Malicious IoT Implants: Tampering with Serial Communication over the Internet*

Philipp Morgner, Stefan Pfennig, Dennis Salzner, and Zinaida Benenson

Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
{philipp.morgner,stefan.pfennig,dennis.salzner,zinaida.benenson}@fau.de

Abstract. The expansion of the Internet of Things (IoT) promotes the roll-out of low-power wide-area networks (LPWANs) around the globe. These technologies supply regions and cities with Internet access over the air, similarly to mobile telephony networks, but they are specifically designed for low-power applications and tiny computing devices. Forecasts predict that major countries will be broadly covered with LPWAN connectivity in the near future. In this paper, we investigate how the expansion of the LPWAN infrastructure facilitates new attack vectors in hardware security. In particular, we investigate the threat of malicious modifications in electronic products during the physical distribution process in the supply chain. We explore to which extent such modifications allow attackers to take control over devices after deployment by tampering with the serial communication between processors, sensors, and memory. To this end, we designed and built a malicious IoT implant, a small electronic system that can be inserted in arbitrary electronic products. In our evaluation on real-world products, we show the feasibility of leveraging malicious IoT implants for hardware-level attacks on safety- and security-critical products.

Keywords: IoT · LPWAN · Implant · Serial Communication · Hardware Attack

1 Introduction

The Internet of Things (IoT) promises to optimize workflows, enhance energy efficiency, and to improve our everyday life. According to a recent estimation [11], 11.2 billion IoT devices will be installed by the end of 2018. These devices are connected in mostly wireless and local networks all over the world, comprising together a global IoT infrastructure. In the past, security concerns have been expressed regarding this powerful IoT infrastructure: Besides security issues in IoT devices [28,34], IoT networks [30], and IoT applications [9], the force of these billions of devices can be weaponized for targeted attacks with impactful

* Extended version of the paper published in the Proceedings of the 21st International Symposium on Research in Attacks, Intrusions, and Defenses (RAID 2018). The final authenticated publication is available online at https://doi.org/10.1007/978-3-030-00470-5_25

consequences. Examples are recent denial-of-service (DoS) attacks on Internet infrastructure [3,20], in which attacker-controlled IoT nodes utilize existing IoT infrastructure to build large botnets.

In this paper, we explore a new threat where the connectivity of low-power wide-area networks (LPWANs) is leveraged as a communication channel to control malicious hardware. Our objective is to prove that public IoT infrastructure can be used to perform attacks at hardware level remotely, even if the target device does not feature a network interface. The underlying threat of malicious hardware arises from an untrusted supply chain, in which electronic products are manufactured and shipped in large volumes. The global supply chain of electronic products consists of a number of sequential steps from designing a new product, fabrication process, and distribution to the installation. Hereby, we focus on the physical distribution process that involves entities such as manufacturers, third-party logistics providers, distributors, retailers, and costumers. In addition, government agencies oversee the flow of goods at borders for legal and documentation purposes. Thus, an electronic product can be physically accessed and manipulated by a number of entities during distribution. These entities could be potential attackers or cooperate with an attacker, and therefore the integrity of an product should not be assumed in general. This contradicts the inherent trust of consumers that new products are not tampered with.

Inspired by the leaked NSA ANT catalog [4], we experiment with the insertion of additional hardware, referred to as hardware implants, into an existing electronic system after the fabrication process. Although the threat of hardware implants seems to be acknowledged by the academic security community, previous research on malicious hardware mainly focused on hardware trojans, i.e., diverse types of malicious hardware inserted during design phase [2,8,12,13,18,24] and fabrication phase [5,37,43] but not during the distribution phase.

We summarize our major contributions in this work as follows:

1. We comprehensively explore a new attack vector: malicious IoT implants. We show that IoT infrastructures can be abused for malicious purposes other than DoS attacks. Although the existence of hardware implants is known [4], we are the first in the scientific community that design and build a malicious IoT implant, a low-cost electronic implant to facilitate hardware-level attacks, that connects to the Internet over an IoT infrastructure.
2. We investigate new vulnerabilities on hardware level that exploit insecurities in serial communication on printed circuit boards (PCBs). We start by identifying the de-facto serial communication standards by analyzing over 11,000 microcontroller (MCU) models. Then, we show that serial communication is vulnerable to malicious IoT implants. For our implementation that focuses on the widely-adopted I^2C standard, we introduce four attack procedures in which our implant directly interferes with the communication on I^2C buses. At the end, we discuss the adoption of these attacks to other serial communication standards.

The presented threat is not considered in current threat models for hardware security [15,35] that mainly cover hardware trojans, side-channel attacks, reverse engineering, piracy of intellectual property, and counterfeiting. Also, guidelines on supply chain risks, such as NIST SP 800-161 [6], consider malicious software insertion but no malicious hardware insertion. Thus, the goal of this paper is to demonstrate and understand the feasibility of Internet-connected hardware implants and their effects on the security of arbitrary target devices to raise awareness for this novel threat.

2 Preliminaries

In this section, we present preliminaries on LPWAN infrastructure, serial communication, and introduce the I^2C communication protocol.

2.1 LPWAN Infrastructure

The global IoT infrastructure is split into millions of local networks that are interconnected via the Internet. From an application perspective, these networks can be categorized into body-area, personal-area, local-area, and wide-area networks. In this paper, we focus on LPWANs, which provide connectivity for thousands of IoT nodes across large geographical areas as their wireless range competes with the ranges of mobile telephony networks. In contrast to mobile telephony networks that support high data rates and bandwidths, LPWANs are specifically designed for low-power machine-to-machine (M2M) applications that communicate at low data rates. As of June 2018, a popular LPWAN technology with deployments in over 100 countries is LoRa [26]. LoRa operates in three frequency bands (433/868/915 MHz) at different channels and bandwidths, and uses a chirp chip spectrum modulation scheme that provides a high resistance against wireless interference. These advanced propagation properties allow transmissions of wireless data over distances of up to a few kilometers. The specifications of LoRaWAN, the LoRa network protocol, are maintained by the LoRa Alliance, a global non-profit organization consisting of more than 500 member companies [25]. From a network perspective, LoRaWAN utilizes a star-to-star architecture, in which so-called gateways relay messages either between IoT nodes or from an IoT node to the central network server and vice versa. The wireless transmissions between IoT nodes and the gateway are based on the LoRa technology, while the Internet Protocol (IP) is used for data transfers between gateways and the central network server.

The cost of deploying LPWANs is significant lower than the roll-out of mobile telephony networks such that even non-profit initiatives are able to provide network coverage for entire cities and regions. A prominent example is The Things Network (TTN), a crowd source initiative that claims to have a fast growing community with over 42,000 people in more than 80 countries. The TTN community deploys LoRaWAN gateways world-wide to achieve their objective of enabling a global network for IoT applications without subscription costs. According to TTN, 10 gateways are enough to cover a major city like Amsterdam with wireless connectivity for IoT applications. Currently, almost 4,000 TTN gateways are globally deployed. Besides non-profit initiatives, the roll-out of national-wide LPWANs driven by telecommunication companies is ongoing in many countries, e.g., India [17], Australia [33], and the USA [39]. According to a forecast [27], LPWANs will supersede mobile telephony networks in providing wireless connectivity for IoT applications by 2023.

