Safety and Security for Medical Devices: Analysis and Implementation of a Secure Software Update for Embedded Systems

Andrei Lorengel [(1)], Jan Pelzl [(2)]

[(1)] Hamm-Lippstadt University of Applied Sciences,
D-59063 Hamm, Germany
E-Mail: andrei.lorengel@stud.hshl.de
Web: www.hshl.de

[(2)] Hamm-Lippstadt University of Applied Sciences,
D-59063 Hamm, Germany
E-Mail: jan.pelzl@hshl.de
Web: www.hshl.de

Abstract – In the recent decades, medical devices developed from stand-alone devices to smart and networked systems. Innovation in medical devices mainly is driven by software. Software increasingly determines the devices' functionality, making it an indispensable part of the product. Remote diagnosis, remote configuration as well as monitoring capabilities are only some examples of the powerful information technology being used within modern medical devices. However, software driven products bear the risk of manipulation - with severe threat to life or physical condition.

With the advent of first attacks on medical devices, first recommendations by officials have been introduced. As an example, the US Food and Drug Administration (FDA) recently published an update of its security guidance document "Post Market Management of Cyber-Security in Medical Devices", containing several security recommendations for the safe and secure operation of medical devices in the field [1]. It is just a matter of time until first obligatory standards will come up for the US and for Europe.

As a consequence of this development, software for medical devices needs to ensure security and, thus, reliability of the entire device. The manufacturer of software must meet all the requirements of the Medical Devices Act when the software used for therapy and diagnosis measures for a human being [2]. On the one hand, this implies respective development processes to develop reliable, safety-critical software. On the other hand, the nature of software allows updates in case of improved functionalities or bug fixes in the field. This powerful feature allows manufacturers to update functionalities even years after production. However, it is imperative to only allow updates with officially released and tested software. It shall not be possible to unintentionally or intentionally update devices with third party software and, thus, allow manipulation.

With this contribution, we will discuss the importance of secure software updates for medical devices and will demonstrate secure software update with the help of modern IT security. From a technical perspective and depending on the development process, software updates can be extremely complex. Small changes might lead to life critical situation after the update process.

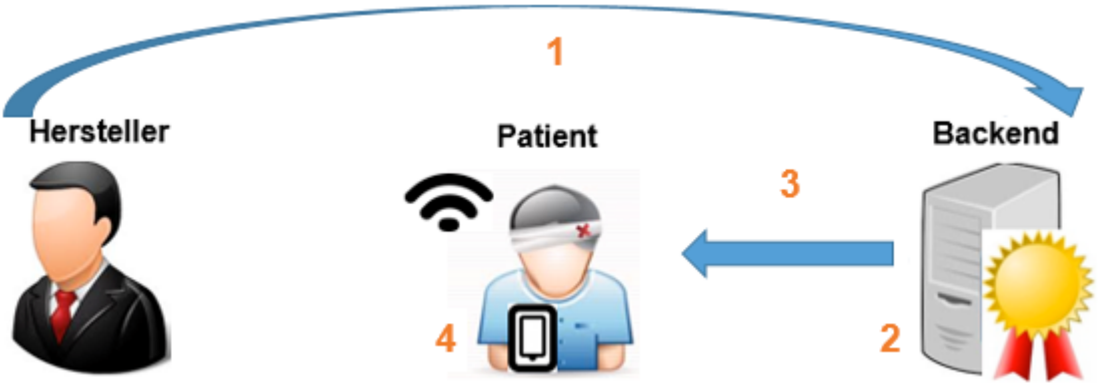

Fig.1: Simplified secure software update of a medical device via a digital interface - all connections are encrypted and authenticated. Step 1: Certificate and signature of software is created by the manufacturer and send to a central server. Step 2: Central server keeps track of software revisions, devices in the field and schedules updates with remote devices. Step 3: Software update is transferred securely. Step 4: Device locally checks the software and performs the update.

Figure 1 shows a simplified principle of a secure software update for medical devices. Cryptographic mechanisms ensure the authenticity of the software and only allow devices to update if the software is from a trusted source. Practically, there are two possible flavors of realization: use of symmetric cryptography or asymmetric cryptography. In both cases, the manufacturer has to use a confidential secret key to generate a certificate of the software by signing it. In the device, the signature has to be verified with the corresponding key. In the symmetric case, the device has to use the same (symmetric) key to verify the signature, a so-called message authentication code. Whereas in the asymmetric case, the device requires the public key of the manufacturer to verify the signature. Both variants do have advantages and disadvantages: Symmetric signatures require a secure storage of the (secret) key in every device but are extremely efficient. Asymmetric signatures require more computational power but therefore only require authentic keys. Depending on the security features of the device and the way of handling keys for devices (e.g., individual keys per device vs. global keys), the security of the realization of a secure software update varies. However, the best choice of algorithms and key management principles is not only a matter of security but also of cost-effectiveness and a good fit to existing production and development processes at the manufacturer site. [3, 4]

With this work, we discuss the tradeoffs of different variants of secure software update and show an example how to achieve secure software updates for a typical embedded linux using digital signatures.

References

[1] U.S. Food and Drug Administration: Postmarket Management of Cybersecurity in Medical Devices, Guidance for Industry and Food and Drug Administration Staff, December 28, 2016. http://www.fda.org

[2] A. Gärtner: Medizinische Netzwerke und Software als Medizinprodukt (Praxiswissen Medizintechnik), 2008

[2] J. Pelzl: IT-Sicherheit für Biomedizintechnik – Typische Anwendungsfälle, Lecture Notes, 2015

[3] Secure Over-the-air Software Updates im Automobil: http://www.all-electronics.de/sota-software-updates-im-automobil/ (accessed Feb. 08, 2016)