UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Xuejun Wu

Security in Remote Update of Medical Devices

Master's Thesis (30 ECTS)

Supervisor(s): Professor Tuomas Aura
Arnis Paršovs, PhD

Advisor(s): Liu Chuang, Msc

Tartu 2022

Security in Remote Update of Medical Devices

Abstract:
As IoT becomes a significant element in many industries, the idea of smart-hospital is emerging. Securing medical devices deployed in health institutions requires regular updates of the medical device's configuration and software. It is essential to build a remote update system for medical devices to ensure the delivery of the updates. This thesis analyzes recent reports on the high-level threats and attacks that target IoT systems in health institutions to derive the security requirements that need to be considered in designing a remote update system for medical devices. Also, we list some remote update technologies that are used in other industries and analyze their extendability and compatibility with the security requirements in the healthcare sector.

Keywords:
remote update, medical devices, threat analysis, security requirements

CERCS: T120, Systems engineering, computer technology

Turvalisus meditsiiniseadmete kaugvärskenduses

Lühikokkuvõte:
Kuna asjade internetist saab paljudes tööstusharudes oluline element, kerkib esile nutika haigla idee. Tervishoiuasutustes kasutatavate meditsiiniseadmete turvalisus nõuab meditsiiniseadme konfiguratsiooni ja tarkvara regulaarset uuendamist. Värskenduste kohaletoimetamise tagamiseks on hädavajalik ehitada meditsiiniseadmetele kaugvärskendussüsteem. Selles lõputöös analüüsitakse hiljutisi aruandeid kõrgetasemeliste ohtude ja rünnakute kohta, mis on suunatud tervishoiuasutuste IoT-süsteemidele, et tuletada turvanõudeid, mida tuleb meditsiiniseadmete kaugvärskendussüsteemi kavandamisel arvestada. Samuti loetleme mõned kaugvärskendustehnoloogiad, mida kasutatakse teistes tööstusharudes, ning analüüsime nende laiendatavust ja ühilduvust tervishoiusektori turvanõuetega.

Võtmesõnad:
kaugvärskendus, meditsiiniseadmed, ohuanalüüs, turvanõuded

CERCS:T120, Süsteemitehnoloogia, arvutitehnoloogia

Preface

I want to thank Professor Tuomas Aura and Professor Arnis Paršovs for their guidance and support in completing this thesis.

ESPOO, 10.03.2022

Xuejun Wu

With the support of the
Erasmus+ Programme
of the European Union

Symbols and abbreviations

Abbreviations

AES	Advanced Encryption Standard
ASP	Application Service Provider
BVN	Bitstream Version Number
CCM	Cipher Block Chaining - Message Authentication Code
CT	Computed Tomography
DDoS	Distributed Denial-of-Service
DoS	Denial-of-Service
DICOM	Digital Imaging and Communications in Medicine
ECG	Electrocardiogram
EMR	Electronic Medical Records
ENISA	The European Union Agency for Cybersecurity
FPGA	Field Programmable Gate Array
HTTPS	Hypertext Transfer Protocol Secure
ICD	Implantable Cardioverter Defibrillator
IoT	Internet of Things
JTAG	Joint Test Action Group
LAN	Local Area Network
LPWAN	Low-power Wide-area netwrok
MEC	Mobile Edge Computing
MQTT	Message Queuing Telemetry Transport
MRI	Magnetic resonance imaging
NAT	Network Address Translation
NVM	Non-volatile Memory
PACS	Picture Archiving and Communication System
RAM	Random-access Memory
REST	Representational State Transfer
RFID	Radio-frequency Identification
ROCE	Remote On-demand Code Execution
ROM	Read Only Memory
SARFUM	Security Architecture for Remote FPGA Update and Monitoring
SBM	Secure Bitstream Manager
SCADA	Supervisory Control and Data Acquisition
SHA2	Secure Hash Algorithm 2
SRUP	The Secure Remote Update Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UART	Universal Asynchronous Receiver/Transmitter
URL	Uniform Resource Locator
UUID	Universal Unique Identifier
VLAN	Virtual Local Area Network
WAN	Wireless Area Network
WLAN	Wireless Local Area Network
HART	Highway Addressable Remote Transducer Protocol

Contents

Preface **3**

Symbols and abbreviations **4**

1 Introduction **6**

2 System Architecture **7**
2.1 PACS System 7
2.2 Honeypot Healthcare Institution system by TrapX 7
2.3 Target Architecture 8

3 Threat analysis **10**
3.1 Assets to protect 10
3.2 Potential attackers and attack vectors 11
3.3 Security Challenges 12
3.4 High-level Threats and Vulnerabilities 13
3.5 Vulnerabilities and attacks 15
3.5.1 Vulnerability taxonomy 16
3.5.2 Attacks on remote healthcare devices 17
3.5.3 Attacks on clinical IoT devices 19
3.6 Threat prioritization 20
3.6.1 Threat assessment model 20
3.6.2 Threat prioritization 22
3.7 Security requirements 24
3.7.1 Remote healthcare system 24
3.7.2 Clinical IoT device 26

4 Survey of existing remote update technologies **30**
4.1 SRUP: The Secure Remote Update Protocol 30
4.1.1 Protocol details 30
4.1.2 Enhancements of original protocol 31
4.2 Remote software update in LPWAN 33
4.2.1 ROCE 34
4.2.2 Software update management 35
4.3 SARFUM 35

5 Evaluation of existing remote update technologies **38**
5.1 SRUP 38
5.2 Remote software update in LPWAN 39
5.3 SARFUM 40

6 Summary **42**

References **44**

Licence **45**

1 Introduction

Internet of Things (IoT) is a system that establishes a connection between different devices through the internet. With the development of computing and communication technologies in the last decades, IoT becomes a significant element in many industries, including healthcare. As IoT systems are more ubiquitous in healthcare systems, healthcare institutions can collect, analyze, and exchange data remotely. Digital records also decrease the chance of data loss.

On the other hand, the security of IoT devices is a big obstacle in practice. According to recently published reports [eni15][med15][med16], securing devices in healthcare is facing more challenges compared to other industries.

First, many medical devices, such as life support equipment, must operate almost all year round. As a result, resolving security issues is usually delayed. Commonly, devices that are compromised by adversaries still operate for a long time even if the hospital has suspected the possible intrusion. That is because shutting down the devices may cause damage to patients and affect hospital operation. Many hospital thinks shutting down devices lead to greater risks to hospital comparing to the possible intrusion from adversaries.

Second, many medical devices must go through complex approval of different administrations to ensure the product's safety. These devices are mostly closed systems. The device can only be updated or configured by the manufacturer. As a result, it is hard for the IT departement in health institution to secure the device. The most common practice nowadays is to install the devices behind the firewall. However, according to the research of TrapX [med16], the attacker can defeat this strategy in all their case studies. That is because many medical devices run out-of-date operating systems. In one case study, the attacker camouflaged sophisticated tools inside the MS08-067 worm to exploit devices with an older version of Windows. Since Windows 7 and later versions are not vulnerable to this worm, many security solutions choose to ignore the alert even when this kind of attack is detected.

As a result, it is essential to have a secured method to update software in medical devices remotely from manufacturer's server. For manufacturers, the cost of sending technicians to customers to perform regular updates is too high. With remote update systems, the manufacturer can perform regular updates of end devices at a higher frequency and much lower costs. There are many existing solutions for remote updates of IoT devices. However, the requirements are more sophisticated when it comes to medical devices. Medical devices usually contain sensitive information, and update failure will have much more catastrophic consequences. This thesis focuses on existing remote update techniques and assesses their practicability in medical devices.

The main contribution of this thesis is demonstrating security requirements in the remote update of IoT devices in the medical industry and evaluating existing remote update architectures and techniques against the security requirements we derive. We derive the security requirements by analyzing the high-level threats and security challenges mentioned in several reports with verified attack scenarios. Then, we evaluate three remote update techniques to check whether they are compatible with the security requirements.

The thesis is organized as follows. Section 2 describes the system architecture we are targeting. Section 3 describes the threat analysis of the targeted system and the security requirements for different components. Section 4 describes existing technologies for secure updates. Section 5 evaluates the current technologies against the security requirements of related components. Finally, Section 6 gives the conclusion.

2 System Architecture

Most healthcare institutions have multiple IoT systems that have different types of devices. To analyze the security requirements of remote update systems in the healthcare industry, we first need to understand what types of devices generally exist in health institutions. We first demostrates existing components in healthcare IoT systems before discussing the possible threats and security requirements of medical devices. This section lists system architectures of two different healthcare systems. A target architecture, which includes devices that commonly exists in most healthcare IoT systems, is built based on the existing examples. The threat analysis and reality check on the existing remote update methodology can be formed targeting the components mentioned in target architecture.

2.1 PACS System

This architecture is from the security analysis of medical devices produced by TrapX [med16]. The PACS subsystem is a subsystem of a top 2000 global hospitals. The hospital has malware detection software and gateway firewalls with a security strategy specifically designed to target the operating systems used in the hospital. Also, they have internal firewalls that divide networks into different segments. The PACS is one of the segments. Last, they have endpoint security installed to protect internal clients. Figure 1 depicts the architecture of the system.

The communication protocol used in PACS is DICOM, an international standard to transmit, store, print, and display information in medical imaging applications. This protocol can handle images, but also files of other types, such as PDFs.

The PACS systems include CT scan imaging, x-ray film imaging, MRI imaging, and Ultrasound. The information collected from the medical instruments is gathered on PACS servers. Users can access those images from internal clients and external clients. The internal clients, such as workstations that are installed in the hospital, are protected with endpoint security software. Since the system architecture is not the focus of TrapX, remote clients are not mentioned in the paper. It is not clear whether the remote clients include patients' personal devices.

2.2 Honeypot Healthcare Institution system by TrapX

TrapX is a cyber deception platform. It creates authentic traps to create targets for attackers. When the traps are attacked, they trigger alerts. By monitoring the trap, the anatomy of the attack and the methodology used by the attackers can be detected.

Figure 2 depicts the healthcare network system created by TrapX. It includes websites and functional IT infrastructure. The User VLAN connects desktop computers. The Medical VLAN connects the blood gas analyzers, EMR, PACS, CT scanners, and X-rays. The Wireless VLAN connects wireless devices, tablets, and smartphones. The SCADA VLAN connects devices that monitor patients' status and lab results. TrapX also installed gateway firewalls and anti-virus web filters in internal clients.