2.2 Serial Communication

Although electronic products provide a large diversity in function, features, and appearance, their underlying hardware platform follows similar design principles. Typically, the hardware platform consists of a number of integrated circuits (ICs) that are mounted on PCBs and interconnected via on-board communication interfaces. A typical PCB comprises multiple sensors and actuators. Generally, one or more MCUs are present to process the data received from the sensors, as well as memory chips to store data persistently, and network interfaces to communicate with external entities.

For the communication between ICs exist a number of serial and parallel data transmission mechanisms. In parallel communication, multiple bits are transmitted simultaneously over multiple communication channels. This is in contrast to serial communication, where bits are sent sequentially over a single communication channel. Since the cost of ICs is also determined by the number of input and output pins, ICs on PCBs often use serial communication to interact with each other. Serial communication mechanisms can be categorized into synchronous and asynchronous systems. Synchronous systems associate a clock signal to the data signals, which is shared by all bus participants. In asynchronous systems, the data signals are transmitted without a shared clock signal. Most of the serial communication systems comprise a hierarchy of master and slave ICs. MCUs are typically masters and control the communication as well as command slaves, e.g., memory and sensors, to send data or to execute particular tasks.

To determine the most important serial communication interfaces on PCBs, we performed a parametric search on the product databases of six leading MCU suppliers: NXP, Renesas, Microchip, STMicroelectronics (STM), Infineon, and Texas Instruments (TI). In 2016, these suppliers had in sum a market share of 72% of all sold MCUs based on the revenue [16]. We analyzed more than 11,000 MCU models regarding their serial communication interfaces and found that 86.7% have a UART interface, 83.5% support I^{2}C and 63.8% SPI. We also analyzed the support for further serial communication interfaces, such as CAN (34.3%), USB (30.2%), and Ethernet (11.5%), which are mainly application-specific and not as widely supported as SPI, I^{2}C and UART. A detailed analysis can be found in Table 1. Although the support of a serial interface is no warrant that this interface is also used in a product that features this MCU, these numbers indicate the de-facto standards that are supported by leading MCU suppliers. Table 2 shows a comprehensive overview of the most important on-board serial communication interfaces that we introduce in more detail.

Table 1: Number of MCU models sorted by supplier and product families (as of January 2018). If a database entry of an MCU model had no parameter regarding a certain interface, we assume that this interface is not supported. *Notation: 'MS' - market share of MCU sales by revenue in 2016, 'Family' - MCU product family as advertised by the supplier (if applicable), 'Bit' - bit size of the MCU architecture, '#MCUs' - number of MCU models, 'ETH' - Ethernet.*

Supplier	MS	Family	Bit	#MCUs	#MCUs that support					
					UART	I^2C	SPI	CAN	USB	ETH
NXP	19%	i.MX	32	251	243	243	243	219	243	220
		Kinetis	32	928	812	812	812	264	334	72
		LPC	32	540	534	531	482	228	276	136
		MPC	32	762	0	290	94	475	0	0
		S32	32	17	17	6	1	7	0	0
		VF	32	35	34	34	34	34	0	0
Renesas	16%	Various	8	566	550	354	3	36	1	0
		Various	16	2,358	2,304	2,226	485	340	72	0
		Various	32	2,318	2,313	2,069	1,924	1,441	1,298	585
Microchip	14%	AVR	8	49	39	43	45	0	5	0
		PIC	8	116	106	104	104	0	0	0
		PIC	16	366	366	366	366	0	58	0
		PIC	32	241	241	220	241	0	175	0
		SAM	32	255	255	255	255	0	187	0
STM	10%	STM8	8	137	30	42	33	21	0	0
		STM32	32	799	799	796	794	490	598	167
Infineon	7%	Various	8	140	140	16	135	0	0	0
		Various	16	156	156	93	145	88	0	0
		Various	32	308	205	189	206	29	14	11
TI	6%	MSP430	16	536	471	446	500	0	0	0
		DRA	32	28	28	28	28	28	18	25
		DSP	32	175	67	72	54	0	54	48
		Perform.	32	254	124	235	248	170	60	0
		Sitara	32	43	25	26	26	20	26	37
		TDA	32	12	12	12	12	12	8	12
		By bit size	8	1,008	865	559	320	57	14	0
				8.8%	85.8%	55.5%	31.7%	5.7%	1.4%	0
			16	3,416	3,297	3,131	1,496	428	130	0
				30.0%	96.5%	91.7%	43.8%	12.5%	3.8%	0
			32	6,966	5,709	5,818	5,454	3,417	3,291	1,313
				61.2%	81.9%	83.5%	78.3%	49.1%	47.3%	18.9%
		In sum		11,390	9,871	9,508	7,270	3,902	3,435	1,313
				100.0%	86.7%	83.5%	63.8%	34.3%	30.2%	11.5%

Table 2: Comparison of serial communication systems.

Bus system	Signal Lines	Synchronous	Asynchronous	Half Duplex	Full Duplex	Multi-Master	Protocol
UART	2	○	●	○	●	○	○
I^{2}C	2	●	○	●	○	●	●
SPI	3+	●	○	○	●	○	○

UART. The Universal Asynchronous Receiver/ Transmitter (UART) is a serial communication interface that uses two data signals: one for receiving, and another one for transmitting. The communicating parties have to agree on the data rate and are synchronized via a start bit. UART supports full-duplex communication, which means that data can be transmitted in both directions simultaneously. UART's main use case is the communication with external hardware components via cables. In contrast, SPI and I^{2}C are used for the communication of peripheral devices on the same circuit board, and thus for shorter distances.

I^{2}C. The Inter-Integrated Circuit (I^{2}C) bus [31], also known as 2-Wire Interface (TWI), was designed by Philips Semiconductor in 1982 with the objective to provide a simple communication mechanism between ICs on a PCB. The original specifications from 1982 allow 100 kHz communication, use 7 bit addresses, and the number of devices per bus was limited to 112 (as a number of addresses is reserved). I^{2}C requires two signal lines, data and clock, and allows half-duplex communication, i.e., data can be transmitted in both directions but not at the same time. Compared to other serial buses, I^{2}C includes a communication protocol that allows masters to communicate with slaves in a coordinated way. I^{2}C is well suited for general purpose communication and electronic products comprising a number of ICs that communicate with each other.

SPI. The Serial Peripheral Interface (SPI) is a serial communication system that uses at least three signals: two data signals and a clock. If the master controls more than one slave, then a further selection signal is required for each slave. SPI is used for full-duplex data transfers that reach data rates up to 1Mbit/s. The main drawback of SPI is the number of signal lines, which increases linearly with the number of slaves. For each slave, an additional select signal line is required, which requires additional I/O pins at the master IC and this adds challenges in placing the signal lines on the PCB. Another drawback is the limitation to only one master. Thus, SPI is well suited for cases in which a single master is connected to one or two slaves and a high data rate in both directions is required.

2.3 I^{2}C Communication Protocol

Although serial communication interfaces have diverse properties regarding synchronization, data rates, and complexity, there are architectural similarities from a security perspective. The most obvious property of these systems is that none of their specifications define any kind of cryptographic security measure. Therefore, the majority of the demonstrated attacks can also be adapted to other serial communication interfaces. In the implementation and evaluation of this work, we focus on the I^{2}C serial bus [31] for following reasons: I^{2}C facilitates a sophisticated communication protocol, in contrast to UART and SPI. Furthermore, I^{2}C and UART are the most widely supported serial communication interfaces, and in 32-bit architectures (which make 61.1% off all evaluated MCU models), I^{2}C is even the most supported serial communication interface.

