SOVEREIGNTY IN INFORMATION TECHNOLOGY

Arnd Weber
Steffen Reith
Michael Kasper
Dirk Kuhlmann
Jean-Pierre Seifert
Christoph Krauß

SECURITY, SAFETY AND FAIR MARKET ACCESS BY OPENNESS AND CONTROL OF THE SUPPLY CHAIN

WHITE PAPER V1.0, MARCH 2018

CONTACT INFORMATION

Contact
Arnd Weber: arnd.weber@kit.edu
Steffen Reith: Steffen.Reith@hs-rm.de
Michael Kasper: michael.kasper@fraunhofer.sg
Dirk Kuhlmann: dirk.kuhlmann@alumni.tu-berlin.de
Jean-Pierre Seifert: jpseifert@sect.tu-berlin.de
Christoph Krauß: christoph.krauss@sit.fraunhofer.de

Fraunhofer Singapore
Nanyang Technological University (NTU)
50 Nanyang Avenue, Level 5-5, SG 639798, Singapore

Fraunhofer Institute for Secure Information Technology (SIT)
Rheinstrasse 75, D 64295 Darmstadt, Germany

Hochschule RheinMain
Unter den Eichen 5, D 65195 Wiesbaden, Germany

Karlsruher Institut für Technologie
Institut für Technikfolgenabschätzung und Systemanalyse
Karlstrasse 11, D 76133 Karlsruhe, Germany

Technische Universität Berlin
Institut für Softwaretechnik und Theoretische Informatik
Ernst-Reuter-Platz 7, D 10587 Berlin, Germany

Project Website and URL for Download
http://www.QuattroS-Initiative.org/
https:/www.QuattroS-Initiative.org/whitepaper_latest.pdf

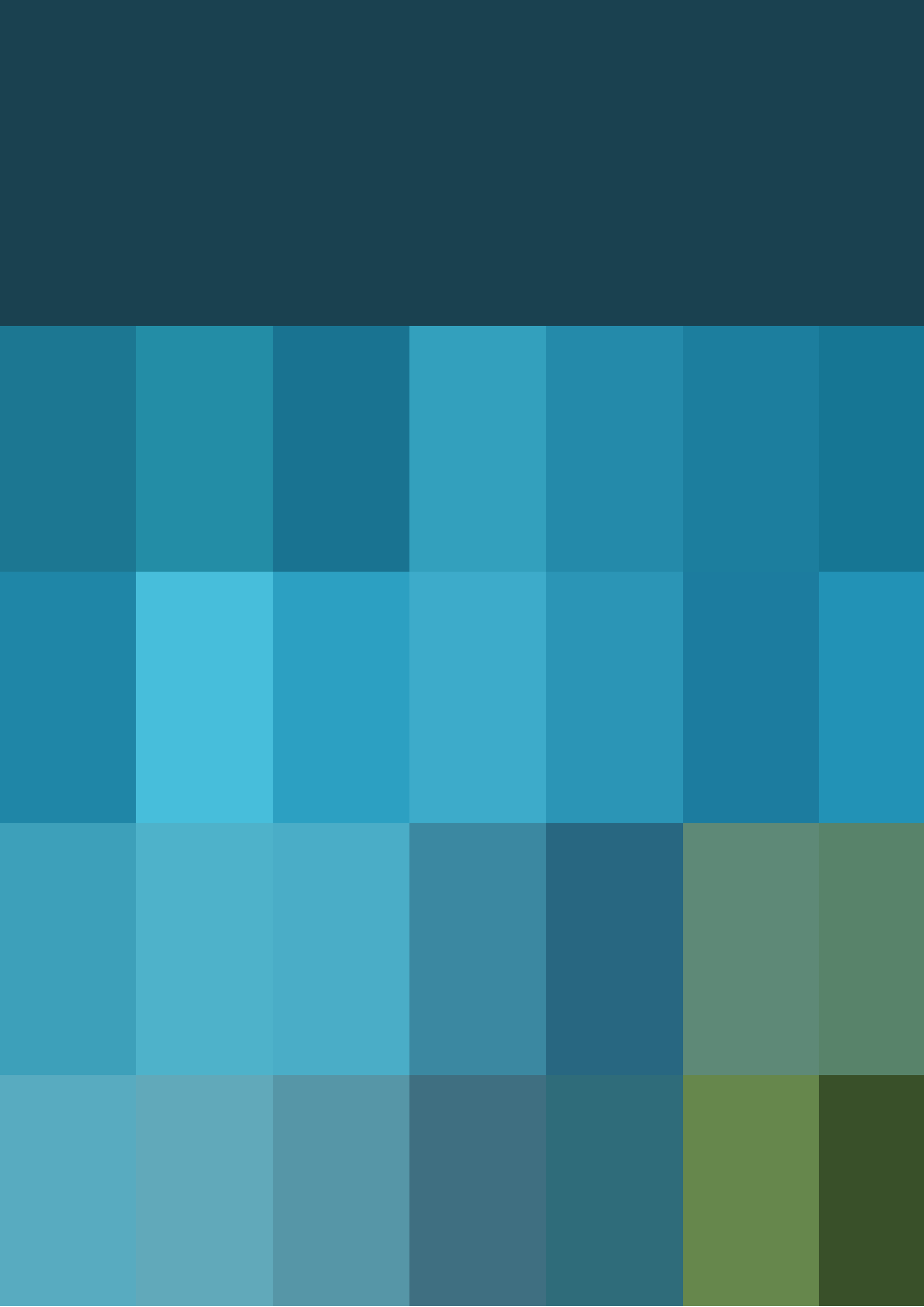
Please quote as:

Arnd Weber, Steffen Reith, Michael Kasper, Dirk Kuhlmann, Jean-Pierre Seifert, and Christoph Krauß: Sovereignty in Information Technology. Security, Safety and Fair Market Access by Openness and Control of the Supply Chain. 2018. http://www.QuattroS-Initiative.org/

Frontpage:

Royalty free image, source: Pixabay

TABLE OF CONTENTS



ABSTRACT

Digital Sovereignty by Openness and Control of the Information Technology Supply Chain

Digital sovereignty is the capability of nation states, societies and communities to control essential operational features with regard to the information technology they use. In particular, this includes the security characteristics of these components. Digital sovereignty can be supported by employing a continuously growing share of validated IT components with verified – and ideally proven and certified – quality and security assurances.

The safety and security of IT elements has become a major issue in international markets for IT. In theory, this should give competitive advantages to producers who supply security-hardened and certified IT elements. However, the question on how to determine the actual trustworthiness of security-certified components and modules has presented a long standing problem. To underpin the trustworthiness of security assurances, it should ideally be possible to independently validate the certified claims.

One way towards this goal would be to open up the whole IT supply chain – from the application software down to the hardware including all tools used to produce them – in such a way as to employ open development tools, processes and validation methods. Research on open tools of this kind, including those for hardware and silicon manufacturing, is already in process, but the transition towards an open development and production process will take some time.

An alternative to this open path consists in attempting to gain control of the supply chain by purchasing suppliers and manufacturing plants on an international scale. Such attempts are increasingly becoming visible in the IT, mechanical engineering and telecommunications industries.

Thrust towards more secure systems could also emerge from governments defining and implementing a regulatory framework intended to increase liabilities for producers of products that are faulty or have backdoors. Such a framework could make product certification and formal proofs of their correctness mandatory, at least in key fields of IT applications.

Enthusiasts and industry can support open processes by promoting global discussions on the merits of the new approaches, by helping to implement them, and by carrying out research on their practical viability.

Researchers from the Fraunhofer Institute (SIT), Fraunhofer Singapore, the Karlsruhe Institute of Technology (ITAS), the Rhein-Main University of Applied Sciences and the Technical University of Berlin/T-Labs have jointly written a White Paper on the utilization of open methods and components for enhancing the state of IT security. The authors give an overview of current problems con-

“WITH TODAY’S TECHNOLOGIES, THERE IS NO ALTERNATIVE TO HAVING A SECURE SUPPLY CHAIN.”

GERNOT HEISER, DATA61, AUSTRALIA

fronting the global information technology supply chain. They analyse the risks arising from newly emerging malware using “zero-day” exploits, “Trojan horses” produced by nation states which stealthily reside in software and hardware, stolen chip designs or counterfeit electronic components.

These and related problems threaten not only the security, but also the safety of many national IT infrastructures and products. The authors presume that the current division of labour – with the US producing most of the software and China producing most of the hardware – and the lack of competition resulting from it, not only increases the likelihood of insecure products, but is also economically problematic for other regions.

Based on this analysis, the authors discuss options to address these problems. Since IT security spawns across all sectors – hardware, software, protocols, systems, services and infrastructures – the only means to improve the general protection level is deemed to retain and, where necessary, to regain control of the supply chain in its entirety.

The authors conclude that in order to prepare against future attacks, it is not sufficient to just update security tools. Ultimately, any reliance on closed hardware components means relying on black boxes, which will, by necessity, constrain attempts to build security tools on top of them in a perfect way. However, free and open hardware is already emerging and will soon be employed by major vendors such as Nvidia and Western Digital. It can be anticipated that more industries, in order to obtain more flexibility and control, will apply the new approach, much like they adopted Linux and Android for the same reason.