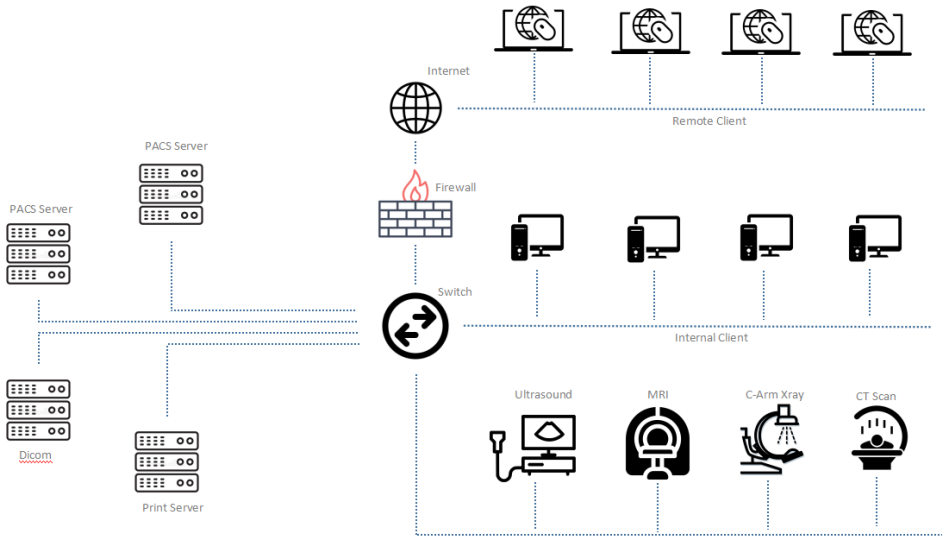

Figure 1. PACS

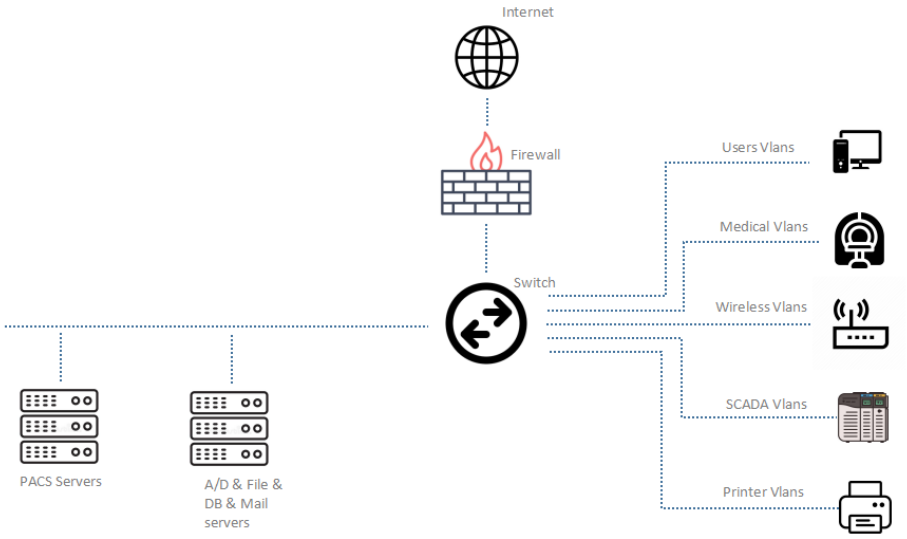

Figure 2. Honeypot medical system architecture

2.3 Target Architecture

Based on the system architectures above, the following target architecture is created. Figure 3 depicts the target system architecture. The target architecture can be divided into two sections: The first section is remote healthcare IoT system; The second section is clinical IoT system.

The components in clinical IoT systems include medical devices, wireless devices, and internal clients. The **medical devices** include devices inside hospitals, such as CT scanners, X-rays, and ECG instruments. Depending on the type of medical device, the computing

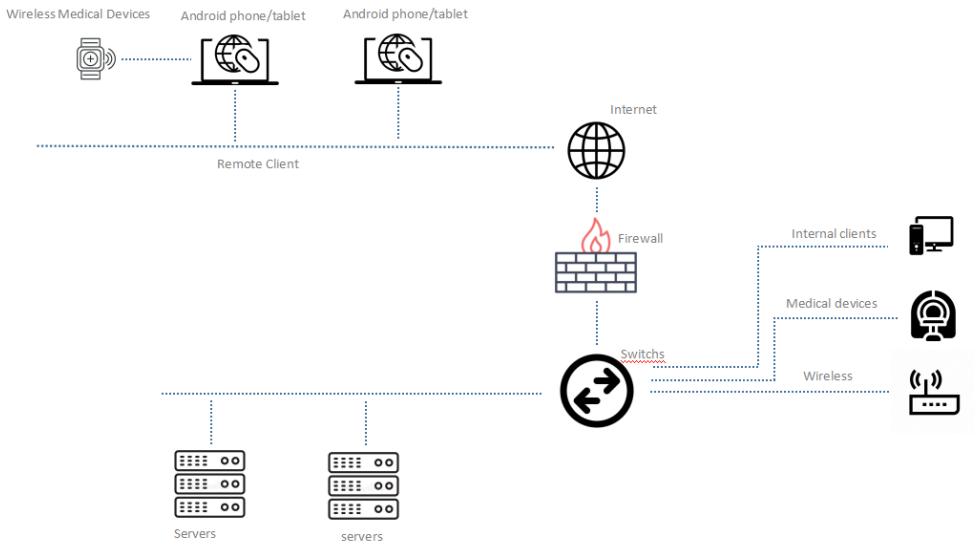

Figure 3. Outline of the target system architecture

power and storage available are different. Some devices are connected to workstations. Some upload data to the server through wireless network devices directly. The **wireless devices** are usually used for hospital managements, such as smart camera, smart light and RFID reader. Those devices are common targets of adversaries, since the wireless devices usually have week end point security and communicate with systems that contains patients' information. The **internal clients** are clients that helps hospital staffs in operating medical devices and accessing patients' information. The internal clients can be tablets, laptops and desktops that operates inside hospital network.

The components in remote healthcare IoT system includes wireless medical devices and remote clients. **Wireless Medical Devices** include wearable devices and implantable devices. Most wireless medical devices have limited computing power and storage. Wireless devices usually connect to remote clients through WiFi or Bluetooth. **Remote Clients** acts as an intermediate level between the medical devices and server. The user could be the staff at a healthcare institution, such as nurses or doctors. The user could also be patients for household medical devices or implantable devices. The connection between remote clients and servers is on-demand over the public internet. It means that the communication channel is not secured and the remote clients are not always online. Remote clients may contain sensitive information depending on their function.

Hospital Server is related to both clinical IoT systems and remote healthcare IoT systems. It is managed by admins, such as the cyberdefense team in a healthcare institution or technical staff from the manufacturer. The server remotely monitors the technical status of devices and stores sensitive information about patients. The server is usually under a controlled environment and is less likely to be compromised. Since the server's security is not the focus of this thesis, this component will be considered trusted in this thesis.

3 Threat analysis

This section demonstrates security requirements derived from the threats and real-world attack scenarios in the healthcare industry. We survey existing reports on security challenges and security incidents in the healthcare industry. Then we prioritize the attacks and threats based on their impact and probability of exploitation. Based on the threat prioritization, we derive the security requirements targeting the common components mentioned in the target architecture.

3.1 Assets to protect

In many recent attacks on health institutions, attackers usually use medical devices as an entry point. [Edu15] Since many devices are interconnected with each other, attackers can penetrate the healthcare network, establish backdoors within different medical devices, and find out attack paths to compromise clinical information systems. This section will focus on common assets in IoT-enabled health institutions. [eni17]

- **Remote healthcare assets** allow the hospital to provide healthcare services remotely. It dramatically extends the range of services from the hospital. Devices for monitoring, diagnosing, and distributing drugs are considered remote healthcare assets. Examples of such assets are equipment for blood pressure and heart rate measurements, automated dosing equipment, threshold-triggered alarm, implantable medical devices, and others. Remote healthcare assets usually operate outside the hospital, making them difficult to secure. The attacker can target the device itself or the network in the patient's home.

- **Networked medical devices** are common assets of the health institution. They allow the hospital to integrate information into internal servers to perform data analysis and persistence. Some examples of networked medical devices are CT scanners, life support machines, assistive robots, etc. These devices usually operate inside the hospital network. The availability of such devices is crucial to the operation of health institutions.

- **Networking equipment** ensures the connectivity of devices in a health institution. The requirements of such equipment are similar to the networking equipment used in a traditional hospital, but with advanced features, such as bandwidth and routing protocols. Examples of such equipment are switches, routers, IoT gateways, etc.

- **Identification systems** are used to manage patients, staff, and hospital equipment. The identification systems are also connected with different devices and information systems. Since the unavailability of an identification system will have a critical impact on hospital operation, it can become the target of attackers. One example of an identification system is the RFID system for locating patients and hospital equipment.

- **Client devices** are used in the hospital to provide doctors, nurses, and patients with information. Examples are desktops, laptops, tablets, pagers, etc. Client devices are usually the secondary target for attackers since these devices usually run much more up-to-date operating systems. Usually, these devices are compromised by connecting to infected medical devices.

- **Servers** are the backbone of all hospital systems. Thus, servers are under the protection of multiple firewalls and are strictly monitored by the security team of the health institution. They contain sensitive patient information and impact all systems related to hospital operations.

- **Data** are considered essential assets from the perspective of information security. Loss of data can stall hospital operations. There are many different types of data, such as clinical data, financial and operational data, research data, staff data, etc. There are many ways an attacker can generate profit by selling, editing, and encrypting existing data. Thus, it is important to ensure the integrity, confidentiality, and availability of sensitive data.

- **Application logic** is the logic of installed applications in the servers and medical devices. The application logic is vital for both health institutions and medical devices manufacturers. If attackers can modify the internal logic or find exploitable vulnerabilities of the manufacturer's product, all users of that product are at risk.

3.2 Potential attackers and attack vectors

This section lists different threat actors that can be potential attackers. They will have different attack vectors targeting a health institution based on their types. Also, we will list different attack vectors in a health institution. The result of this section is based on the research of ENISA. [eni17]

Potential attackers include:

- **Insider threat**: Examples of such threats are doctors, nurses, maintenance staff, etc. Since insiders have different privileges in the systems, they have a much broader attack surface than other attackers.

- **Malicious patients or guests**: Visitors to the hospital (primarily patients) are also potential attackers. They usually have limited access to assets in a health institution. Since they can physically interact with assets in a health institution, their malicious intent can also have a significant impact.

- **Remote attackers**: Nowadays, many health institutions provide remote care services. That means the network in a health institution is not a closed network. Hospital servers provide information services to patients and gather data from remote medical devices that operate in patients' homes. It is possible for an attacker to remotely compromise information systems in the hospital, such as laboratory information systems, radiology information systems, picture archiving and communication systems, etc. Also, the remote attackers can target remote healthcare devices and network at patients' homes to gather sensitive information.

- **Other causes**: Examples of such threats are the failure of software and equipment. Sometimes, security incidents can happen because of environmental factors or human errors.

Attack vectors include:

- **Physical interaction**: Attackers can directly interact with devices they have access to. Even an outsider has a chance to have access to medical devices by pretending to be a patient. In [SKP+18], the physical access to a device is distinguished into two access levels. An insider has direct physical access to the target IoT devices. An outsider does not have physical proximity to the target device. However, an outsider may gain knowledge of the target device by tampering other another device of the same type (e.g. extracting a common pre-shared key).

- **Wireless communication**: It is common for medical devices to upload data to internal clients through a wireless network. By intercepting the communication of the identification system or mobile devices, adversaries can access patients' data.

- **Wired communication**: Wired communication is related to online health care information systems or cloud services, such as drug inventory and patient healthcare records. Adversaries can target the network by remotely compromising devices inside the network or by accessing the network infrastructure.

- **Interaction with hospital staff**: Social engineering is one of the most common techniques adversaries use. Instead of targeting the information system, the adversaries target hospital staff who have privileged access. Adversaries can acquire patient data and implant ransomware inside the hospital network by fooling or convincing the staff to send a command or carry out tasks.

3.3 Security Challenges

Health institution faces many security challenges. The majority of the challenges are common to other critical infrastructures. Based on the research conducted by ENISA [eni15], the following are the most critical security challenges for healthcare institutions that are related to this thesis' subject.

- **System availability**: System availability is essential for the healthcare industry. Without continuous accessibility of critical systems, the authorized professionals cannot ensure the quality of healthcare services. Not only will the malfunction of medical devices endanger patient health, but a failure of the patient information system can also misguide physicians or nurses to inject incorrect drugs or make treatment plans without all necessary information about a patient, which leads to physical damage to patients.

- **Lack of interoperability**: Smart hospitals include many different systems that are interconnected with each other. The security strategy of one system will be affected by the situation of other related systems. It means the health institution needs to have an agreed-upon framework, protocols to secure information exchange, and up-to-date operating systems as the backbone to support security updates from manufacturers.