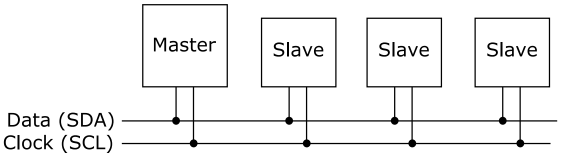


Fig. 1: An exemplary I^{2}C bus system with a single master and three slaves.

I^{2}C uses two signal lines: one clock line (denoted as SCL) and one data line (denoted as SDA). ICs are chained along these two signal lines, which are referred to as bus. In order to request and send data from one IC to another, each IC has a distinct address. Furthermore, each IC can be configured to act either as master or slave. The I^{2}C standard supports multiple masters, which can initiate transactions on the bus. The master that currently performs a transaction also generates the clock signal. Slaves cannot start own transactions and remain passive until they respond to the requests of masters. Typical examples of masters are MCUs and processors, while sensors, memory chips, and actuators are usually configured as slaves.

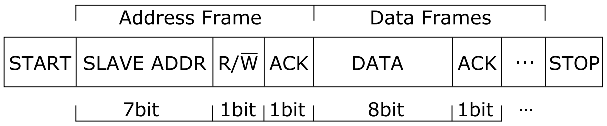


Fig. 2: An I^{2}C transaction consists of an address frame and one or more data frames.

A transaction between master and slaves contains two types of frames (cf. Figure 2): An *address frame* that informs all participants at the bus for which slave the message is intended, and one or more *data frames*, each consisting of an 8-bit data block. To start a new transaction, a master sends a start condition indicating its intention to occupy the bus. If more than one master aims to use the bus at the same time, the master get access that pulls the SDA line with a clock signal first. The other masters wait until the current bus master completes its transaction via a stop sequence. Upon receiving a start sequence, all slaves on the bus listen for an address frame. The master sends the 7 bit address[1] of the corresponding slave after which only this particular slave continues listening. Then, the master sends an 8th bit to indicate whether he wants to write or read. Once these 8 bits are sent by the master, the receiving slave sends a bit to acknowledge its readiness to receive data. In case of no explicit acknowledgment bit was received, the master aborts the transaction.

After the address frame is sent, the transmission of the data frames starts. Depending on whether the master indicated its intention to read or write, either the master or the slave writes data on the SDA line and the corresponding device acknowledges the receipt. Finally, the master sends a stop condition to complete the transaction.

3 Threat Model

Serial communication on PCBs is security-critical as many high-level applications rely on correct data transmissions to function properly. For instance, spoofing of a temperature sensor with false values can have a significant impact on manufacturing processes that require a particular temperature. The injection of wrong gyroscope data into the serial communication of an unmanned aerial vehicle can lead to a crash. Eavesdropping the passcode entered into the pin pad of a safe grants an attacker access to the content without using brute force. The manipulation of loudspeakers in headphones can injure the hearing ability of the user. All these examples show that attacks on serial communication between ICs have serious impacts. To this end, we define following security goals for the serial communication between ICs on PCB boards: (a) *Confidentiality*: Only legitimate ICs have access to the data that is transmitted on the serial bus. (b) *Integrity*: The tampering with data on the serial bus during transfer is recognized by the legitimate ICs. (c) *Availability*: The legitimate ICs always have access to the transmitted data on the serial bus.

In this paper, we present a threat model that involves a so-called malicious IoT implant. Malicious IoT implants are electronic systems that are inserted into an existing system after the fabrication process, which feature a bidirectional direct wireless connection to a public IoT infrastructure. The system that hosts the implant is denoted as target system. We refer to the entity that inserts the implant into the target system as attacker. The objective of the attacker is to violate the security goals of the serial communication between ICs.

3.1 Untrusted Supply Chain

From an economic perspective, a supply chain can be described as a series of inter-related business processes ranging from the acquisition and transformation

[1] For simplicity, we only consider the 7 bit address space in this paper. There exists the possibility of 10-bit addresses as described in Section 2.2.

of raw materials and parts into products to the distribution and promotion of these products to the retailers or customers [29]. The supply chain process can be divided into two main business processes: *material management* and *physical distribution*. In this work, we focus on the physical distribution as malicious IoT implants are inserted into the target system after its fabrication.

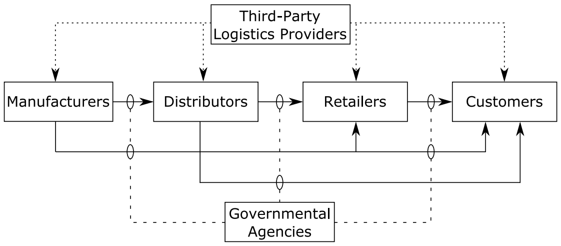


Fig. 3: Physical distribution of goods in the supply chain process. Solid lines: flow of goods. Dashed lines: flow of services (third-party logistics providers) or possibility of interception (government agencies).

We identified a number of stakeholders that are involved in the physical distribution process shown in Figure 3: *Manufacturers* use raw materials and parts to produce goods. *Distributors* buy goods from manufacturers, store and resell them either to retailers or customers. *Retailers* sell goods to customers. *Third-party logistics providers* manage the flow of goods between point of origin and destination, which includes shipping, inventory, warehousing, and packaging. *Government agencies*, e.g., customs inspection, enforce regulations and document the flow of goods in and out of a country. *Customers* receive and consume goods, while having the ability to choose between different products and suppliers. Hence, the physical distribution process provides many entry points for attackers to gain physical access to a target device. Potentially any of these stakeholders can either be an attacker or cooperate with an attacker. Therefore, we assume an untrusted supply chain in our threat model.

3.2 Attacker Model

We assume that the attacker has physical access to the target device as described in Section 3.1, and is able to remove the device's enclosure without leaving physical traces. The attacker identifies access points on the PCB to which a malicious IoT implant can be connected within a reasonable amount of time. We further assume that the target device only requires a power supply, neither Internet nor network access are necessary. The attacker succeeds with an attack if the implant is able to interfere with the communication of the serial buses and cannot be detected without opening the enclosure of the product. Thus, we assume that the attacker targets systems that are not likely to be disassembled by the user. Furthermore, we assume that the attacker has access to a public IoT infrastructure within the wireless range of the implant. In this case, the attacker is not required to be physically present within the wireless range of the implant.

The motivations to utilize malicious IoT implants are various. Governmental organizations might have an interest to use this approach for surveillance, industrial espionage, or the manipulation of infrastructure in enemy states. Leaked documents of the National Security Agency [4] indicate the usage of similar malicious hardware for these purposes. Besides governmental entities, criminal organizations and terrorist groups can use malicious IoT implants to achieve similar goals for financial and political profit. All these groups are likely to be experienced in covert operations, and have the potential to access target devices in the supply chain.