To turn the concept of open security components into reality, it is necessary to promote and foster the global discussion about the nature of the underlying problems and about viable strategies to address them. Hardware and software prototypes should be designed and built in an open process by industry and academia, and these components should be manufactured in trustworthy plants. First instances of open security components could target small systems in the automotive field or infrastructures. A migration path could lead from these and similar embedded solutions, via architectures that peer up closed and open hardware and software, towards the paradigm of open security being adopted for the production of mainstream IT components.

PREFACE

This White Paper contains suggestions for improving the information technology supply chain, with regard to IT products, with regard to the security and safety of products using IT as well as with regard to having a fairer distribution of the value added.

The technical suggestions can be described as proposals for securing the entire supply chain (because if higher layers were secured, attacks would take place at lower layers). The measures should therefore include not only the software, but also the hardware and the tools used in its production. Security- or safety-relevant components and systems should be open, if needed certified and, ideally, formally verified. In principle, such steps can be made mandatory with regard to closed systems and confidential parts. The paper argues, however, that open components would help significantly in making security-relevant components secure. One consequence of such openness is that any layer can be precisely specified and verified. Furthermore, the White Paper also discusses how the objectives can be met. A community should emerge that produces open components and makes a business of selling the hardware and related services; this process has already started. Governments could support such efforts by, for instance, subsidizing the production of prototypes. Governments could also support the transition by enacting the appropriate legislation.

If the approaches proposed here were pursued at a global scale, supply chains would be more resilient. Income would be distributed in a fairer way, i.e. not concentrated in essentially two countries. In the longer run, a world of more secure and even proven systems would emerge. It can be expected that the process will proceed despite interruptions due to flaws being discovered in the newly produced components and due to the intervention of nation states and of incumbent investors producing closed components. The latter might try to prevent a disruption of their business processes.

This White Paper contains two versions of its content. The first is a short version which omits many technical details, but provides the big picture to interested readers such as investors, policy makers or members of the general public. The second version contains the arguments in a comprehensive way, with all necessary details for readers such as company representatives or computer scientists interested in implementing some of the approach.

ACKNOWLEDGEMENTS

This paper would not have come into being without the requirements for secure computers that Andreas Pfitzmann† expressed to Arnd Weber in Karlsruhe, Germany, back in 1988. With regard to the present text, the authors are indebted to valuable input and comments made by Paul Duplys, Dirk Fox, Torsten Grawunder, Gernot Heiser, Armand Puccetti, Arnaud Saffari, Dirk Stegemann, Michael Waidner and Michael Wilson. Responsibility for errors rests, of course, with the authors.

ABOUT THE AUTHORS

Michael Kasper is head of the department Cyber- and Information Security at Fraunhofer Singapore, Singapore and is associated as senior researcher with Fraunhofer Institute for Secure Information Technology (SIT). He co-headed the department of Cyber-Physical Systems Security at the Fraunhofer Institute for Secure Information Technology until January, 2017 and has more than 20-years experience in information security and computer science.

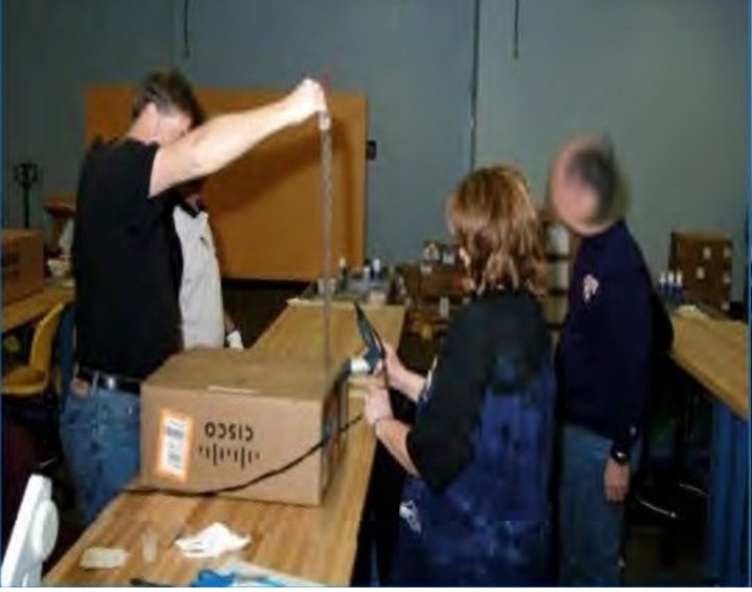
Prof. Christoph Krauß is head of the department Cyber-Physical Systems Security at Fraunhofer Institute for Secure Information Technology (SIT), Germany and professor for network security at University of Applied Sciences Darmstadt.