- **Access control and authentication**: According to a recent study performed by KPMG [KPM15], many data security incidents are caused by sharing data between third parties and insiders. That shows that authentication and access control is especially are critical. Authentication systems should be deployed to identify the identity

of patients and staff inside the hospital. Once the identity is verified, the access control policy can define each identity's information level and privilege. Access control is one of the most vital systems to ensure data privacy and integrity.

- **Network security:** Many modern medical devices are connected to the network, which brings new challenges to the healthcare industry. It is problematic to secure the hospital network, which contains old medical devices, which run old operating systems and are not designed to be connected to the internet, is problematic.

- **Data loss**: The progress of digitizing patient records means personal confidential information is stored on servers nowadays. Ensuring the integrity, confidentiality, and accessibility of patient data is essential. Thus, countermeasures for system failure, network faults, security attacks, and natural disasters are important. The hospital should have the ability to achieve data recovery in those situations quickly.

3.4 High-level Threats and Vulnerabilities

The use of IoT in the healthcare industry faces specific challenges. The report of ENISA [eni17] listed several emerging vulnerabilities health institutions are facing. In this section, the vulnerabilities that need to be considered in designing remote update systems will be listed. The list of vulnerabilities includes both technical aspects and social aspects.

- **Interconnection of IoT devices**: The IoT devices are interconnected with each other. Some devices even can connect to other devices automatically. Thus, the security decision made for one system can have a global impact. Many medical devices were designed to operate in an isolated internal network in the healthcare industry. However, to ensure the operation of smart devices nowadays, hospitals have to allow those devices to communicate with old hospital systems and have access to the internet. That gives adversaries extra attack surfaces to gain access inside the internal network.

- **Physical security**: Since hospitals are crucial public infrastructure, medical IoT devices are exposed inside public places without strict access control. It means that securing the physical perimeter of all kinds of medical IoT devices is impossible. However, many medical devices do not have countermeasures to physical tampering. Here are some of the possible design or implementation flaws, which are mentioned in reference [SKP+18].

    1. **Lack of tamper resistance**: The device may not have hardware security control detecting physical tampering (e.g. key extraction attack).
    2. **Weak hardware implementation**: Untested hardware implementation may leak sensitive information. The adversary may extract stored keys through a side-channel attack [RSWO17].

- **Firmware security**: If the integrity of firmware is compromised, an adversary can modify and re-install the modified firmware into the device. If the full image of firmware is leaked, adversaries can reverse engineer it and recover the stored credentials to compromise the target device [RSWO17].

- **Network vulnerabilities**: The protocols used in the network are one of the major targets of adversaries. The adversary can target the communication protocol or the key management mechanisms.

    1. **Communication protocol**: The vulnerabilities can appear on the network layer and application layer protocols. Some of the network protocol families often used in IoT systems, including ZigBee, WirelessHART, and WiFi, have several security flaws that adversaries can utilize. [ML16][Lom15] On the other hand, misconfiguration and implementation flaws in application-layer protocols can also lead to security incidents. [RS16]
    2. **Key management**: Because of the limitation of the devices, some devices use a common key for all the devices of the same model. This enables an attacker to extract the key from one device and easily attack all the devices. Some devices have simple key management schemes in place. However, the key for authentication can be easily extracted in some cases [Ede16].
    3. **Weak crypto algorithms**: Limited by the computation power and storage, the crypto algorithm implemented may be too weak. That gives the attacker the opportunity to decrypt and forge the data by intercepting the package in the network.

- **Abusive use of medical devices**: Most devices are only built for the intended use, which assumes users are not malicious and are well informed. However, by abusively using the devices, the adversaries or users who are not familiar with the device can leak sensitive information or harm patients. That leads to vulnerabilities and risks inside the hospital network.

- **Mass-scale deployment of homogeneous IoT devices**: Just like IoT systems in other industries, there is a mass-scale deployment of homogeneous devices. This is a potential attack path for adversaries. For example, they could use an infected device like an amplifier to perform DDoS attacks. [Ste17] The adversaries only need to find one attack path, but companies need to remove all vulnerabilities. Since it is impossible to fix all vulnerabilities, it is important to assume some of the devices are compromised already and assess how will that impact the security of the critical infrastructure.

- **Life span of medical devices**: For medical devices, the life span is much longer compared to devices in other industries. The CT and MRI machines will not be changed in less than three years. Even if the hospital changes the device, the device is still outdated when considered from the IT point of view. The testing and production of devices take almost three years according to EU legislation. The same situation applies to modern hospitals since the smart IoT components are built based on old infrastructure.

- **Endpoint security of medical devices**: Because of the limited storage resources and processing power, the IoT devices have little malware detection and prevention capability. Without remote update methods, it is nearly impossible to upgrade massively-deployed homogeneous devices.

- **Unvisible internal functioning of IoT devices**: For security reasons, the manufacturer of medical devices does not publish the internal functioning of devices and data streams they produce. On one hand, it increases the difficulty for adversaries to compromise the devices. On the other hand, the cyber defense team is not informed about the potential threats and risk decisions made by manufacturers, which makes it impossible for the hospital to formulate security strategies targeting specific devices.
- **Uninformed breaches**: Without a proper way to alert users about the security incidents, the security breach can persist for a long period before being detected. The compromised medical devices are footholds for adversaries for further reconnaissance. Traditionally, the action for handling security incidents is denying the services. However, this approach does not work in a hospital, since the close of services endangers patient safety even more than the attack of adversaries.
- **Unauthorized control**: Without a proper access control system, unauthorized users may gain access to critical systems through endpoint devices. The authentication and authorization of staff can help prevent this situation. The "need-to-know" basis and understanding of the cyber security perspective are missing in some access control policies in hospitals.
- **Ignoring security measures**: Sometimes the security measures are ignored by staff because the policies are considered unnecessarily inconvenient. Sometimes, the security measures are circumvented simply because of time pressure or because it conflicts with other objectives, such as patient experience or personal privacy.
- **Use of personal devices**: In a smart hospital, the use of personal devices can lead to extra vulnerabilities. However, it is not practical to not allow physicians or patients to use mobile or wearables. On one hand, a strict policy is necessary for data security. On the other hand, these policies can lead to complaints from patients and staff. In many cases, the IT staff is not even aware that such devices are being used and are connected to the hospital network.
- **Unqualified medical devices**: In this case, the business introduces too many new devices to the system, which gives the IT department little time to test those devices properly to check whether they comply with the security requirement of the hospital. The asset addition rate often exceeds the capability of the IT department to follow proper procedures for security checks. Thus, some mote devices (such as CCTV), may be deployed without careful evaluation.
- **User behavior**: User behavior is always an aspect of concern in security assessment. The adversaries can use the habit of staff and patients to find vulnerabilities that are not visible to the cyber security team in the hospital. It is important to assess how deeply is hospital staff involved in the system. If the system needs support from many security policies and human operations, the adversaries can find vulnerabilities by misguiding users or utilizing workaround solutions of hospital staff.

3.5 Vulnerabilities and attacks

This section lists vulnerabilities that directly cause security incidents. We first list common vulnerabilities and threats in medical IoT systems. Then, we describe different attack

scenarios targeting each component in our target system architecture.

3.5.1 Vulnerability taxonomy

1. **Malicious actions**: Malicious actions are intentional attacks from adversaries. This category does not include actions that work around hospital policies and regulations without malicious intention. The malicious action can be deliberated internally and externally.

    - **Malware**: Malware is a major threat to health intuition. Depending on the type of organization it targets, a large number of organizations can be attacked with low effort. The type of malware includes ransomware, worms, trojans, viruses, rootkits, exploit kits, botnets and spyware. Through the coordination of different types of malware, hijacking can be performed at the network level and generate profit by accessing personal information, acquiring prescriptive drugs, or ransomware attacks.
    - **Device tampering**: Since the medical devices are not resource-limited, the device may be reprogrammed or reconfigured. It is possible for adversaries to tamper medical devices without physical access to the device. Sometimes, tampering devices without malicious intention may result in catastrophic failure. Also, by tampering Non-medical IoT devices inside the hospital network, the adversaries can initiate attacks that target internal clients and medical devices.
    - **Attacks that target protocol:** Many IoT systems have specifically designed application protocols to establish communication between devices. However, the protocol may have vulnerabilities that adversaries can utilize to spoof information and alter device configuration.
    - **Reverse engineering:** Reverse Engineering is usually the last solution to penetrate a system for the adversaries since it takes time to understand the internal logic of the system and device. But it is also a powerful method that adversaries can use. By targeting IoT devices, the adversaries can attack homogeneous medical devices with an established attack path through reverse engineering.
    - **Device and data theft**: Depending on the volume of the devices, the threat level of this kind of attack is different. However, with the implementation of RFID authentication systems and facial recognition, the number of sensors and CCTV is increasing rapidly. Thus, the likelihood of such an attack is rising rapidly.
    - **Social engineering attack**: Common social engineering attackers include phishing and baiting that misleads hospital staff to send commands or alter devices.
    - **Skimming**: Skimming targets RFID tokens specifically. Since RFID tags are used more often in the authentication system, it is important to have countermeasures against such attacks.
    - **DoS attacks**: Since hospitals rely on web or cloud resources, attackers can perform DoS attacks on hospital services. This may disrupt information transit between hospitals and affect the patient care process.

2. **Human errors**: During the configuration and operation of devices, human errors can occur. Human errors are often related to limited human resources, inappropriate processes, or insufficient training.

- **Configuration errors**: When medical systems or medical devices are not configured properly, the attack surface is extended and the operation of such devices may endanger patients.
- **Absence of audit logs**: Audit logs are important for both maintenance of systems and devices and appropriate incident management. When audit logs are missing, it is hard to track the originality of the security incidents and to take corrective actions.
- **Non-compliance**: Non-compliance is especially pertinent for hospitals that rely on mobile applications that can be installed on personal devices. This can result in the leak of sensitive data since personal devices are not in a controlled environment.
- **Physician or patient errors**: When a hospital heavily depends on IT assets and users are not IT experts, such errors might occur. Such errors may be the result of poor concentration due to long working hours, or workarounds due to complex and inefficient policies and procedures.

3. **System failures**: With the increasing complexity of systems, it is important to ensure the normal operation and recovery of the system when part of the system fails. The system can fail for many reasons, including software failures, damaged firmware, network components failure, or overload of the system. The failure of devices, which are not designed to be used as IoT devices, can happen much more often compared to other devices that are designed more recently.

4. **Supply chain failure**: Supply chain failure is not under the control of the affected organizations. As hospitals are increasingly dependent on third-party services, a hospital needs to have emergency countermeasures that handle third-party failure. Examples of third parties that hospitals might rely on are cloud services providers, medical device manufacturers, network providers, etc.

3.5.2 Attacks on remote healthcare devices

The remote healthcare devices are mostly wireless programmable devices. The implantable devices have constrained resources and can only be updated remotely with a programmer in close range. The wearable devices can have a communication channel with a range of up to 60 m. The wearable medical devices transmit data to a nearby process unit (e.g. smartphone, computer) for the initial process. Then, the processing unit transmits data to an advanced process platform, which belongs to a healthcare institution, for further analysis. [MMD17] The attack on remote healthcare devices is based on vulnerabilities of the devices and the patient's home monitoring network.

Here are attacks that target wireless medical devices:

Reverse engineering of protocol Attack targeting the remote devices sometimes target the communication protocol. Marin et al. [MSG+16] demonstrated an attack that can exploit Implantable Cardioverter Defibrillators (ICDs) remotely. For the latest ICDs, the programming header first activates the device into reprogramming mode through a short-range communication channel (less than 10cm). Then, the device can be reprogrammed through

a long-range communication channel (2-5m). Through reverse engineering techniques on the proprietary network protocol, attackers can circumvent the short-range communication to activate the ICD into reprogramming mode. Also, several weaknesses in an application protocol are found, which allow the adversaries to retrieve sensitive patient data, drain the ICD's battery, and send arbitrary commands to the device.