We further categorize the potential motivations of an attacker to interfere with serial communication on PCBs in four high-level objectives: (1) *Disable Services and Infrastructure*: The attacker can use a malicious IoT implant to completely disable a serial communication bus of a device. As a result, an MCU or processor cannot communicate with peripheral ICs anymore. This immediately leads to consequences in high-level applications. (2) *Bypass Security Mechanisms*: Due to the implant's ability to directly interfere at hardware-level, security mechanisms at software-level can be overruled. An example is a lock using an authentication mechanism such that only authorized people can unlock it. A malicious IoT implant can bypass security mechanisms and send commands directly to the actuator that controls the lock. (3) *Bypass Safety Mechanisms*: Safety mechanisms can be overruled that same way as security mechanisms. An example is a software-implemented safety mechanism that controls the closing of an elevator door, which can be circumvented by a malicious IoT implant, and in consequence, injure passengers. (4) *Exfiltrate Data*: A malicious IoT implant can eavesdrop data and commands on the serial bus and forward them via the implant's wireless interface to the attacker. This way, an attacker gains information about the current state of a device. Also, the attacker might be able to extract secrets, e.g., a passcode entered into a pin pad, or a production machine configuration that reveals a company secret.

4 Malicious IoT Implant

In this section, we present the design and implementation of the malicious IoT implant that is able to interfere with serial bus communication.

4.1 Design Criteria

To achieve its objectives, the attacker has certain design criteria regarding the malicious IoT implant: ① *Small Dimensions*: Size is a constraint as the implant has to be hidden inside the enclosure of the target device. In addition, small dimensions of an implant make detection harder. ② *Wireless Connectivity*: If the implant should be remotely controlled, it requires a radio transceiver. This transceiver should provide a communication interface to an LPWAN infrastructure such that physical presence of the attacker is not required. ③ *Access to Serial Communication*: The implant acts as a legitimate participant on the serial bus and is able to eavesdrop on legitimate transactions and to insert malicious transactions. ④ *Invisibility*: The implant does not influence the normal mode of operation except during an active attack. ⑤ *Low-Power*: The implant is either powered by an external power source, i.e., battery or accumulator, or supplied with power from the target device. To increase the lifetime of the implant as well as the target device, the implant should consume as less energy as possible. ⑥ *Low-Cost*: The implant should be designed in a low-cost way using mainly off-the-shelf components.

To the best of our knowledge, we are the first (in a scientific context) that design and implement an implant, which fulfills all of these design criteria. In Section 4.2, we present the attack procedures that interfere with the I^{2}C communication. In Section 4.3, we describe our approach to provide wireless connectivity to the implant over LPWANs. Finally, in Section 4.3, we describe the implementation of hardware and software, respectively.

4.2 Attack Procedures

To achieve the attacker's high-level objectives, we propose hardware-level attacks that interfere with the communication on the serial bus. To perform these procedures, the implant must be connected to the SDA and SCL signal lines of the target device.

Eavesdropping Eavesdropping is a passive attack in which the implant observes and stores data that is transmitted on the I^{2}C bus. This data can then be relayed to the attacker via the wireless interface of the implant.

Denial-of-Service A DoS disables all communication on the I^{2}C bus. A malicious IoT implant can perform such an active attack by permanently pulling the SDA and SCL lines to a low voltage state. As a result, no further data can be transmitted on the bus. All other bus participants have to wait until the implant releases the signal lines.

Injection of Transactions In this active attack, the implant acts as additional master on the bus. Most implementations offer time gaps between transactions, in which the masters and slaves are in idle state. The implant has the chance to execute own transactions on the bus during this period of time. The injection of own transactions allows to perform further implicit attacks: (a) *Read out memory and configurations*: The implant can read out data from memory chips as well as the configurations of slaves. These information can then be exfiltrated to the attacker via the wireless interface. (b) *Reconfiguration*: The implant can send commands to modify the configuration of slaves consistently. For example, a pre-configured threshold can be altered or, in some cases, a slave could be completely disabled. This ultimately allows for slave impersonation attacks, in which the implant responds to messages of the legitimate master instead of the disabled slave.

On-The-Fly Bit Modification Whenever a logical 1 is sent on the I^{2}C bus, the transmitting IC releases the SDA signal. A pull-up resistor connected to SDA then pulls the voltage of the signal to high level and the next clock signal carries the bit value. As an active attack, the implant can utilize this idle state to pull the SDA signal to low level, which results in the transmission of a logical 0 instead of the sent logical 1 on the bus. Due to the electronic characteristics of the I^{2}C bus, a modification of logical 0 to logical 1 is not possible.

4.3 Implementation

Wireless Connectivity. We use the LoRa technology (cf. Section 2.1) as wireless communication interface for the implant. Competing LPWAN standards to LoRa [1] exist, such as SigFox, Weightless, and LTE Narrowband IoT, but they are currently not supported by such a large community of industrial and private partners as LoRa. However, the presented attacks could also be facilitated using one of these LPWAN technologies.

TTN acts as service provider to connect the implant to the Internet using LoRa communication. Application builder can register an account at the TTN website and get access to the network infrastructure in order to connect to their deployed IoT nodes via LoRaWAN. An account can be created easily using a user name, email address and password. The purpose of the application is not checked by TTN.

Hardware Architecture. The hardware architecture of the implant consists of a PCB that is equipped with various ICs as shown in Figure 4. The implant can be connected to a power source that provides an input voltage between 3.3V and 16V. Power can be supplied via the VCC and GND pads, either from the target device or using a battery.

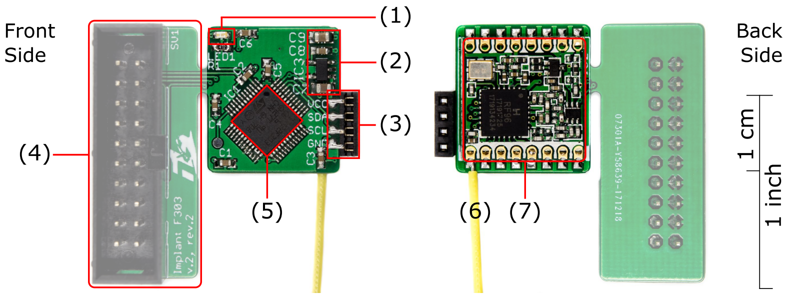


Fig. 4: Components of the malicious IoT implant: (1) indicator LED, (2) power converter, (3) I/O interface for serial bus signals (SDA, SCL) and power supply (VCC, GND), (4) removable programming and debug interface, (5) MCU, (6) wired monopole antenna, (7) LoRa radio transceiver.

The front side of the implant features a power converter, an I/O interface, an MCU, a number of capacitors, as well as an optional indicator LED. This LED is activated when the implant is supplied with power and blinks each time a LoRa message is sent or received. The MCU STM32F303CBT6 [40] contains an ARM Cortex-M4 core and 128 Kbytes of Flash memory. The radio transceiver RFM95W-868S2 [14] is mounted on the backside of the implant. This module supports the LoRa technology and uses the 868 MHz frequency band. We soldered a simple wired monopole antenna of length 86.4 mm (quarter of the 868 MHz wave length) to the transceiver.

For programming and debugging of the implant, a serial wire debug (SWD) interface is added to the implant. This interface can be physically removed (through breaking or cutting off) after the final version of firmware is installed on the implant.

Software Architecture. The software architecture is based on the STM32CubeMX platform [41] that includes the hardware abstraction layer and the link layer for the MCU. The real-time operating system FreeRTOS builds on top of this vendor-specific platform. A number of libraries is installed: The board support package provides drivers for the interfaces of the implant. The LMiC library [21] implements the LoRaWAN stack, and communicates with the LoRa module. The Arduino JSON library is used to decode and encode messages received within the payload of the LoRa messages. On top, so-called ‘tasks’ are defined. For example, the ‘attack task’ implements the attack procedures, while an ‘LED task’ defines the state of the indicator LED. The implant is registered as application belonging to the TTN account of the attacker and can be operated via the TTN web console.