Dirk Kuhlmann is Technical Consultant for the UK company Tilix. His current work concerns secure distributed IT systems for supporting decentralized energy production and distribution. From 1995 to 2017, he was employed as a researcher at Hewlett Packard Laboratories Bristol, UK, focussing on the utilization of Open Source based components for improving platform and system security.

Prof. Jean-Pierre Seifert is chairing the field „Security in Telecommunications“ at TU Berlin and at Telekom Innovation Laboratories, Berlin, Germany. He has been working in research on hardware security at Infineon (Munich), Intel (Portland) and Samsung (San Jose) in the USA. In 2002 Prof. Seifert has been honoured by Infineon with the award “Inventor of the Year” and has received as well several Intel Achievement Awards in 2005 for his new CPU security instructions for the Intel microprocessors.

Prof. Steffen Reith is a professor for theoretical computer science with RheinMain University of Applied Sciences, Wiesbaden, Germany. During his career in industry he developed cryptographic functionalities which are still in mass production in newly built cars.

Dr. Arnd Weber is an economist, with a PhD in sociology. He is Senior Researcher with the Institute for Technology Assessment and Systems Analysis of Karlsruhe Institute of Technology, Germany. He has worked in various IT-related research projects, including some on behalf of the European Commission and of the European Parliament.



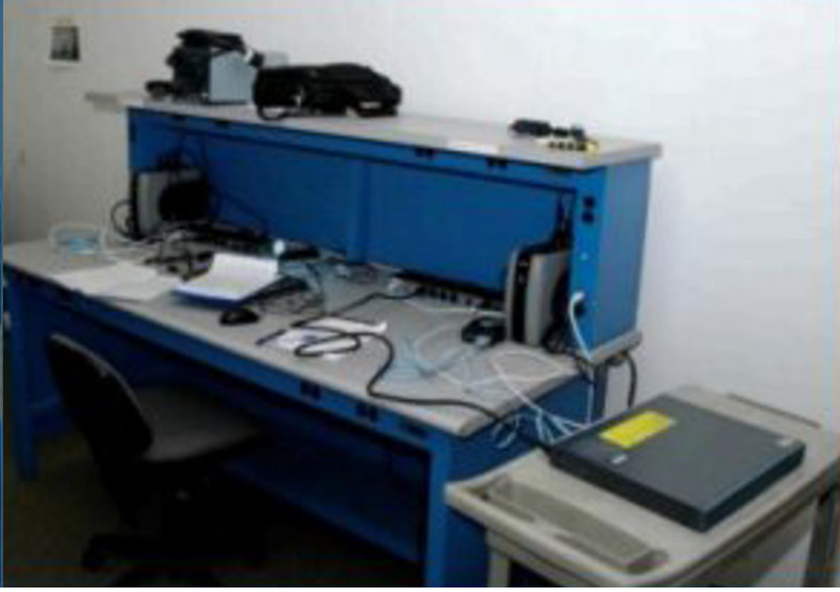


Fig. 1a: Opening of a Cisco parcel by NSA employees, for modifying the hardware and installing implants (from the Snowden-documents, Leaksource 2013).

Fig 1b: Load Station designed specifically for installation of tailored implants directly into targets' electronic devices.

INTRODUCTION AND SUMMARY FOR THE GENERAL PUBLIC

This White Paper contains suggestions for improving the information technology supply chain, specifically with regard to the security and safety of products, as well as with regard to having cheaper, fairer and more resilient supply chains. Suggestions for actions are also included. This document is written from the perspective of those who do not live in a country which dominates the current supply chains, in other words from that of private, business and government users such as those in Europe, Southeast Asia or Latin America, but with the intention to point out paths which are beneficial for the whole world. After an introduction, in which the authors' motivations and methods are briefly described, problems confronting the IT supply chain are reviewed in **Section 2**, such as:

- Eavesdropping on business secrets, passwords, etc.: As one interviewed expert put it: "All intellectual property is available to whoever wants to take it". This includes physical attacks as well as pre-implanted "computer network exploitations" (CNE in NSA parlance). Snowden revealed that 70,000 systems had been infiltrated (cf. Fig. 1 for a physical attack). Meanwhile, more weaknesses may have been implemented, by what-ever nation state, and be used and abused by malicious insiders or knowledgeable criminals. The recent hardware bugs Meltdown and Spectre in, e.g. Intel processors, may be used to eavesdrop on any kind of data, on PCs as well as on servers (Lipp et al. 2018, Kocher et al. 2018).