Replay attack Replay attack is common technique adversaries use to exploit IoT devices. By replaying or delaying data transmission, adversaries can fool the device to behave abnormally. Radcliffe [Rad11] suggested a possible attack path targeting the insulin pump. By replaying messages from continuous glucose monitors and jamming the actual message, the adversary can cause the insulin pump to indicate higher sugar readings.

Configuration tampering Tampering the configuration of remote medical devices can be surprisingly easy. According to Radcliffe [Rad11], the wireless peripheral needed to communicate with the insulin pump can be purchased without the need for a prescription. The command codes of insulin pumps are published on multiple websites even though the manufacturer does not disclose them. By changing the setting of an insulin pump, the device can have a potentially deadly effect on the patient (e.g. change the amount of insulin injected per grams of carbon hydrates consumed).

Reprogramming Physician programmers can reprogram implantable devices with simple/no authentication. As a result, with access to a physician programmer, adversaries can program any supported implantable devices. Also, some physician programmers and home monitoring devices use similar embedded radio circuitry. It is potentially possible to leverage home monitoring devices to reprogram implantable devices. [Rad11]

Here are attacks that target custom made home monitoring devices based on micro-controller or FPGA:

Physical tampering The adversary may be able to physically tamper the device to initiate attacks. The embedded devices commonly provide debugging functionality (e.g. JTAG, UART), which allows adversaries to gain privileged access to home monitoring devices. Sometimes, the debugging interface can be accessed through a PC serial port or USB.

Reverse engineering Since the home monitoring device does not operate in a controlled environment, the adversaries can easily have access to the devices. Some device does not have obfuscation and encryption of firmware, which allows the adversaries to easily identify critical coding section through function names, software debugging symbols, and source code comments. [RB17]

Credential and infrastructure data leakage Sometimes, hard-coded credentials and infrastructure data are used in embedded devices. The adversaries can use hard-coded credentials to authenticate to the patient support network. Also, the adversaries can use infrastructure data (e.g. phone number, IP address) to identify the authentication servers for the patient support network.

Corrupted firmware In the current remote firmware update architecture, home monitoring devices do not necessarily validate the source of the update package. As a result, adversaries can perform a man-in-the-middle attack and issue compromised firmware.

Unencrypted data Adversaries have multiple means to extract file systems from home monitoring devices. Since the scrapping process of a home monitoring device is uncontrollable, it is hard to secure data once the device is discarded. Also, adversaries can extract files by mounting removable media. [RB17] Without the protection of encryption, the cost for an adversary to retrieve patient data is trivial.

3.5.3 Attacks on clinical IoT devices

IoT devices operating in a hospital are protected in a network behind a firewall. Usually, the hospital has different systems to support hospital operations. The devices in different systems are indirectly interconnected with each other. Some systems, which require internet access, need to communicate with old devices that are not designed to be exposed to public network. That is one of the major causes of security breaches in modern hospital.

Camouflaged malware In 2016, TrapX Labs released a report that shared their research in attacks that target smart hospital. [med16] These attacks target medical devices which run old operating systems and operate inside the hospital network.

The attack used a variant of an old malware, such as a variant of the MS08-067 worm. Since recent operating systems are not vulnerable to this kind of worm anymore, the worm is not alerted by the endpoint security software and firewall of the hospital. However, since the medical devices use old operating systems and the IT staff are not aware of this situation, the adversaries can compromise the system of old medical devices. Since no security strategy targeting this kind of vulnerability, the attack remains undetected. The adversaries intentionally embed new tools with the worm and started their campaign after penetrating the firewall.

Most hospitals use a firewall to protect medical devices and hospital internal networks.However, the firewall cannot offer enough protection for medical devices. Since the internal software operations of medical devices are not visible, the cyber defense team cannot set up a firewall targeting the vulnerability of each medical device. Thus, attackers and their malware have defeated the strategy of the firewall in many health institutions according to the research of TrapX.

Medical device hijacking After discovering the security challenges, TrapX conducted a cyber-deception operation by creating a fake healthcare provider network system. [med15] The goal of this research is to analyze the tactics of attackers and learn their techniques once they successfully compromise some medical devices. The following is the anatomy of a common medical device hijack attack.

- **Stage 1**: Attacker researches target, find one or more attack paths to penetrate firewall of the hospital.
- **Stage 2**: Attacker gains access to devices inside the hospital network and starts to escalate privileges to gain control of more devices and systems.

- **Stage 3**: Attacker gains access to internal client or server, which allows them to extract confidential patient data and financial records.
- **Stage 4**: Attacker leaves ransomware inside hospital network to extract funds from healthcare institution after cleaning up the traces.

TrapX found that when a network is penetrated, remediation of the medical device cannot help recover compromised medical devices. Commonly, more than one medical device is compromised after a successful attack. The only way to clean up the network is to shut down all the devices and remediate the devices at the same time. If only part of the devices is cleaned up, most of the devices are infected again as soon as they are back online. This leaves the hospital with few options since shut down of hospital operations causes greater loss compared to the ransom.

Attacks that target internal client Sometimes, the adversaries can penetrate internal clients easily with social engineering attacks. The independent security evaluators attempted to gain a foothold in the internal client by randomly dropping 18 USB sticks, which contains simulated malware inside the hospital [Ind16]. After 24 hours, the evaluators successfully compromised a device at the nurse station, which enables them to harvest account physicians and nurse that uses that device.

Attacks that target non-medical devices Sometimes the attack can target non-medical devices that are inside the hospital network. In this attack scenario [Ind16], a vulnerable vendor kiosk is utilized. The research team first compromised the vendor kiosk physically. Then, the internal network was scanned and several exploitable mobile computer stations were identified. At last, through those compromised mobile computer station, the research team were able to access a bloodwork barcode scanning device. The device contains sensitive information, such as patient names and identification information. Also, the adversaries can manipulate the output of the device to change the identification number, contaminating samples and causing inappropriate treatment in real attack scenarios.

3.6 Threat prioritization

Because of the limitation of technology, budget, and human factors, it is unrealistic to mitigate all possible threats with one system design. It is important to find a balance between security and efficiency in system evaluation. However, it is hard to define the cruciality of threats and vulnerabilities without their relationship with actual attack scenarios. Thus, we will categorize the threats and vulnerabilities based on the attack scenarios mentioned in sections 3.5.2 and 3.5.3. Then we prioritize the attack scenarios with the threat assessment model.

3.6.1 Threat assessment model

In this section, we discuss the model this thesis uses to assess the severity of possible attacks. We will use the model used in [SKP+18] as the baseline. We will use different parameters to assess the probability and impact of each threat. Figure 4 demonstrates the structure of the assessment model.

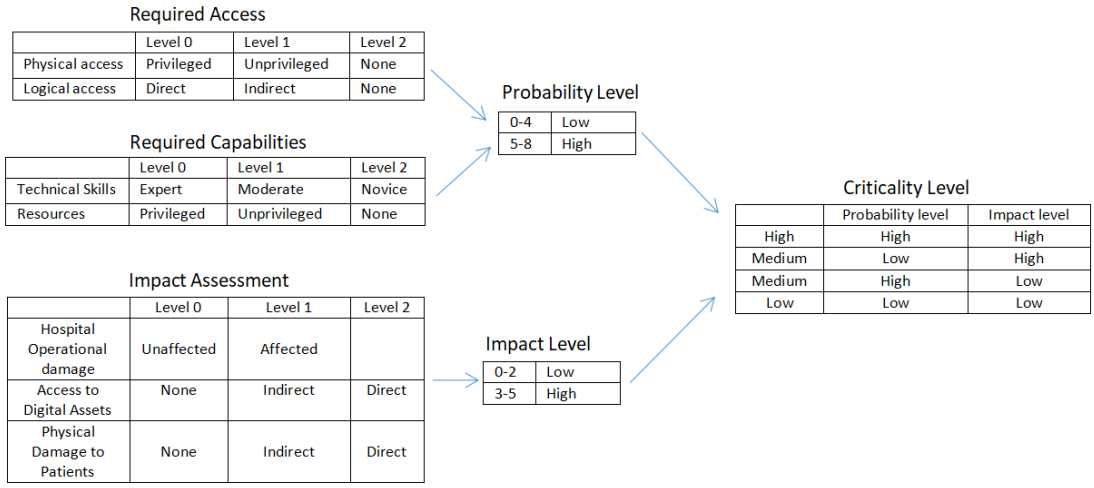

Figure 4. Threat Prioritization

We use the following parameters to assess the probability of each attack. With a higher probability level, the adversaries are more likely to initiate the attack targeting the threat.

1. **Required access to the IoT**: This examines the level of access that is required to trigger the attack. There are two types of accesses. One is physical access, the other is logical access.

   (a) **Physical access** can be distinguished into three access levels. *Privileged access* (Level 0) means the attack requires physical proximity to the target device and it is hard for adversaries to have physical access to the medical device. *Unprivileged access* (Level 1) means the attack requires physical proximity and the device is easy to access. *None* (Level 2) means the attack does not require physical access.

   (b) **Logical access** can be distinguished into three access levels. *Direct access* (Level 0) means the attack requires an available interface to logically connect to the target device. *Indirect access* (Level 1) means the attack requires logical access to other devices (Not including devices adversaries purchase to initiate the attack) that locates in the same network environment to communicate with the target device. *None* (Level 2) means the attack does not require logical access to devices to initiate the attack.

2. **Required capabilities**: This describes the required skill level and amount of resources to enable the attack.

   (a) **Technical skills**: The technical skills required to enable the attack is divided into three level: level 2 as *novice*, level 1 as *moderate*, and level 0 as *expert*.

   (b) **Resources**: The resources are another factor to consider. If an attack can only be enabled with specific devices, such an attack requires high resources and is less likely to happen. We divide the required resources into three levels. *privileged device* (Level 0) means the required device is expensive or requires a specific

license to be presented before purchase. *unprivileged device* (Level 1) means the required device can be acquired from an online store at an affordable price. *None* (Level 2) means the attack does not require a unique device to initiate the attack.

We use the following parameters to assess the impact of different threats. The impact of an attack is closely related to the motivation of adversaries because the scale of impact is related to the scale of profit adversaries can generate. With a higher score, the exploitation of threats will have a greater impact. The same attack can have unforeseeable outcomes in different scenarios.

1. **Hospital operational damage:** Depending on different system architectures, the same threat can have different impacts on hospital operations. Since the system architecture we present is simplified and generalized, it is hard to assess whether the critical infrastructure is compromised under attack. Thus, the damage to hospital operations is only divided into two levels. *unaffected* (Level 0) means the normal operation of a hospital is not affected. *affected* (Level 1) means the hospital system is under threat and the compromised system will partially or paralyze the hospital operation.

2. **Access of digital asset:** The damage to digital assets is divided into three levels. *None* (Level 0) means the digital assets are not compromised. *Indirect access* (Level 1) means the attacker can acquire unlabeled digital assets by utilizing the threat. In this case, the attacker may need additional information to map the data to the patient. *Direct access* (Level 2) means the attacker can acquire data from internal clients or databases.

3. **Physical damage to patient:** The potential risk of physical damage is divided into three levels. *None* (Level 0) means the threat cannot cause physical damage to patients. *Indirect damage* (Level 1) means the threat may indirectly help adversaries to acquire control of functionalities that can damage patients' health. *Direct damage* (Level 2) means the adversaries can cause direct physical damage to patients by utilizing the threat.