5 Evaluation

Dimensions. Small dimensions are crucial in order to insert the implant into arbitrary target systems, and furthermore, to avoid visual detection. The implant has a size of 19.5x17.8 mm and a height of 4.5 mm. We measured the weight of the implant to be 3 grams. Note that these dimensions are measured without the debug header, antenna, and wires connected to the target. We assert that the dimensions of the implant are small enough for many threat scenarios, in which the enclosure provides a suitable amount of space. We assume that the layout of malicious IoT implants can be further minimized if we waive the usage of off-the-shelf hardware components.

Power Consumption. The malicious IoT implant has to be powered either by the power supply of the target device, or using an external battery. We determined that the power consumption of the implant during *sleep mode* (i.e., radio is duty cycling) is 110μA for 3.3V input voltage, while the implant consumes around 42mA in *attack mode* (i.e., radio listens continuously). For comparison: a regular 3.7V Lithium polymer battery with a capacity of 2000mAh supplies an implant in sleep mode for more than two years, or 176 hours in attack mode. Thus, attackers can wake a sleepy implant even months after the insertion into the target device.

Wireless Range. The wireless range determines from which distance an attacker is able to remotely control a malicious IoT implant. Also, it indicates in which areas the implant has coverage by an LPWAN. The implant utilizes the LoRa technology, which achieves a wireless range of 2-5 km in urban areas and up to 15 km in sub-urban areas [1]. It is hard to make general statements about the wireless range of the implant as the propagation of radio waves depends on many variables, e.g., the enclosure of the target device, building structures and walls, nearby electrical installations, as well as other deployed wireless networks that interfere with the LoRa frequency bands.

Cost. Once we have the final schematics, we are able to build a batch of 10 implants for the hardware costs of approximately 194 Euros. The cost per unit decrease with an increasing batch size: For a batch size of 100 units, the hardware cost add up to around 1075 Euros. Thus, we can build a malicious IoT implant using mainly off-the-shelf components for less than 11 Euros per unit (assuming a batch size of 100 units). These costs comprise the customized PCB as well as all electronic components including MCU, radio transceiver, LED, power converter, and capacitors. Not included are laboratory equipment, labor costs, shipping costs, and consumable materials.

5.1 Effort of Insertion

The procedure of implanting malicious hardware into the target device consists of three steps: identifying access points on the PCB, analyzing the communication on the bus, and inserting the implant into the device.

In the first step, we open the case of the target device and look whether there is enough space to insert the implant. If so, we identify the PCBs and list the descriptors of all ICs. Then, we search for the datasheets of these ICs on the Internet. The identification of ICs on a PCB can also be automated using image recognition [19]. A datasheet usually contains a feature description as well as a pin layout, which we use to identify ICs that support I^{2}C. After we confirm that an IC supports I^{2}C, we check whether the I^{2}C pins are used. Optical indications are signal lines on the PCB that are connected to these pins. Then, we look for suitable solder points on the PCB where we can later attach the wires to the implant. It is not advisable to directly solder the wires onto the pins of an IC since this requires a very precise way of working and can easily lead to damages or electrical shorts with neighboring pins. Good access points are larger solder joints, for example, at surface-mounted capacitors or at through-hole connections. As second step, we use a logic analyzer to inspect the communication on the bus. Using logic diagrams, we identify the ICs that communicate with each other, the bus frequency, and the transmitted data (datasheets might help again). As a result, we configure the software of the implant accordingly. In the third step, we solder wires onto the access points after we have removed the power supply and batteries. Then, we attach the wires to the implant. If required, we fixate the implant within the target device such that the antenna does not touch other electronic parts. We supply the target device with power again, and if the insertion was successful, the indicator LED on the implant turns on. In addition, we test whether the implant can be remotely controlled. Finally, we close the casing of the target device and try to remove all traces of this modification procedure.

The danger of damaging the PCB boards during the insertion of the implant is low if we take standard precautions: The process of insertion should be performed in an electrostatic discharge protected area. In this area, all conductive materials and workers are grounded and mechanisms to prevent the build-up of electrostatic charges should be in place. Furthermore, the power supply needs to be safely removed to prevent electrical shorts. Then, the danger of damaging the target device is mainly reduced to the threat of thermal influences on the ICs from the soldering process and physical damages.

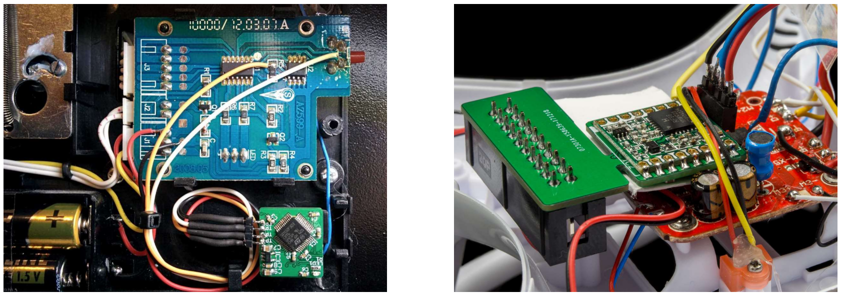
In the physical distribution process, time is crucial. Thus, the time to insert the implant into the target system should be appropriate. If we want to insert the implant into a large batch of similar target devices, the customization of the implant is only required once. From our experience, the process of customization can add up to a few hours. The insertion process needs to be performed for each target device. In our experiments, the manual inserting of the implants takes a few minutes, in some cases we were even able to insert the implant within less than a minute.

5.2 Feasibility of Attacks

We demonstrate the feasibility of the attacks outlined in Section 4.2 through inserting the malicious IoT implant into three exemplarily target devices: One evaluation board and two real-world products. We selected the real-world products through searching online in databases of disassembled products, e.g., iFixit, for security- and safety-relevant devices that indicate the usage of I^{2}C communication.

Evaluation Board. The first hardware platform is an evaluation board that was specifically designed to test the implementation of the implant. It imitates a monitoring application that observes the temperature of an industrial manufacturing process. If the temperature exceeds or undercuts a preconfigured threshold, an alarm is triggered and the light of an LED diode warns the operator. From a technical perspective, the MCU reads temperature sensor data from the registers of the sensor via I^{2}C, and shows the value on an LCD display. The lower and upper bounds of the temperature threshold are stored in the registers of the temperature sensor.

After attaching the implant to the SDA and SCL solder pads of the evaluation board, we are able to perform all attacks described in Section 4.2. During sleep mode, the implant does not interfere with the normal operation of the evaluation board. In attack mode, the implant eavesdrops the current temperature values as well as the threshold configuration, which both are requested multiple times per second by the MCU. The implant then relays these values to the attacker's operator interface. Upon receiving the DoS command from the attacker, the



(a) Cash box (b) Drone (implant with debug header)

Fig. 5: Malicious IoT implant (green PCB) inserted into exemplary devices.

implant disables all communication on the bus. On the target device, the MCU cannot read data from the sensor anymore and throws an exception, which results in a bus error message on the display. Furthermore, the implant can inject own transactions to read the data stored in the registers of the sensor, and to write new values to the registers. This way, the attacker is able to reconfigure the threshold that triggers the alarm. Finally, we are able to manipulate legitimate temperature values on the bus by performing the on-the-fly bit modification attack. Exemplary, we changed one bit of a temperature value byte such that the bus transferred 0x0F instead of 0x8F. As a result, the MCU reads a temperature of 15.9 °C instead of 28.9 °C.