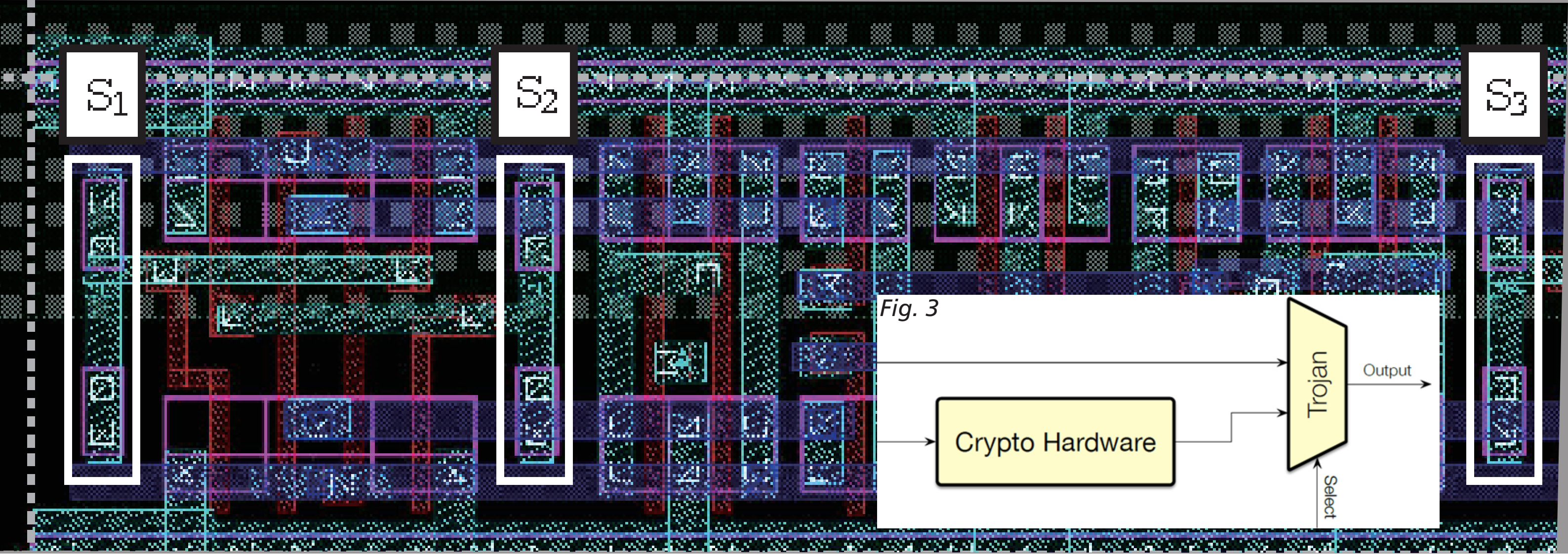
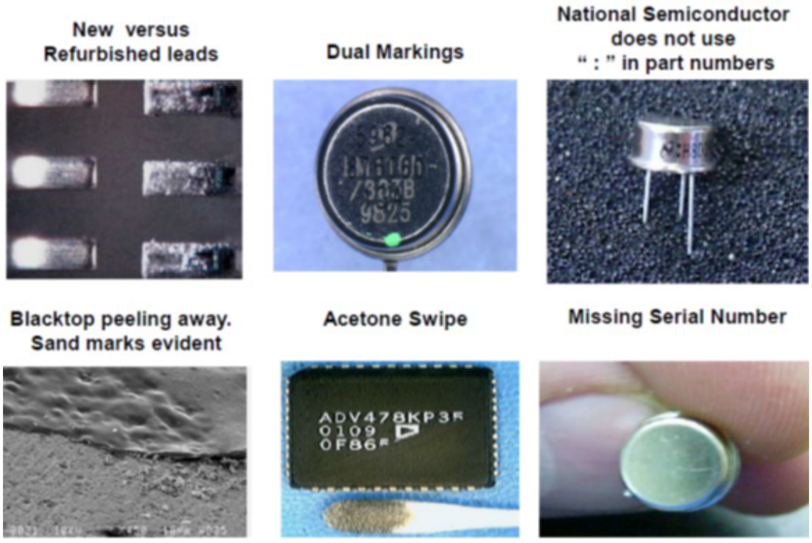


Fig. 2: Reverse engineered area of a stealthy dopant-level hardware Trojan (Sugawara, 2014).

Fig. 3: The schematic design of a hardware Trojan horse bypassing encryption (cf. Rajendran et al. 2010).