3.6.2 Threat prioritization

This section prioritizes the threats and discusses how those threats and attacks affect the environment in which the devices operate and the security requirements of each component. Figure 5 shows the threat prioritization. Using the result of the threat assessment model as a reference, the attacks are classified into two different categories.

Here are attacks that have a relatively higher probability and higher impact:

- **Replay attack:** To initiate the replay attack, the adversaries need intercept devices to intercept and replay the message to the remote device. The attack can misguide the device to overdose the patient, which will cause direct damage to the patient. Thus, it is important to have a monitoring system and alert system, which will alert patients and administrators of abnormal network traffic and device status.

- **Reverse engineering:** It takes skills and time to use reverse engineering techniques to find vulnerabilities in a system. Reverse engineering can be accomplished by

observation of information change or disassembly/decompilation of package. Thus, the confidentiality of messages during the remote update and remote monitoring should be secured.

- **Credential and infrastructure data leakage:** The adversaries can extract credentials by physically tampering the device or spoofing the message in device registration. With the lost credentials, the adversaries can create fake monitor devices to collect user data and send a false signal to increase the dosage of each injection of medicine.
- **Corrupted firmware:** In the remote update of the remote device, the device may not validate the source of firmware, which allows adversaries to inject modified firmware into the device. The remote update system should provide mote devices with the means to verify the source of firmware updates.
- **Unencrypted data:** Depending on the system design, IoT devices communicate with each other with unencrypted payload because of the limitation of resources. Patient information can be extracted through the communication of those devices, such as RFID readers and smart cameras.
- **Camouflaged malware:** It is required to embed complex tools into old malware variants to penetrate the firewall of the healthcare industry. After gaining a foothold inside the hospital, the adversaries can paralyze devices in the hospital, collect confidential healthcare data and cause physical damage to patients.
- **Attacks that target internal client:** In the attack scenarios mentioned in the previous section, the adversaries compromised internal clients with USB drives dropped around the hospital, which require almost no resources and skills. Similarly, with compromised internal clients, the adversaries can cause a similar impact compared to camouflaged malware.
- **Attacks that target non-medical device:** Similar to camouflaged malware and attacks that target internal client, compromising non-medical devices are another method to establish a foothold inside the hospital network.

Here are attacks that have a relatively lower probability or lower impact:

- **Configuration tampering:** Since the device that is used to reconfigure the device can be easily purchased online [Rad11], even adversaries with basic computer skills can easily reconfigure the device. Thus, it is important for the remote device to have a method to validate the identity of the reconfiguration device.
- **Reverse engineering of protocol:** The impact of a compromised communication protocol can lead to the leak of sensitive information and physical damage to patients. As described before, the compromised protocol allows the adversary to abusively use a monitoring device to reprogram remote devices. On the other hand, analyzing protocol to find utilizable vulnerabilities require skills and time effort. Also, the revenue that can be derived from one vulnerability is limited, since it is hard to find massive deployed medical devices. Thus, the probability of

- **Reprogramming:** Reprogramming medical devices requires a special device that cannot be purchased without a license or prescription. For wearable medical devices, such as insulin pumps, it is possible to replace the firmware by physically tampering the devices, which require higher technical skills. Compared to other types of attack, reprogramming of devices is less likely to happen in most cases. However, if the adversaries are able to generate profit by selling modified devices, it is likely to find a massive amount of modified devices in the market.
- **Physical tampering:** For remote health care, physically tampering the device requires physical proximity. It is not easy for an adversary to enter patients' homes. Considering clinical medical devices, medical devices are under access control. Thus, this kind of attack has a relatively lower probability to happen.
- **Reverse engineering:** Similar to the situation of reverse engineering of protocol, reverse engineering of devices requires not only time and skills but also physical proximity.
- **Medical device hijacking:** Medical device hijacking is a set of strategies that adversaries will take after gaining a foothold inside a hospital network. Depending on the countermeasures taken by the cyber-security team in the healthcare industry, the impact and difficulty of penetrating an internal network are different.

3.7 Security requirements

In this section, we list the security requirements based on the threat analysis we performed above targeting different system components. Some of the security requirements may seem irrelevant or technically unsolvable for the remote update and monitoring system. However, those requirements will help us to make pre-assumptions about the environment where the system operates.

3.7.1 Remote healthcare system

Depending on the type of remote devices, the security requirements are different. For some devices, fail of a boot will cause catastrophic results (e.g. cardioverter defibrillator). The remote update is not an option for this kind of device in this case, since the patient needs human intervention when the update causes malfunction. Thus, we only discuss the remote devices that will not cause damage to patients' health in case of startup failure in this section.

- **Wearable and implantable device**
    1. **Tamper resistance:** When it comes to a wearable device, the attacker can perform attacks physically, such as extracting the key through a side-channel attack.
    2. **Cryptographic requirements:** When it comes to remote healthcare devices, it is hard to control the environment in which it is deployed. Thus, when there exist enough computing and storage resources, cryptographic algorithms should be implemented to secure the data and communication of devices. Since the resource for remote devices is limited, the security and complexity of crypto algorithms should both be considered.

3. **Data protection:** Since the remote devices have limited resources and security measures, the size of sensitive data that's stored on the device should be minimized. The device should ensure the integrity of data and messages in the remote update and monitoring. If the remote device does not have enough resources, concessions can be made on the confidentiality of certain types of messages, such as device status information or device configuration, in remote monitoring.
4. **Remote configuration method:** The maintenance of the configuration of remote health care devices is always a difficult problem to solve. The attacker can target misconfigured application-layer protocol. Also, to update the firmware, it is necessary for the device to update its configuration at the same time. In this case, a reliable remote configuration update method is necessary.
5. **Proper key management:** The number of solutions for key management is constrained by the limited resource of remote devices. The baseline for key management is that different devices of the same model should use different private keys for authentication and the device should not use easily accessible data, such as sequence number printed on the label of the device, as the private key. If there is not enough resource for a proper key exchange protocol to be implemented, the device can use a pre-shared key.
6. **Security of protocol:** The design of protocol should have the ability to mitigate common techniques adversaries use, such as the reply attack and man-in-the-middle attack.
7. **Authentication:** The remote device should have some method to ensure the update command is coming from authenticated monitor device. Since the adversary can easily have access to the monitoring device, it is important to ensure that only authenticated monitoring device can update the remote device. It means the device should have a proper registration process.
8. **Remote monitoring:** When it comes to remote update methodology, it is important to have the ability to monitor the status of the device remotely. Thus, when an update fails, the system can alert the administrator to take necessary action to resolve the issue. Also, the system can alert users about the unusual status of devices, such as the drain of battery and unusual data from remote devices.
9. **Alert system:** The alert system is used in case of failed updates and detected breaches. Sometimes, the update fails and the system fails to roll back to the last version. In this case, the device should be able to send an alert to the user to inform the user about the situation of the device. In case of detected breaches, the system should have methods to alert the user about device anomalies so that users can take necessary action to protect themselves.
10. **Dispose procedure:** The device needs to be discarded properly, since it may contain sensitive information, such as the private key and patient personal information. Also, the monitoring device needs to be alerted about the disposal of the device.

- **Home monitoring device**

1. **Tamper resistance:** Monitoring device is more likely to suffer from attacks involving physical tampering. Also, the monitoring device usually contains much more sensitive information, which gives adversaries a higher motivation to initiate the attack.
2. **Cryptographic requirements:** When it comes to the monitoring device, the available resources are much more flexible. Thus, we have much more choices of crypto algorithms in this case. Since the disposal of home monitoring devices is not performed under a controlled environment, stronger crypto algorithms can be used to increase the difficulty of extracting data from monitoring devices.
3. **Monitoring and alert system:** The monitoring device should have protocol and systems to help the remote device to build the remote monitoring and alert system to ensure that users and administrators are informed about the status of devices.
4. **Key management of remote devices:** The proper key management is especially important for the monitoring device. Many remote update methods for resource-constrained IoT devices use a pre-shared symmetric key for authentication. The monitoring device should ensure that the correct key is registered when the remote device is first deployed and the key is deleted when the remote device is unregistered or discarded.
5. **Key management of monitoring device:** The monitoring device should have a proper key management system to comply with the hospital server.
6. **Authentication:** First, the monitoring device should validate the source of the update command and firmware package. Second, the monitoring device should validate the identity of the device before sending the firmware package. Third, the monitoring device should have an authentication procedure to validate the user is authenticated to use the device.
7. **Disposal procedure:** When the device is discarded or reset, all sensitive data should be deleted. Also, the server should be notified when the monitoring device is disposed and the remote devices that are managed by the remote monitoring device should be handled properly.

3.7.2 Clinical IoT device

Compare to the network environment of remote healthcare devices, the healthcare institution protects its devices through many different security measures. However, we can not assume the internal network is not compromised. Based on our threat analysis, the adversaries can successfully penetrate the hospital firewalls and escalate privileges inside the network in multiple healthcare institutions. When we consider the security requirements for clinical IoT devices, it is assumed that the adversaries have the ability to gain a foothold inside the hospital network, perform reconnaissance and acquire control of some of the internal clients.

For clinical IoT systems, different parties are responsible to secure different components. Manufacturers are responsible for Medical devices, whose internal logic is invisible to the health institution. The health institution is responsible for the server, internal client, and other supportive systems (e.g. smart camera/RFID systems, smart light).

For most supportive systems in the health institution, the operating environment is similar to the remote medical device. The resource that is available to remote device is

constrained and a huge number of homogeneous devices are deployed. Depending on the design of the system, the remote devices can be managed by regional management clients or hospital servers. Here are the security requirements for different components of supportive IoT systems.

- **Mote devices:** For devices such as smart lights and thermostats, there are not enough resources to implement any security measures. Thus, we only discuss the security requirements of devices that have available resources for deploying security measures in the remote update and monitoring.

    1. **Privilege control:** Since the massive-scale deployment of homogeneous devices, it is hard to ensure that all devices are not compromised. Thus, the privilege of each device should be managed carefully to prevent the attacker from gathering information and escalating privilege after getting a foothold on the device.
    2. **Complexity of cryptography:** The complexity of cryptography that is used should be considered since the resource that is available on mote devices are limited.
    3. **Data protection:** The device should ensure the integrity of data. For devices such as thermostats and smart lights, the message for remote monitoring does not need to be confidential. For devices such as smart cameras and RFID readers, the confidentiality of all messages needs to be secured.
    4. **Key management:** The key management system is necessary for remote devices. At least, the device should secure the pre-shared key. A key distribution mechanism can be optional.
    5. **Authentication:** The device should authenticate the originality of the message in remote update and remote monitoring. This means the device should have a registration process before deployment, either manually or automatically.
    6. **Disposal procedure:** When the device is discarded, the management server should be notified.

- **Management client:** The management clients are desktops and local servers that are used for device management and initial processing of data. Some mote devices do not have enough resources to receive updates from servers directly. Thus, the update and monitoring of devices need to be managed by these management clients.

    1. **Endpoint security:** The endpoint security is critical, especially for desktops that are used for device management. The desktop can be easily compromised according to our threat assessment. Thus, internal firewalls and routine scan of systems is necessary in this case.
    2. **Data protection:** The device should handle ensure confidentiality and integrity when it transfers the data to the central server. Also, the device should not minimize the information it stores, since these clients are usually the targets for adversaries after a foothold is established inside the hospital network.
    3. **Registration and key management:** The device should be registered before being deployed. The management clients need to collect information, such as device identification code and pre-shared key, before deploying the device.

When a device is unregistered or discarded, the management clients should delete information related to the device.

4. **Authentication:** The management client should have an authentication process to ensure the identity of devices to prevent adversaries from using an unregistered device to send incorrect status info or receive firmware updates.