Cash Box. As a second hardware platform, we inserted the implant into a First Alert 3040DFE cash box that allows access by entering a pin into an electronic pin pad. Each time a pin is entered into the pin pad, the MCU uses I^{2}C communication to read the master pin stored in an EEPROM. If the entered pin matches the master pin, the content of the cash box can be accessed. The master pin is set by pushing a red button that is located inside the cash box, and then entering the new master pin two times into the pin pad. Assuming that the attacker inserts the implant at some point during the physical distribution process, the attacker is later able to eavesdrop and set the master pin, and thus, to access the content of the box. As shown in Figure 5a, we attached the SDA and SCL wires of the implant to solder points of a pull-up resistor and the reset button, respectively. To supply the implant with power, we attached the VCC and GND wires to solder points connected to the batteries of the cash box. The enclosure of the cash box provides plenty of space for the implant and a wired monopole antenna. Controlling the implant from the operator interface, we performed eavesdropping, DoS, and the injection of transactions. First, the attacker is able to monitor the bus to retrieve the master pin that is requested by the MCU each time a pin is entered into the pin pad. The implant then exfiltrates the master pin via the wireless interface. In addition, the implant can disable all I^{2}C communication upon receiving a DoS command from the attacker. Then, the MCU cannot read the master pin from the EEPROM anymore, and thus, does not unlock the cash box. During the evaluation, we detected a manufacturerspecified modification of the I^2C bus: during idle times, the master constantly writes an oscillating signal on the SDA line. This signal clears for around 300ms after a button on the pin pad was pushed. Since the attacker cannot inject own transactions as long as the oscillating signal is occupying the SDA line, the implant has to wait to execute its transactions until the user pushes an arbitrary button on the pin pad. Through the injection of own commands, the attacker can then read the master pin from the EEPROM and also set this pin to an arbitrary value. During sleep mode, the implant does not influence the normal operation of the cash box.

Drone. We used a Syma X5C-1 drone as third hardware platform to evaluate the malicious IoT implant. The drone features a gyroscope and accelerometer sensor that stabilizes the drone during flights. An MCU reads data from this sensor every 3ms using I^2C communication, and subsequently adjusts the individual speed of the four rotors according to its flight position. As depicted in Figure 5b, we attached the SDA and SCL wires of the implant to a pin of the MCU as well as a pin of the sensor. Also, we attached the GND and VCC wires to solder points that are connected to the battery power supply of the drone. The body of the drone provides enough space for the implant and its antenna. Also, the drone is capable of carrying the implant without any effects on its flight characteristics. During sleep mode, the implant does not affect the normal operation of the drone. We performed eavesdropping and DoS attacks on the drone. Using the implant, the attacker can eavesdrop on the sensor data that is requested by the MCU. This sensor data contains triple-axis angular rates as well as triple-axis accelerometer data. Parts of these aggregated information can be sent to the attacker in regular intervals. Upon receiving a DoS command from the attacker, the implant blocks the I^2C bus through pulling both lines to low. The MCU of the drone cannot read data from the gyroscope and accelerometer, and thus, the speed of the rotors is not adjusted anymore. In consequence, the flight position of drone destabilizes and the drone hits the ground.

6 Discussion

The results of our evaluation underline two major threats: As a first threat, the emergence of IoT infrastructure provide novel attack vectors besides DoS attacks on Internet infrastructure [3,20]. As we demonstrate, malicious IoT implants connected to LPWANs can be leveraged to exfiltrate secret information, manipulate the functionality of target devices, and in worst case, might even pose a threat to humans. Such attacks can be performed anonymously as one can register an account and set up the application without any identification at the website of the LoRaWAN service provider TTN. Furthermore, the attacker can control the implant from a remote location over the Internet. These attacks are not specific to LoRaWAN and can also be performed using other competing LPWAN standards. We note that the usage of traditional mobile telephony infrastructure (e.g., GSM and LTE) would not satisfy the design criteria given in Section 4.1 since a GSM or LTE radio transceiver consumes more energy, the attacker would have to pay for data transmissions, and in most countries a SIM card registration requires an official identification document. The effort of building such an implant is relatively low for experts since the hardware and software design is based mainly on off-the-shelf components and open-source software, respectively. Thus, the dissemination of LPWANs open up new attack vectors, which did not exist before when traditional mobile telephony infrastructure was the only wide-area connectivity provider.

As a second threat, serial communication on PCBs is vulnerable to malicious hardware inserted during physical distribution in the supply chain. While the presented malicious IoT implant is tailored to attack I^2C buses, other serial communication systems, such as UART and SPI, could be adapted with a reasonable effort. However, we might only be able to apply a subset of the presented attacks to other bus systems due to different approaches in the electronic design of these systems. In contrast to other serial buses, I^2C facilitates a communication protocol that allows multiple masters on the bus. Since the implant acts as a master, the injection of own transactions in SPI and UART communication is not easily possible. Nevertheless, we can eavesdrop the communication between ICs to exfiltrate information and perform DoS attacks through pulling all lines of the communication system to a low voltage state. In our evaluation, both attacks had a significant impact on the target devices' security and reliability.

One might ask why should attackers use malicious IoT implants when malicious software (malware) could do the same job? Although we agree that the effort of facilitating malware might be lower, malware falls short in several scenarios. First, if the target device has no Internet connection, then malware has usually no communication channel to the attacker. For this reason, neither of our three evaluation devices could be remotely attacked using malware due to missing network interfaces. Second, in case a direct interference with serial communication on hardware level is desired, e.g., to circumvent software protection mechanisms. Third, malware could be detected by other software, in contrast to implants that are "invisible" at software level. During the evaluation, the implant had no influence on the regular operation of the target device except if the attacker performs an attack. Since the attacks directly influence the communication on hardware level, an incident investigator is not able to find digital traces in the log files of the target device's software. The only indications might be exceptions triggered by the MCU and physical evidence, e.g., the presence of an implant or traces on the PCB that indicate that an implant was attached.

So far, malicious IoT implants have been considered neither in theoretical hardware security models, nor in practical approaches to secure hardware against malicious modifications. Since we demonstrated feasibility of these threats, we conclude that future hardware security efforts have to take implants into account.

6.1 Limitations

The threat of LPWAN-connected malicious IoT implants comes with a number of limitations for attackers. Each implant needs to be inserted manually, which renders this attack procedure unsuitable for large-scale operations in which thousands of devices have to be modified. Furthermore, expert knowledge in electronic engineering and programming of software is necessary for the preparation and insertion of an implant. Moreover, a number of potential target devices, e.g., mobile phones and tablets, might not provide enough space within the enclosure to carry an implant that is designed using mainly off-the-shelf components. Also, the feasibility of utilizing an LPWAN-connected implant is limited through the coverage of the selected service provider's LPWAN infrastructure. Finally, the amount of exfiltrated data is restricted since LPWANs only provide low data rates to achieve their low-power objectives. Nevertheless, the bandwidth between implant and attacker is reasonable for most threat scenarios.

6.2 Countermeasures

We analyze a variety of potential approaches to encounter malicious IoT implants, which we divide into detection and safeguard mechanisms. While detection mechanisms disclose the presence of a malicious IoT implant in a system, safeguard mechanisms prohibit an implant from interfering with the serial communication.