- Disruptions and sabotage, for example botnets producing malicious floods of messages bringing down the Internet: “Someone is learning how to take down the Internet” (Schneier 2016). Disruptions of processes via zero-day attacks and “computer logic bombs“ (Liang, Wang 1999) are possible. Ransomware using zero-day attacks, including known and unpatched vulnerabilities is another issue (e.g. WannaCry). Kill switches in, for instance, critical infrastructure or arms which could be switched on and, after use, switched off again. Parts of chips, not only tamper-resistant ones, but also mainstream processors, are confidential. For instance, certain details of US processors are only available to US citizens. This poses a hard to judge risk; it is not unlikely that hidden instructions for an undocumented backdoor can be identified. Therefore they can be abused for espionage or sabotage (cf. Cimpalu 2017, Domas 2017).
- Scope of security issues: Also addressed in this report are the stealthiest attacks, which need to be anticipated regardless of whether they originate from imperfect software or even from hardware. Consider the novel types of hardware Trojan horses using dopant-level or capacitor effects (cf. Becker et al. 2014 and Yang et al. 2016; see Fig. 3 for the schematic design of a hardware Trojan). In this way, even the actions of nation states and risks of uncertain likelihood are taken into account, thus also preparing for an increase in global political tensions. It is also argued that if a layer is secured, such as the communications layer via encryption,

Fig. 4: Examples of counterfeit parts (Hughitt 2010).



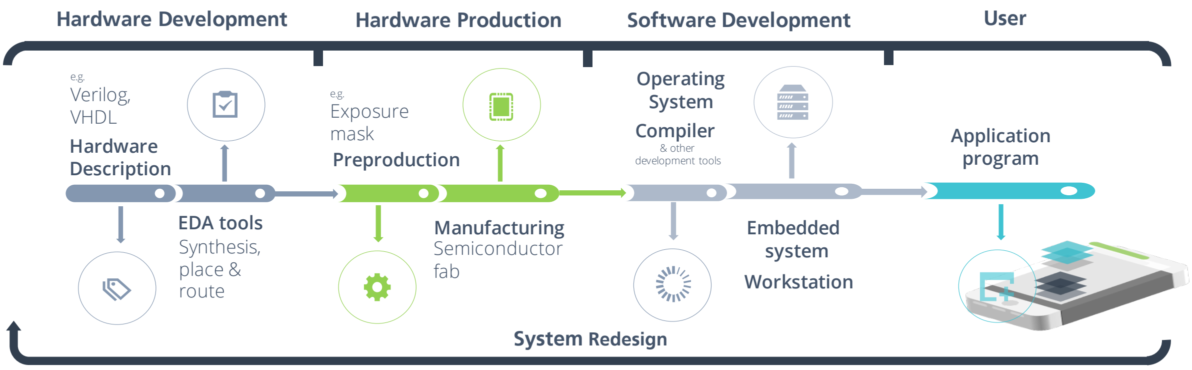


Fig 5: Information technology supply chain.

The figure shows a simplified version of the IT-supply chain. Note that, each step can be affected by a malicious attack which causes a complete loss of trustworthiness for the following stages. Therefore a secure development process has to cover the complete supply chain to be effective. Unless systems are redesigned from scratch (from silicon dioxide, with vacuum tubes or so), it is impossible to prove that an entire supply chain with all its inputs is secure. Trust can only emerge over the course of years that the components did not contain any hidden weaknesses.

Description of the figure: On a computer, an application runs on top of an operating system, which controls the activities of the IT system. The operating system contains a kernel for its key functions, and other interfaces, e.g. for managing the peripherals like hard disks, network access or user input. The kernel, the operating system and any applications are built using programming languages, the output of which is translated by a compiler into a machine language (software), such that the hardware can understand and run the application. The circuits inside the chips (hardware) are produced using hardware description languages. To simplify the hardware design process it is common to use "IP cores", which are reusable abstract hardware components given as pre-produced, and mostly unchangeable hardware descriptions. The complete hardware description is turned into a register transfer level design (RTL), which is turned into a netlist. Using this netlist, which is essentially a circuit diagram, all the parts are physically placed on the chip and routed, i.e. connected by wires as needed. The final result of this process is called "tapeout". Traditionally, a tape was produced to be used in a fab, hence the term. The latter parts of hardware production are performed using EDA tools (electronic design automation). In a fab, the ultimate processes are performed, starting from the "tapein", producing the information for the photolithographic machine. Ultimately, with etching and other steps, tiny transistors, wires and other electronic components are produced – in several layers – on the chip. Computers are used in all the steps.