The medical devices are usually maintained by the manufacturer. Clinical medical devices usually have greater computing and storage resources available compared to the mote devices. Also, they handle sensitive health information. Here are the security requirements of **medical devices**:

1. **Tamper resistance:** Since most medical devices operate in a controlled environment, it is hard for adversaries to have physical access to the device. On the other hand, the internal attackers may not have the knowledge base to initiate an attack through physical tampering.

2. **Data protection:** The device should ensure the integrity and confidentiality of messages related to the device status in the remote monitoring system.

3. **Remote monitoring** The device should have mechanisms to allow the server to check the integrity of the system in devices and the device status remotely.

4. **Registration and key management:** The device should have proper key distribution and registration procedure before deploying.

5. **Authentication:** The device should authenticate the originality of the update command. Also, the device should check the identity of the monitoring device before sending the device status.

6. **Dispose procedure:** The device should notify the server about the disposal of devices. The device should delete sensitive information and the server should delete information related to the device.

The security requirements for the server are not discussed since it is operating in a strictly controlled environment. Also, the server is a whitebox for the cyber defense team in the health institution, which means they can set deploy a defense strategy targeting the requirements of the server. In this thesis, the server is considered a trusted component.

Figure 5. Threat Prioritization

Description of attacks			Possibility assessment			Impact assessment				
Types of attack	**Related vulnerabilities**	**Related high-level threats**	**Required Access [Physical, Logical]**	**Required Capabilities [Skill, Resource]**	**Score**	**Hospital Operational Damage**	**Access to Digital Asset**	**Physical Damage to Patient**	**Score**	**Criticality Level**
reverse engineering of protocol	Device tempering, Reverse engineering, Attacks that target protocol	Communication Protocol, Weak Endpoint Security of Medical Devices	[Unprivileged, None]	[Expert,Unprivileged]	4	unaffected	Indirect	Direct	3	Meidum
replay attack	Device tempering, Attacks that target protocol	Communication Protocol	[Unprivileged, None]	[Moderate, Unprivileged]	5	unaffected	Indirect	Direct	3	High
configuration tempering	device tempering	Abusive use of medical device	[Unprivileged, Indirect]	[Novice, Unprivileged]	5	unaffected	None	Direct	2	High
reprogramming	Device tempering	Abusive use of medical device, Unauthorized control	[Privileged, Direct]	[Moderate, Privileged]	1	unaffected	Indirect	Direct	3	Meidum
physical tempering	Malware, Medical device tempering	Lack of tamper resistance, Weak hardware implementation, Firmware security, Unauthorized control	[Privileged, None]	[Moderate, Unprivileged]	4	unaffected	Direct	Indirect	3	Meidum
reverse engineering	Reverse engineering, Device and data theft	Firmware security,	[Privileged, None]	[Expert, None]	4	unaffected	Direct	Indirect	3	Meidum
credential and infrastructure data leakage	Device and data theft	Key management	[Unprivileged, None]	[Moderate, Unprivileged]	5	unaffected	Direct	Indirect	3	High
corrupted firmware (During remote update)	Medical device tempering, Attacks that target protocol	Firmware security, Communication protocol	[None, None]	[Moderate, None]	7	unaffected	Direct	Indirect	3	High
unencrypted data	Device tempering	Key management, Lack of tamper resistance	[Unprivileged, None]	[Novice, None]	7	unaffected	Direct	Indirect	3	High
camouflaged malware	Malware	Interconnection of IoT devices, Life span of Medical devices, Weak Endpoint security of Medical devices, Unvisible internal functioning of IoT devices	[None, None]	[Expert, None]	6	affected	Indirect	Indirect	3	High
medical device hijacking	Malware, Device tempering	Interconnection of IoT devices, Mass-scale deployment of homogeneous IoT devices, Weak end-point security of medical devices	[None, Indirect]	[Expert, Unprivileged]	4	affected	Direct	Indirect	4	Meidum
attacks that target internal client	Malware, Social engineering attack	Interconnection of IoT devices, Uninformed breaches, User behavior	[None, None]	[Novice, None]	8	affected	Direct	Indirect	4	High
attacks that target non-medical devices	Device tempering	Lack of temper resistance, Interconnection of IoT devices, Mass-scale deployment of homogeneous IoT devices	[Unprivileged, None]	[Moderate, None]	6	affected	Indirect	Indirect	3	High

4 Survey of existing remote update technologies

The remote update systems of IoT devices are widely used in other industries. We surveyed existing remote update technologies based on the security requirements demonstrated in previous section. This section demonstrates three remote update technologies that resolve some of the security requirements in the remote update for different healthcare system components. These technologies are very suitable as prototypes for further development targeting the security requirements of specific systems and devices.

4.1 SRUP: The Secure Remote Update Protocol

Poulter et al. [PJC16] proposed a remote update protocol build based on MQTT in 2016. The protocol is intended to send commands to IoT devices over MQTT to initiate the remote update. The protocol utilize public-key encryption to identify both server and devices. Thus, a key-distribution mechanism is required during device registration. The protocol is designed to operate in WAN. The devices and MQTT brokers typically locate inside a LAN, which is protected by a firewall. The devices should have the privilege to send and receive messages through the internet. The server typically locates outside the firewall. The servers should have the ability to communicate with the MQTT broker.

Instead of pushing the software to the device, the server sends a signal (the initiate message) to initiate the device to download software updates in SRUP. It is suggested that the ability to upload the software to devices from the server implies a port can be opened directly on devices. This increases the risk of that device being compromised. Thus, SRUP uses a signal to initiate software download on IoT devices. It means the IoT device does not have to be addressable from the Internet: as some devices may operate inside a firewall or behind a router using NAT.

The protocol relies on MQTT to ensure the delivery of messages. With different settings of quality of services, the MQTT broker can ensure the device can receive the update message at least once even if the device is offline when the server sends the initiate message to the MQTT broker.

4.1.1 Protocol details

The protocol consists of three messages. The *initiate message* from the server initiates the update. Using the URL provided by the initiate message, the device downloads the software update. The transmission of software/firmware update packages is using TLS and HTTPS to ensure authentication and confidentiality. Also, the protocol uses SHA2 to ensure the integrity of software. If the device has enough computational power, the message can be implemented using MQTT over TLS for maximum security. After downloading the software/firmware update package, the device sends a *response message* to report the status of the download. After receiving the response message, the server sends the *activate message* to signal the device to install the new software or apply the new configuration. After finishing the installation, a *second response message* from the device can be used if necessary to report the status of package installation.

1. **Initiate message:** The server sends this message to the device to initiate the update.
    - *Version:* The version of SRUP.

- *Message Type:* The type of SRUP message: SRUP_MESSAGE_TYPE_ INITIATE.
- *Signature:* The cryptographic signature of the message is calculated by the private key of the server.
- *UUID:* The universally unique device identification number.
- *Token:* A unique token to identify this SRUP transaction.
- *URL*: The URL for the device to retrieve software update packages.
- *Digest:* The Hash value of the update to be retrieved.

2. **Response message:** The device sends this message to the server to report the update status. The response message is sent after the initiate message to report whether the download of the update package is successful.

   - *Version:* The version of SRUP.
   - *Message Type:* The type of SRUP message: SRUP_MESSAGE_TYPE_ RESPONSE.
   - *Signature:* The cryptographic signature of the message is calculated by the private key of the device.
   - *Token:* The token specified in the initiate message.
   - *Status:* The status of update, including successfully received update, server no response, file retrieve failed and digest unmatched. The type of status can be added according to the requirement.

3. **Activate message:** The server sends this message to the device to request the device to activate the newly downloaded package.

   - *Version:* The version of SRUP.
   - *Message Type:* The type of SRUP message: SRUP_MESSAGE_TYPE_ ACTIVATE.
   - *Signature:* The cryptographic signature of the message is calculated by the private key of the server.
   - *Token:* The token specified in the initiate message and included in the response message

4. **Second response message (optional):** The device sends the second response message to the server to report the status of activating the new package. The structure of the second response message is similar to the first response message.

4.1.2 Enhancements of original protocol

In 2017, Poulter et al. [PJC17] proposed several enhancements to expand the functionalities of protocol and address unresolved problems.

Mitigation of Replay attack The original protocol has a token to identify the SRUP transaction. However, after capturing the first update message and detecting the second update message, it is possible for adversaries to send the captured initiate message and activation message to roll the devices back to the first version. Also, adversaries can use this mechanism to stage a DoS attack against a specific device by constantly sending it the same update message. In this case, the device will be forced to apply the same update constantly.

The common mitigation method to replay attack is not suitable for SRUP. The first method is to introduce a one-time nonce in the communication. However, by introducing the nonce, the protocol requires two more messages: 1. a response to the initiate message from the device which contains the nonce from the device; 2. a second message from the server which confirms the initiation. This increases the communication overhead and can lead to update failure for devices with poor-quality network connections. The second method is to introduce an accurate timestamp. However, this requires all devices have access to a synchronized time signal. The third method is to require each device to keep a log of all the tokens it received, which consumes the limited storage resource for IoT devices.

The adopted approach is to implement a sequence ID in the protocol. By implementing a 64-bit sequence ID, which permits a system to send 1,000,000 update messages. The server will store the last value that is used to communicate with the device and increment the value by one for the new message. the device will store the sequence ID of the latest message. When a new message is received, if the sequence ID is less than the record, the message is discarded.

Extra functionalities in multi-server environment The original SRUP assumed there was no requirement for the devices to communicate with multiple servers. However, in real-world scenarios, one clinical medical device may use hardware components from different manufacturers. Also, with the ability to receive updates from multiple servers, the remote update system is less likely to be affected by the supply chain failure. Thus, it is important to allow the device to receive update packages from different servers. To use the SRUP in a multi-server environment, several challenges need to be addressed. First, the device needs to identify which public key to use to verify the signature. Second, with multiple servers, the token that identifies each message is not unique anymore. Thus, the recipient of a message needs to be identified through other approaches.

There are three possible solutions to the first challenge. First, the device can verify the signature against each public key. However, this increases the computation overhead and is not suitable for resource-limited devices. The second is to use the MQTT topic hierarchy to signify the sender. Each server will be represented using the MQTT subtopics. The device can subscribe to the higher-level topics and use the subtopics to indicate the originality of update messages. However, this practice requires the SRUP to be used in MQTT. Also, maintaining the hierarchy of MQTT is relatively complex. The third method, which is the method that's implemented, is to include a server ID within the SRUP message. This method slightly increases the communication overhead. However, this method requires less maintenance after implementation compared to the complexity of the second approach.

The solution to the second challenge is similar to the first one. The first approach is to have subtopics mapping to specific devices. With multiple servers subscribing to a specified subtopic of a specific device, combined with the server ID, the recipient of a message is identified. The second is to include an additional destination ID, which allows the SRUP to be used without MQTT.

Key Distribution and device registration Since SRUP uses public-key cryptography, the problem is simplified to ensure the integrity of exchanged public keys. If the key exchange message is intercepted, the adversary can impersonate the server to the device or the device to the server by replacing the public key with the adversaries' public key.

The suggested solution by SURP is to implement a RESTful HTTPS web service on the server for device registration. The registration URL can be hard-coded into devices or manually entered. Also, it is possible to use a non-textual encoding of URL, such as QR-code or two-dimensional bar-code. The device will send a POST request to send its UUID and public key. In return, the server will respond with the URL of the MQTT broker and the server's public key. Since the registration requests can be received from any device, the registration does not prove the identity of the device. Thus, the registration of devices does not establish any trust relationship between the server and the device.

Proof of Identity SRUP suggested two possible join mechanisms for registered devices. The first is the Human-in-the-Loop system. The second is an autonomous system.