Detection Mechanisms. A trivial approach to detect malicious IoT implants is *visual inspection* of the PCBs. The advantage is that no expensive equipment is required. On the other hand, this requires the removal of the enclosure for most products, which could be quite a cumbersome task since many products are not intended to be disassembled. Therefore, this approach becomes impractical if large batches of products should be investigated. Also, future implant layouts might become smaller and can be implemented into PCBs hidden as legitimate ICs, which makes visual detection much harder and more time-consuming. In addition, non-expert user might not be able to recognize malicious hardware elements if the implant is camouflaged as a legitimate part of the PCB.

Since malicious IoT implants have a physical appearance, another detection approach is to *compare the weight* of suspicious products with the weight of an evidently unmodified product. The advantage of this approach is low costs as only a precision scale is needed. The disadvantage is that an attacker can potentially reduce the weight of a modified device by removing small pieces of the enclosure. Also, this approach is not suitable for heavy devices since the weight of the implant might be hidden within the measurement tolerance.

In *anomaly detection*, potential side-channel effects resulting from the presence of an implant are observed. For instance, the implant consumes a certain amount of power as evaluated in Section 5, which might be supplied from the host system. Thus, the power consumption of manipulated products should show anomalies compared to unaltered products. Also, malicious IoT implants provide a wireless interface that emits radio waves, which can be detected with special equipment. The advantage of anomaly detection procedures is potential large-scale automation. The disadvantage is the need for hardware extensions on the products or special equipment in testing facilities.

Safeguard Mechanisms. Another way to protect against the insertion of malicious IoT implants is the adding of *tamper-evident features*. For example, the packaging of a product can be sealed in way that the attacker cannot access the product without irreversibly destroying the sealing. Also, physical security measures, such as a locked encasement or resin encapsulation, could be in place to protect the PCB against tampering. Tamper resistance does not always prevent the implementation of an implant but it increases the attacker's effort and makes the detection of malicious actions much more likely.

The usage of *cryptographic security measures* can be a countermeasure to circumvent malicious IoT implants to read and inject messages into the serial buses. Lazaro et al. [23] proposed an authenticated encryption scheme for I^{2}C buses. In their proposal, the I^{2}C data frames are encrypted and authenticated using AES-GCM, while addressing frames are not protected. The calculation of ciphertext and signature is directly implemented into the master and slave ICs. The authors assume a pre-installed key on each IC that was installed in a secure environment. The advantage of encryption is that it provides an efficient way to lock out non-authorized entities. As a disadvantage, all ICs on the bus must implement the encryption mechanism and need to be equipped with key material. Most probably, this requires a change of the I^{2}C specifications.

Shwartz et al. [38] propose the idea of a hardware-based *interface proxy firewall* to protect I^{2}C buses against malicious hardware. Unfortunately, they do not present a technical concept of their idea such that a design of this firewall remains future work. From our perspective, the challenge of this firewall is to distinguish between legitimate and malicious bus participants. Since a malicious participant can easily spoof a legitimate participant, a simple black list or white list approach is not effective. To protect against this threat, an authentication infrastructure or physical security measures are needed. As an advantage, this firewall would not need to be part of the official I^{2}C specifications. On the backside, we need additional hardware on the PCB to implement the firewall.

Oberg et al. [32] observed information flows in the I^{2}C bus system by applying taint tracking. After identifying explicit and implicit information flows, they proposed to add an *adapter* to each slave device that is placed between this device and the bus. These adapters coordinate access to the slave devices by allowing only access to one device at any given point of time. We note that these proposals only consider passive attackers but not active attackers. Thus, using a malicious IoT implant, it is still possible to manipulate data on the bus since the implant has no adapter that controls the access to the bus. The advantage of this approach is that these adapters do not have to be specified in the I^{2}C standard. The disadvantage is the need for additional hardware components on the PCBs that increase the space requirements, cost, and energy consumption.

7 Related Work

Previous research investigated the insertion of malicious hardware at three stages: in the design phase, during fabrication phase, and in the post-fabrication phase.

Especially hardware trojans attracted a high amount of research in the last decade. From a high level perspective, hardware trojans are malicious modifications of the hardware during the design or fabrication process. In contrast, malicious hardware implants are alien elements that are added to a system after the fabrication process.

There exist different approaches to insert malicious trojans into hardware. An approach are modifications of the system design at hardware description language (HDL) level, which results in the adding of additional logic to the IC. Prototypes of these trojans have been mainly implemented and evaluated using field programmable gate arrays (FPGAs). The threat of malicious hardware trojans was first shown and evaluated by Agrawal et al. [2]. They also proposed a detection mechanism based on side-channel fingerprinting. King et al. [18] introduced hardware trojans that are able to gain unchecked memory access as well as to execute malicious firmware on the target. Lin et al. [24] presented a hardware trojan that provides physical side-channels to exfiltrate cryptographic material from an IC. Hicks et al. [13] proposed unused circuit identification (UCI), a method to identify and remove suspicious circuits using data flow graph analysis. A year later, Sturton et al. [42] presented a prototype of a hardware trojan that defeats the UCI detection mechanisms. Fern et al. [8] used hardware trojans to build a covert communication channel between different components in a system-on-a-chip. Gómez-Bravo et al. [12] presented a hardware trojan that attacks I^2C communication, targeting a mobile robotic application. Another approach of inserting malicious hardware is the implementation of hardware trojans at gate level during fabrication. In contrast to modifications at HDL level, this approach does not add additional logic to the system but only modifies existing hardware elements. Shiyanovskii et al. [37] introduced lifetime-reducing reliability trojans, which induce aging effects resulting from alternations of the fabrication processes. Becker et al. [5] demonstrated a variant, in which a hardware trojan is implemented at gate level by manipulating the dopant polarity of existing transistors. Kumar et al. [22] used hardware trojans to inject faults during the execution of a lightweight cipher, enabling them to retrieve secret keys. This hardware trojan was also induced by altering the dopant area at gate level. A final approach of inserting malicious hardware is the adding of analog circuits to the system. The concept of an analog hardware trojan was introduced by Yang et al. [43]. They demonstrated that an attacker is able to insert analog circuits into a system at fabrication time.

The first ICs that relate to hardware implants were called mod chips [36], which modify functions of the target system, e.g., to circumvent copyright protection mechanisms in video playback devices or to enable restricted features in game consoles. Compared to design and fabrication phase attacks, less attention was paid by the academic community to malicious hardware attacks in post-fabrication phases. Shwartz et al. [38] demonstrated how aftermarket components, e.g., third-party touchscreens used in repairs of broken mobile devices, could be manipulated such that a malicious mobile phone app can get root access to the device. In a non-academic context, Datko and Reed [7] implemented a hardware implant inspired by the NSA Ant catalog [4]. Their proof-of-concept features a GSM interface to ex-filtrate data and connects to the target system via a VGA display adapter using I^{2}C communication. To relay data from the computer, a malware on the target system is assumed that sends data via I^{2}C to the implant. In contrast to our work, this implant does not fulfill design criteria ① and ⑥. FitzPatrick [10] presented a number of proof-of-concepts for hardware implants that connect to targeted systems via I/O pins or JTAG. These implants are able to perform privilege escalations as well as turn on and off I/O pins. Although these implants fulfill most design criteria, they lack a communication interface to an IoT or cellular infrastructure (②).