Box 1: Information technology supply chain

lower levels will be attacked, e.g. the operating system and thereafter firmware and hardware. If such lower levels were secured, too, e.g. the hardware, then the tools used for securing and building them would logically become a target.

- Supply chain risks: Producers depend on those in the supply chains. If some company intends to use IT in its products, it must trust various components, many of which are black boxes: "Nobody tells them what is in their chips", as one interviewed expert put it. Furthermore, there is theft of chip designs. Fake and manipulated chips and the resulting risks are another issue (see Fig. 4; cf. Abesamis/Leblanc 2015). The interruption of the hardware supply chain by strikes, catastrophes, etc. is also taken into account.

- Safety issues resulting from insecurity (as in the cases of critical infrastructures or transport systems) are also addressed.
- Issues of income (wages and profits) due to the concentration of IT industries in a very few countries, mostly in the US and China, are an issue (though more are involved in the global procurement chains of components and tools, cf. Fig.8). The major designers of software, hardware and tools form a relatively closed circle, hindering cheap and easy innovation.

In **Section 3**, technical options for solutions are sketched. An important limitation of today is that software-only approaches cannot be properly designed as long as the hardware is a black box for the developer. Subsequently, open hardware is discussed as a component of a solution, including the need for open processors and open development and production tools to be used in the fabrication of semiconductors. Additionally, verification of any component would be useful, with verification meaning that components have either been checked to fulfil their purpose or, better, been formally verified, i.e. proven to be correct. In Box 1, a high-level overview of the supply chain is provided.

Moreover, it should be mentioned that formally verified and open components are emerging (see Fig. 7 of an implementation of RISC-V, an open source processor, and Fig. 6 for using a proven open source operating system kernel to secure a variety of programs). As proofs are expensive, the design of more open, unproven components can also make sense for providing cheap solutions which can be more trustworthy than closed ones. Open components would also have the advantage of being available for many years, even for decades, as is necessary in the automotive industry, with home automation and in other Internet-of-Things (IoT) markets. The graphics processing units producer Nvidia already plans to sell mass products using an open processor starting in 2018 (Sijstermans 2017), as does Western Digital (2017). Ultimately, control over the semiconductor fabrication plant, or "fab" for short, and the programs and computers used in it would be needed.

The White Paper argues that, regarding law enforcement, the superior approach would be to observe the suspects, as opposed to compromising entire classes of devices or components of the IT supply chain, which risks revealing business secrets and disrupting the reliable functioning of products.

Finally, the question of how to deal with closed systems, as used in the production of semiconductors and elsewhere, is reviewed. An option is certification by trustworthy institutions. More distributed, i.e. regionalized, competition would achieve more resilience. As many of today's closed systems are very fast and energy-efficient, this is certainly a problem only solvable in a somewhat distant future, when open tools might become available to design fast general-purpose computers. Split hardware can help, with trustworthy systems running in parallel and connected to traditional ones. Even displays showing separate windows have already been demonstrated; they offer a convenient parallel handling of newly built, secure systems, and legacy office and engineering systems.

In **Section 4**, the economic and political options for facilitating technical solutions are discussed. First, it is envisioned that a process similar to that of the development of

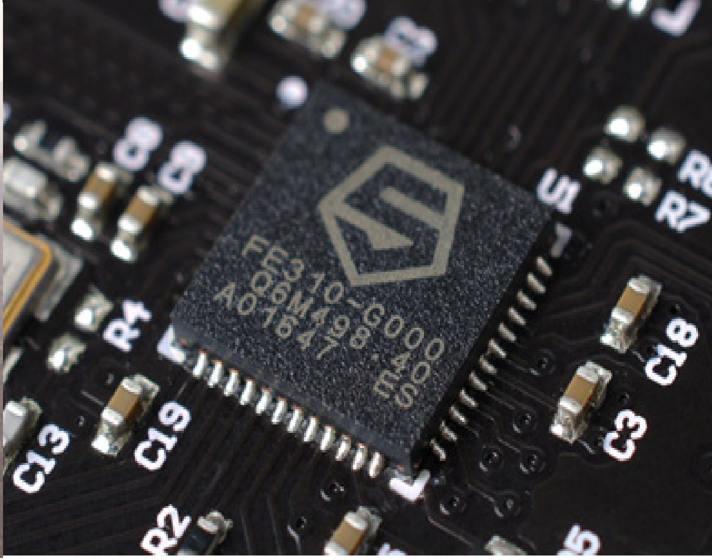


Fig. 6: Proven seL4 in unmanned helicopter (Data61 2017, cf. New Scientist 2015, Boeing 2015, Klein et al. 2018).
Fig. 7: SiFive RISC-V chips (CC, Halfacree 2017).