The first system involves a human operator in the loop. In this case, the device will initiate the process by sending the join request. Then, the server will respond with a value encrypted with the public key of the device. After the device receives the message, the device would present the decrypted value to the human observer. Then the observer can compare the value with the operator of the server. If the value matches, the identity of the device is confirmed.

The autonomous system has a similar design to the first one. The only difference is that the observer will be another device (beacon device), instead of a human. After the server sends the encrypted value to the device, it will request the beacon device to establish a point-to-point connection with the new device. Then, the beacon device can compare the value from the server and the value from the new device.

Key revocation SRUP has a key revocation process, which is an overlooked aspect in many IoT systems. The device can be resigned or terminated. If the device resigns from a server, the server will delete the public key and data of the device. If the device is terminated, all servers that this device belongs to will delete the data and public key of the device. If the revocation process is initiated by the devices, the device can send the signal to resign or terminate. Then, the server will respond with the decision on whether the resignation or termination is accepted. If the device is disconnected by the server, the server can simply remove the device's data from its list of devices.

4.2 Remote software update in LPWAN

Kim et al. [DYSJH18] proposed an approach to update software in LPWAN in 2017. It was proposed that to reduce the energy and memory consumption of remote software updates, instead of replacing the software image, the update should be performed on a function-by-function base. Based on the ROCE, a remote software management method proposed by Jung et al. [MDYS16], a remote software management design was proposed.

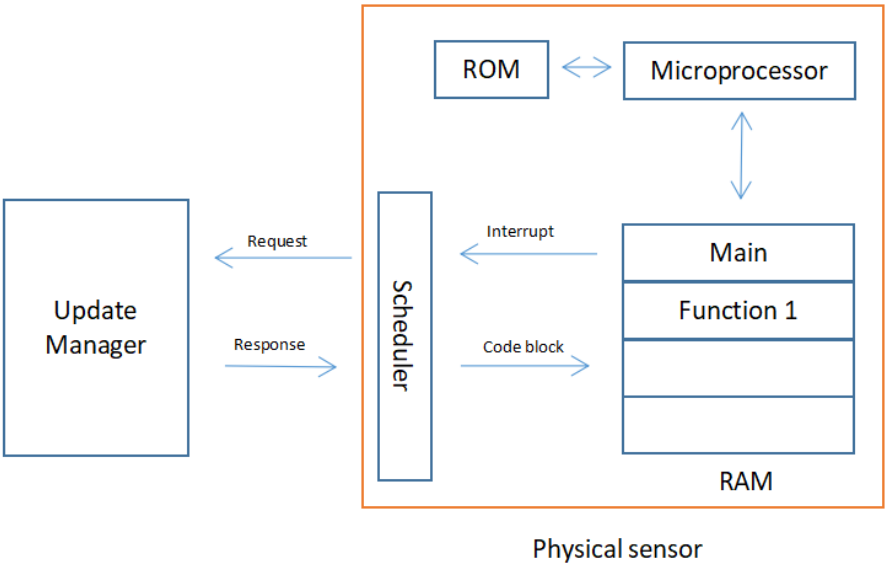

Figure 6. ROCE

4.2.1 ROCE

ROCE is a remote on-demand code execution method for memoryless sensors proposed by Jung et al. Instead of providing the entire code file to a physical sensor, ROCE proposed to partition code into function block units. The physical sensor will be interrupted until it receives the next code block. The physical sensors and the update managers locate inside the same private WLAN. Figure 6 shows the design of ROCE.

The sensor adopts a specific procedure to comply with ROCE. Once the sensor is turned on, it will execute code for initialization, which is included in ROM. After initialization, the sensor will request for main function block through the scheduler. Once RAM received the main function block, the microprocessor will execute the instructions. When a jump or branch instruction appears, the program counter will shift to the scheduler function. The scheduler function will calculate the start address of the next function block, request the function block from the update manager, and add the function block to the scheduler table. This process will be carried out repeatedly. If the requested function exists in the scheduler table, the function will be executed. Otherwise, the scheduler will send a request to the update manager for the function block.

The benefit of partitioning the function into blocks is that the device does not have to maintain the code image in the on-chip flash memory. The on-chip flash memory requires a large area of the chip and is responsible for the generation of a high access current, which consumes a lot of energy. Thus, with the use of ROCE, the lifetime of devices that relies on batteries is extended.

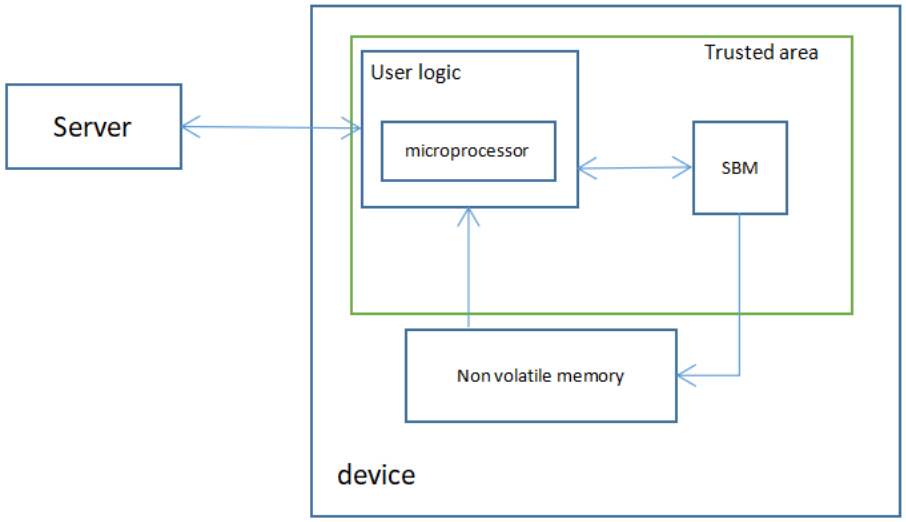

Figure 7. SARFUM

4.2.2 Software update management

Based on ROCE, Kim et al. [DYSJH18] proposed network architecture. The network architecture includes the ASP server, MEC servers, and LPWAN mote devices. The ASP server provides service functionality for available services. It separates code into code blocks and integrates code blocks into service packages. The MEC server, which is the update manager, provides computing and storage resources. It manages and delivers service functions. Also, it analyzes the wireless connection and supports decision-making for software updates of mote devices.

When the remote device is initialized, it will register itself in the MEC server. The MEC server will request a service package from the ASP server. The service package includes code blocks for the device. The scheduler of the device will obtain the required code blocks from the MEC server. If the MEC server contains the required code blocks, it will send the code block. If not, the MEC server will request a service package from the ASP server.

4.3 SARFUM

Benoît et al. [BCE+10] built a remote update protocol called SARFUM to secure the confidentiality and integrity of FPGA bitstreams and to provide functionalities for remotely monitoring FPGA configuration. The SARFUM is designed to allow the system designer (server) to reprogram FPGA remotely over the internet. Figure 7 shows the architecture of SARFUM. The server encrypts the bitstream and computes an authentication tag with a pre-shared key. On the device side, new hardware, which is SBM, is introduced for decryption and verification of authentication tags. To ensure the freshness of the bitstream, a Nonce (Bitstream Version Number) is generated by the server and SBM with a monotonic counter. A microprocessor is embedded in the FPGA chip for network communication. To provide devices the ability to support basic services for network communication after an attack on the main configuration, the non-volatile memory has a golden partition that

contains bitstreams for basic network service.

The SBM has 4 non-volatile registers initialized before deployment:

1. Secret key: A pre-shared secret key between device and server
2. BVN: The Nonce to ensure freshness of bitstream
3. FAIL: A value to indicate whether the last bitstream loaded was properly authenticated
4. UPDATE: A value that specifies the event that triggers the last power-up (1 for update command and 0 for regular platform reset)
5. NVMPART: A value that indicates which partition of non-volatile memory to load from. If it is set to 0, it means the SBM should load bitstream from the golden partition.

FPGA configuration update: To start the update, the server sends the SBM update command. Then the server sends the bitstream, which is ciphered and signed. When receiving the message, SBM verifies the command. If the command is verified, the bitstream and signature are stored in NVM. Then, FPGA is rebooted. The SBM also update its BVN. Once the BVN is updated, the bitstream corresponding to the previous BVN is not accepted by SBM anymore on power-up.

FPGA configuration verification at power-up: The FPGA device loads bitstream from NVM at power-up. The SBM authenticates the bitstream with the BVN and signature of the bitstream before it is loaded by the FPGA device. The following is the procedure for fetching the bitstream from non-volatile memory. If NVMPART is 1:

1. Decrypt the bitstream and verify the authenticated tag. Since the FPGA chip has limited storage, this process is done on-the-fly.
2. If the bitstream is verified, the SBM looks up the UPDATE register. If it is set to 1, SBM sets UPDATE to 0 and prepares a message to signal the successful update to the server.
3. If the bitstream is corrupted, FAIL is set to 1, and NVMPART is set to 0.

If NVMPART is 0:

1. Load golden configuration
2. Verify the tag of the golden partition. The BVN for the golden partition is 0 by default.
3. If the golden partition bitstream is verified, the SBM checks UPDATE. If UPDATE is 1, SBM sends a message to the server to attest to the update failure. The message includes the update status, the device ID, and the authentication tag with BVN.
4. If the golden partition bitstream is not verified, the system is corrupted and needs manual recovery.

Remote monitoring: The Server can send a message to request an acknowledgment message from the device. The message from the server includes the message header COMMON, the current BVN, a Nonce, and an authentication tag. When SBM receives and verifies the message, it will send the acknowledgment message, which includes the message header COMACKMON, BVN of the device, Device ID, the received nonce, and an authentication tag. The message header will indicate the status of the device.

5 Evaluation of existing remote update technologies

In this section, the technologies listed in section 4 will be evaluated. We will list the security requirements that are solved or related to each technology. Also, we will talk about the limitations of each technology.

5.1 SRUP

Since SRUP is designed based on MQTT, it is more suitable in the hospital environment. Even though the protocol provides extra functions that allow the protocol to operate based on the TCP connection, the solution affects the protocol's efficiency. In remote healthcare, most devices have limited resources and can not use TLS to download packages. Thus, when analyzing the security requirements met by SRUP, we only consider the security requirements of clinical devices.

Here are the security requirements that are related to SRUP:

- **Privileged access:** In the registration process of IoT devices, an automatic registration process is suggested to prove the identity of a new device. However, it means the beacon devices must not be compromised. Also, since the beacon device needs to connect to the new device directly, the endpoint security of the beacon device is also important. If the adversaries gain access to beacon devices, the adversaries can freely register unknown devices into the hospital network and infect newly registered devices. It means the adversaries can potentially have access to all the information that IoT devices have access to in the system where the beacon device locates by compromising the beacon devices.

- **Cryptography complexity:** SRUP does not provide other cryptographic options beside public-key cryptography. If the designer of a remote update system wants to use cryptographic algorithms which have less computation overhead, the designer needs to design different key management processes and registration mechanisms.

- **Data protection:** The image of firmware/software is protected with TLS and HTTPS. The protocol also provides the option to encrypt of MQTT message with TLS. The integrity of messages in MQTT is verified with a cryptographic signature with the private key. Also, the protocol has mitigation of replay attack to prevent adversaries to roll back the software/firmware version of IoT devices by replaying old update commands.

- **Key management:** The protocol suggests the server to provide RESTful service to allow devices to send post requests to share the public key and UUID to the server. If the public key match the UUID, the server can start the join process, which verifies the identity of devices by verifying the private key of devices through signature. The device should have copies of the server's public key to authenticate messages from servers. The distribution of additional servers' public keys can be delivered by the original servers.

- **Authentication:** The authentication of devices is achieved through verification of signature through public-key encryption.

