things, http://embedded-computing.com/articles/efficient-software-firmware-updates-the-internet-of-things, 2016.

[10] K. Doddapaneni, E. Ever, O. Gemikonakli, I. Malavolta, L. Mostarda and H. Muccini, "Path Loss Effect on Energy Consumption in a WSN", *Proceedings of the 2012 UKSim 14th International Conference on Modelling and Simulation, ser. UKSIM ‘12 IEEE Computer Society,* 2012, pp. 569-574.

[11] C. B. Z. Shelby and K. Hartke, "The Constarined Application Protocol (CoAP)", *RFC 7252*, 2014.

[12] Krishna Doddapaneni, Ravi Lakkundi, Suhas Rao, Sujay Gururaj Kulkarni and Bhargav Bhat, "Secure FoTA Object for IoT", *IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*, Oct. 2017.