8 Conclusion

In this paper, we described the implementation and evaluation of the first malicious IoT implant showing that IoT infrastructure enables novel hardware-level attack vectors. These threats grow with the expansion of LPWANs, which will supersede mobile telephony networks in terms of providing M2M connectivity in a few years. Future threat models for hardware security have to take these threats into account.

Acknowledgments. We thank Tobias Groß for helpful comments. This work was supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest project.

References

1. F. Adelantado, X. Vilajosana, P. Tuset-Peiró, B. Martínez, J. Melià-Seguí, and T. Watteyne. Understanding the limits of LoRaWAN. *IEEE Communications Magazine*, 55(9), 2017.
2. D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan detection using IC fingerprinting. In *IEEE Symposium on Security and Privacy, S&P 2007*.
3. M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou. Understanding the Mirai botnet. In *26th USENIX Security Symposium, USENIX Security 2017*.
4. J. Appelbaum, J. Horchert, and C. Stöcker. Shopping for spy gear: Catalog advertises NSA toolbox. *Spiegel Online International*, 29, 2013.
5. G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson. Stealthy dopant-level hardware trojans. In *CHES 2013*.
6. J. Boyens, C. Paulsen, R. Moorthy, N. Bartol, and S. A. Shankles. Supply chain risk management practices for federal information systems and organizations. *NIST SP 800-161*, 2015.
7. J. Datko and T. Reed. NSA Playset: DIY hardware implant over I2C. DEF CON 22, 2014.

8. N. Fern, I. San, Ç. K. Koç, and K. Cheng. Hardware trojans in incompletely specified on-chip bus systems. In *Design, Automation & Test in Europe Conference & Exhibition, 2016*.
9. E. Fernandes, J. Jung, and A. Prakash. Security analysis of emerging smart home applications. In *IEEE Symposium on Security and Privacy, S&P 2016*.
10. J. FitzPatrick. The Tao of hardware, the Te of implants. Black Hat USA, 2016.
11. Gartner. Gartner says 8.4 billion connected ”things” will be in use in 2017, up 31 percent from 2016, Feb. 2017. http://www.gartner.com/newsroom/id/3598917.
12. F. Gómez-Bravo, R. Jiménez-Naharro, J. M. García, J. A. G. Galán, and M. Sanchez-Raya. Hardware attacks on mobile robots: I2C clock attacking. In *Robot 2015: Second Iberian Robotics Conference - Advances in Robotics*, 2015.
13. M. Hicks, M. Finnicum, S. T. King, M. M. K. Martin, and J. M. Smith. Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically. In *IEEE Symposium on Security and Privacy, S&P 2010*.
14. HopeRF Electronic. RFM95/96/97/98(W) – low power long range transceiver module V1.0 datasheet. http://www.hoperf.com/upload/rf/RFM95_96_97_98W.pdf.
15. G. Hunt, G. Letey, and E. Nightingale. The seven properties of highly secure devices. Technical report, March 2017.
16. IC Insights. NXP acquires Freescale, becomes top MCU supplier in 2016, April 2017.
17. Kerlink. Kerlink continues global expansion with subsidiary in India for rollout of world’s largest LoRaWAN IoT network, September 2017.
18. S. T. King, J. Tucek, A. Cozzie, C. Grier, W. Jiang, and Y. Zhou. Designing and implementing malicious hardware. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats, LEET ’08*, 2008.
19. S. Kleber, H. F. Nölscher, and F. Kargl. Automated PCB reverse engineering. In *11th USENIX Workshop on Offensive Technologies, WOOT 17*, 2017.
20. C. Kolias, G. Kambourakis, A. Stavrou, and J. M. Voas. DDoS in the IoT: Mirai and other botnets. *IEEE Computer*, 50(7):80–84, 2017.
21. M. Kooijman. Arduino LoraMAC-in-C (LMiC) library. https://github.com/matthijskooijman/arduino-lmic.
22. R. Kumar, P. Jovanovic, W. P. Burleson, and I. Polian. Parametric trojans for fault-injection attacks on cryptographic hardware. In *Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2014*.
23. J. Lázaro, A. Astarloa, A. Zuloaga, U. Bidarte, and J. Jimenez. I2CSec: A secure serial chip-to-chip communication protocol. *Journal of Systems Architecture - Embedded Systems Design*, 57(2):206–213, 2011.
24. L. Lin, M. Kasper, T. Güneysu, C. Paar, and W. Burleson. Trojan side-channels: Lightweight hardware trojans through side-channel engineering. In *CHES 2009*.
25. LoRa Alliance. LoRa Alliance surpasses 500 member mark and drives strong LoRaWAN protocol deployments, June 2017.
26. LoRa Alliance. LoRaWAN global networks – where are we today?, October 2017.
27. Machina Research. With 3 billion connections, LPWA will dominate wide area wireless connectivity for M2M by 2023, February 2015.
28. J. Margulies. Garage door openers: An Internet of Things case study. *IEEE Security & Privacy*, 13(4):80–83, 2015.
29. H. Min and G. Zhou. Supply chain modeling: past, present and future. *Computers & Industrial Engineering*, 43(1):231 – 249, 2002.
30. P. Morgner, S. Mattejat, Z. Benenson, C. Müller, and F. Armknecht. Insecure to the touch: attacking ZigBee 3.0 via touchlink commissioning. In *Proceedings of the*

10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2017.

31. NXP. The I2C-bus specification and user manual – UM10204, April 2014.
32. J. Oberg, W. Hu, A. Irturk, M. Tiwari, T. Sherwood, and R. Kastner. Information flow isolation in I2C and USB. In *Proceedings of the 48th Design Automation Conference, 2011*.
33. C. Reichert. NNN Co and Actility announce LoRaWAN network rollout across Australia, February 2017.
34. E. Ronen, C. O'Flynn, A. Shamir, and A. Weingarten. IoT goes nuclear: Creating a ZigBee chain reaction. In *IEEE Symposium on Security and Privacy, S&P 2017*.
35. M. Rostami, F. Koushanfar, J. Rajendran, and R. Karri. Hardware security: Threat models and metrics. In *The IEEE/ACM International Conference on Computer-Aided Design*, 2013.
36. R. Safavi-Naini. *Digital Rights Management: Technologies, Issues, Challenges and Systems*, volume 3919. Springer Science & Business Media, 2006.
37. Y. Shiyanovskii, F. G. Wolff, A. Rajendran, C. A. Papachristou, D. J. Weyer, and W. Clay. Process reliability based trojans through NBTI and HCI effects. In *2010 NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2010*.
38. O. Shwartz, A. Cohen, A. Shabtai, and Y. Oren. Shattered trust: When replacement smartphone components attack. In *11th USENIX Workshop on Offensive Technologies, WOOT 17*, 2017.
39. Sigfox. SIGFOX expanding IoT network in 100 U.S. cities., February 2017.
40. STMicroelectronics. STM32F303CB datasheet, May 2016.
41. STMicroelectronics. STM32Cube initialization code generator datasheet, July 2017.
42. C. Sturton, M. Hicks, D. A. Wagner, and S. T. King. Defeating UCI: Building stealthy and malicious hardware. In *IEEE Symposium on Security and Privacy, S&P 2011*.
43. K. Yang, M. Hicks, Q. Dong, T. M. Austin, and D. Sylvester. A2: Analog malicious hardware. In *IEEE Symposium on Security and Privacy, S&P 2016*.