- **Dispose procedure:** The protocol provide two different procedure for key revocation. The system is notified when the device is resigned or terminated.

SRUP provides a viable solution for the remote update of devices. Even though it solves most of the security requirements of clinical IoT devices, it still has certain limitations. Here are some of the challenges SRUP faces:

- **Physical Tampering:** As mentioned by the designer of SRUP, if the device does not have encrypted storage and a trusted boot process, the adversaries can steal the identity of the device. The protocol does not have mechanisms to ensure the integrity of the software and hardware of devices.

- **Resource requirements:** The protocol is built on public-key encryption, which requires more processing power. Also, it requires the device to have the ability to support TLS and HTTPS to download firmware packages. Even though having the ability to support TLS is common for most modern micro-controller, medical devices are concerned about introducing new components into the device. The design of new medical instruments is based on old architecture, which utilizes old technology that does not have the capability to support TLS.

- **Built based on MQTT:** This is an addressed problem since the protocol later provides alternative solutions based on TCP connection. Migration of the protocol to TCP should not be problematic. However, the protocol does rely on the mechanism of MQTT to ensure some of its functionalities, such as ensuring the messaging reliability and persistence. The protocol needs further development before being deployed with a TCP connection.

- **Remote monitoring:** As we mentioned before, the protocol does not have methods to ensure the integrity of the software and hardware of devices. Also, it does not provide mechanisms for remote monitoring. The original author implied encrypted data and trusted boot is a possible solution.

To summarize, SRUP is designed for modern IoT devices. The protocol is complete and is targeting modern IoT devices. For clinical medical devices, using more recent technologies may cause issues for the evaluation of the device by authorities. However, it is suitable for devices in supportive systems, such as RFID readers and smart cameras.

5.2 Remote software update in LPWAN

The most important section is ROCE, which is a remote on-demand execution method. It provides the resource-limited device a method to update the firmware. The technique is designed for devices, which do not even have enough storage for a full image of firmware. It means the device only has resources for basic cryptographic algorithms. However, the communication between sensor and update manager is frequent, since the sensor needs to acquire code function by function. Since the sensor needs to load code from the update manager when it is rebooted, applying security measures to sensors can have a major impact on the sensor's efficiency during boot.

Considering the characteristics of ROCE, this update mechanism is suitable for implantable devices, which have limited resources and can only be updated with wireless

communication with specific reprogram devices. The reprogrammer can serve as the update manager, which receives code blocks from the server through the internet. Then, the reprogrammer can update the implantable devices through short-communication channels with ROCE. The reprogrammer should have enough resources to establish a secured communication channel with the server to retrieve the update package. The communication between the reprogram device and the implantable device is secured through physical proximity since the reprogrammer needs to stay close to the patient to reprogram the implantable device.

Even though it is technically possible to allow patients to update implantable devices remotely, allowing patients to have the ability to update implantable devices without visiting the hospital is worrying. Updating the device always involves the risk of an update failure and device malfunction. Even if the server can be notified of the device failure during the remote update, whether the hospital staff and technician can fix the device before causing physical damage to the patient is questionable. Thus, it is important to evaluate whether the device is suitable to be updated remotely before deploying remote update systems for implantable devices.

5.3 SARFUM

SARFUM is a protocol that is designed to protect the hardware Intellectual Property over insecure communication channels. It is designed to protect the confidentiality and integrity of the bitstream sent to the FPGA platform through the internet.

SARFUM targets FPGA, which is a common technique that's used nowadays inside medical devices. Since the requirement of medical devices is not large compared to other devices, FPGA is used to save budgets on the production of medical devices. It is mostly used for monitoring devices or clinical IoT devices. Thus, we focus on the requirements of monitoring devices and clinical medical devices when assessing SARFUM.

Here are the security requirements that are related to SARFUM:

- **Tamper resistance:** In the protocol, it is assumed that the FPGA chips and SBM are tamper-resist. However, with professional tools, it is possible to attack FPGA chips physically [FM21]. The researcher successfully extracts the key from the FPGA chip. But this kind of attack requires expensive equipment and professional knowledge. Thus, the adversaries are unlikely to choose this approach to compromise the device, since the cost may be higher than the profit.

- **Cryptographic requirement:** The protocol choose AES-CCM for authentication and encryption of data. This is a common cryptographic algorithm that is used for IoT devices. It is suitable for encrypting a short message with a small key size [LAN16]. The key size used in AES-CCM can be 128, 195, and 258 bits. Even with the smallest key size, a brute force attack on the algorithm is not achievable.

- **Data protection:** The originality of bitstream is verified through the authentication tag. The data is encrypted too. Also, the BVN is used to mitigate reply attacks. Thus, the confidentiality and integrity of data are ensured.

- **Monitoring system:** Since the bitstream is verified with BVN and authentication tag, the integrity of the bitstream loaded on the FPGA chip is ensured. Also, the protocol provides methods for the server to request machine status.

- **Key management of monitoring device:** Since the protocol needs to have a pre-shared symmetric key inside SBM, the distribution of key must be handled before deployment.

- **Authentication:** The authentication of messages is achieved by an authentication tag. With a pre-shared key, the device can verify the message from the server with a nonce.

SARFUM is effective in the remote update and monitoring. However, it requires a new piece of hardware inside medical devices. It means it cannot be applied to old medical devices. Overall, SARFUM is suitable for monitoring devices in remote health care and clinical medical devices.

Even though the protocol is designed to be used in public networks, the cryptographic choice of protocol is worrying. AES-CCM is considered secure with even the shortest key size, which is 128 bits. However, attacks that target AES-CCM cipher are mentioned in recent research. [RSWO18] For maximum security, it is better to have a gateway device at the patient's home to enable the remote update of devices.

6 Summary

In this thesis, we discuss the security requirements and potential solutions for remote update systems in medical IoT systems. We found several remote update techniques that exist and compare their security measures with the security requirements of different components of the medical IoT systems. Also, we evaluate the extendability of those technologies to confirm whether they can be used as prototypes for further development.

Based on the evaluation of the technologies, we propose a model that is suitable as a prototype for further development. The SURP is suitable to transmit update packages from the manufacturer's server to devices with enough resources. In clinical IoT systems, such devices include internal clients, hospital servers, internal clients, clinical medical devices, and some wireless devices. In remote healthcare, such devices include remote clients. After receiving the update packages, the clinical medical devices and wireless devices can install the package. The hospital server, internal clients, and remote clients can use ROCE and SARFUM to update devices that have limited resources.

The proposed model does not cover all the security requirements. The ROCE needs additional mechanisms to secure the transmission in internal networks. Also, in real-world scenarios, SURP needs to have more defined states to represent the status of devices. Targeting specific devices and systems, the developer needs to further develop the model to satisfy requirements targeting different types of devices.

The major challenge in the remote update of medical devices is not only the method to secure the integrity and confidentiality of firmware packages, but also the need for counter-measures to system failures during the update process. Thus, the remote update protocol needs extra design before being deployed. In most of the technologies we mentioned, the update command comes from the server. The medical device should have the ability to refuse the update and re-initiate the update later. Also, the remote update of implantable life-sustaining equipment does not falls into the scope of this thesis, since it should only be updated with physicians and technicians around.

References

[BCE+10] Benoît Badrignans, David Champagne, Reouven Elbaz, Catherine Gebotys, and Lionel Torres. SARFUM: Security Architecture for Remote FPGA Update and Monitoring. *ACM Trans. Reconfigurable Technol. Syst.*, 3(2), may 2010.

[DYSJH18] KIM DAE-YOUNG, KIM SEOKHOON, and PARK JONG HYUK. Remote software update in trusted connection of long range IoT networking integrated with Mobile Edge Cloud. *IEEE Access*, 6:66831–66840, 2018.

[Ede16] Terence Eden. The absolute horror of WiFi light switches, 2016.

[Edu15] Kovacs Eduard. Medical Devices Used as Pivot Point in Hospital Attacks: Report, 2015.

[eni15] Security and Resilience in eHealth Infrastructures and Services, Dec 2015. ENISA.

[eni17] Cyber security and resilience for Smart Hospitals, Feb 2017. ENISA.

[FM21] Farimah Farahmandi FahimRahman and Tehranipoor Mark. An End-to-End Bitstream Tamper Attack Against Flip-Chip FPGAs. *Florida Institute for Cybersecurity Research*, 2021.

[Ind16] Independent Security Evaluators. Securing hospitals: A research study and blueprint. Technical report, Feb 2016.

[KPM15] Health care and cyber security: Increasing Threats Require Increased Capabilities, 2015. KPMG.

[LAN16] Ertaul Levent, Mudan Anup, and Sarfaraz Nausheen. Performance Comparison of AES-CCM and AES-GCM Authenticated Encryption Modes. 2016. CSU East Bay.

[Lom15] Natasha Lomas. Critical Flaw IDed In ZigBee Smart Home Devices, 2015.

[MDYS16] Jung Minwoo, Kim Dae-Young, and Kim Seokhoon. Efficient Remote Software Management method based on Dynamic Address Translation for IoT software execution platform in Wireless Sensor Network. *Indian Journal of Science and Technology*, 9(24), 2016.

[med15] MEDJACK.4 Medical Device Hijacking, Feb 2015.

[med16] MEDJACK.2 Hospitals Under Siege, 2016.

[ML16] K. Munro and D. Lodge. Hacking the Mitsubishi Outlander PHEV hybrid, 2016.

[MMD17] Sumit Majumder, Tapas Mondal, and M. Deen. Wearable Sensors for Remote Health Monitoring. *Sensors*, 17(12):130, 2017. MDPI.

[MSG+16] Eduard Marin, Dave Singelée, Flavio D. Garcia, Tom Chothia, Rik Willems, and Bart Preneel. On the (in)security of the latest generation implantable cardiac defibrillators and how to secure them. *Proceedings of the 32nd Annual Conference on Computer Security Applications*, 2016. Association for Computing Machinery.

[PJC16] Andrew John Poulter, Steven J. Johnston, and Simon J. Cox. SRUP: The secure remote update protocol. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 42–47, 2016.

[PJC17] Andrew John Poulter, Steven J. Johnson, and Simon J. Cox. Extensions and Enhancements to "the Secure Remote Update Protocol". *Future Internet*, 9(4), 2017.

[Rad11] Jerome Radcliffe. Hacking Medical Devices for Fun and Insulin: Breaking the Human SCADA System. *Black Hat USA 2011*, 2011.

[RB17] Billy Rios and Jonathan Butts. Security Evaluation of the Implantable Cardiac Device Ecosystem Architecture and Implementation Interdependencies, May 2017.

[RS16] Eyal Ronen and Adi Shamir. Extended Functionality Attacks on IoT Devices: The Case of Smart Lights. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 3–12, 2016.

[RSWO17] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. IoT Goes Nuclear: Creating a ZigBee Chain Reaction. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 195–212, 2017.

[RSWO18] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. IoT Goes Nuclear: Creating a Zigbee Chain Reaction. *IEEE Security Privacy*, 16(1):54–62, 2018.

[SKP+18] Ioannis Stellios, Panayiotis Kotzanikolaou, Mihalis Psarakis, Cristina Alcaraz, and Javier Lopez. A Survey of IoT-Enabled Cyberattacks: Assessing Attack Paths to Critical Infrastructures and Services. *IEEE Communications Surveys Tutorials*, 20(4):3453–3495, 2018.

[Ste17] Cobb Stephen. 10 things to know about the October 21 IoT DDoS attacks, Apr 2017.

Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Xuejun Wu**,
*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Security in Remote Update of Medical Devices,
   *(*title of thesis)

   supervised by Tuomas Aura and Arnis Paršovs.
   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Xuejun Wu
19/04/2022