the Linux/Android operating systems could take place throughout the supply chain, even including the tools used in fabs, leading to the production of cheaper, transparent and more secure components. Such a “commons”, as economists call it, would be in the interest of many industrial customers as well as in that of private users, and it could be financed by enthusiasts, governments and industrial companies, much like Linux was. Revenues would be earned from services and the sale of new hardware, possibly even resulting in faster innovation. The new approach has the potential to be disruptive, much like Linux and Android were, which will spur the search for profitable solutions. This open approach could be kicked-off not only by venture capitalists, but also by governments and by user groups, as in the IoT or machine-building industries, and trigger the creation of one open component after the other.

With regard to closed components and systems, the purpose of some activities by investors is to have more control, such as the investments by Softbank in the processor designer ARM, by Siemens in the EDA-tool manufacturer Mentor Graphics (producing electronic design automation tools) or by Bosch in a fab of its own in Germany. As a representative of Bosch put it: “For us it is important that the key technologies are in our own hands and that we are not dependent on suppliers“ (Dirk Hoheisel, according to Reutlinger Nachrichten 2017).

Regarding the closed path, governments could design legislation to enforce the verifiability of critical components and the liability for flaws and backdoors, such as if they are known to nation states but left unpatched by manufacturers. As Bruce Schneier put it (2017): “Many new technologies have led to the formation of new government regulatory agencies. Trains did, cars did, airplanes did… We need government to ensure companies follow good security practices.” Or Michael Waidner et al. (2013): “Regulatory measures [such as] the obligation to be transparent vis-à-vis selected, credible, national test laboratories.”

Funds to support either are available, as the purchases of Globalfoundries and the US Trusted Foundry by Abu Dhabi have shown, or the purchase of ARM by Softbank. So major steps would need to be taken to produce open computers free of flaws and backdoors, such as a national plan or a plan by a family of industrialists. In this way, traditional PCs and servers could be replaced at least in sensitive areas, such as in fabs or in businesses relying on secrets – remember how Rockefeller funded the Macintosh or maybe think of the Manhattan project. See how China develops processors of their own: Triolo et al. (2017) report that China is “developing an independent, domestic technological base for the hardware and software… [forming] a solid cyberspace security shield”. Among other issues, China is looking into the “potential threats from EDA tools” (Qiu et al. 2016) and develops tools of their own (cf. Hu 2010, Li 2013, Empyrian 2017). Russia also develops processors locally (Tass 2014, Baikal Electronics 2017). Furthermore, India is looking into developing their own processors: “We don‘t know really whether the processor we are getting from outside is trustworthy. Is it secure?” asked V. Kamakoti of the Indian Institute of Technology, according to Sharma (2017) who mentions the Shakti-project and raises the question: “An ARM killer from IIT, Madras?” (cf. Merritt 2016). Even the U.S. government has discussed “to reintroduce production for the full vertical stack into the United States”, starting with 5G telecommunications equipment, according to a report based on leaked documents, which the Brookings Institution found “excellent” (Axios 2018, Brookings 2018, Washington Post 2018). The influence of large investors or governments could at least change the closed source systems, which would help if the open source community ran into a lack of thrust. This would be an alternative path if the open source path should not become successful in the near future. The authors do not find it appropriate to predict which regimes will prevail; change in either direction is possible within decades and within years. Much will depend on the development of discussions among investors, in the industry and among the public.

Section 5 contains a comprehensive list of action points, comprising issues such as gathering more knowledge about threats, the production of more open source components and the corresponding tooling, the need for research on chip verification, and the necessity to elaborate in greater detail the options for supportive economic and legal processes.

Section 6 provides an overview of the match of key problems and key solutions. One can imagine being at crossroads and being able right now to change directions so that more open and even formally verified components begin to emerge, which could be combined to form systems. This section discusses issues for investors. At the same time, governments could implement rules for liability, impose requirements for certification, require the use of open, verified or even formally verified components, and support a first set of prototypes. As the production of such components has already begun (see Fig. 7 and Fig. 9), a first, high-level plan is devised. It comprises measures such as creating awareness of the technical, economic and legislative options, creating global discussions, kicking-off the design of open and/or proven components, e.g. with public support for industry or the armed forces, creating interest with investors in a Linux-like supply chain, and working out legislative options concerning what to make mandatory first. As far as the authors know, this is the first plan to address threats anywhere in the whole supply chain, including advanced persistent threats (APTs) and novel types of Trojans. In conclusion, all the reasons for an optimistic outlook are compiled.

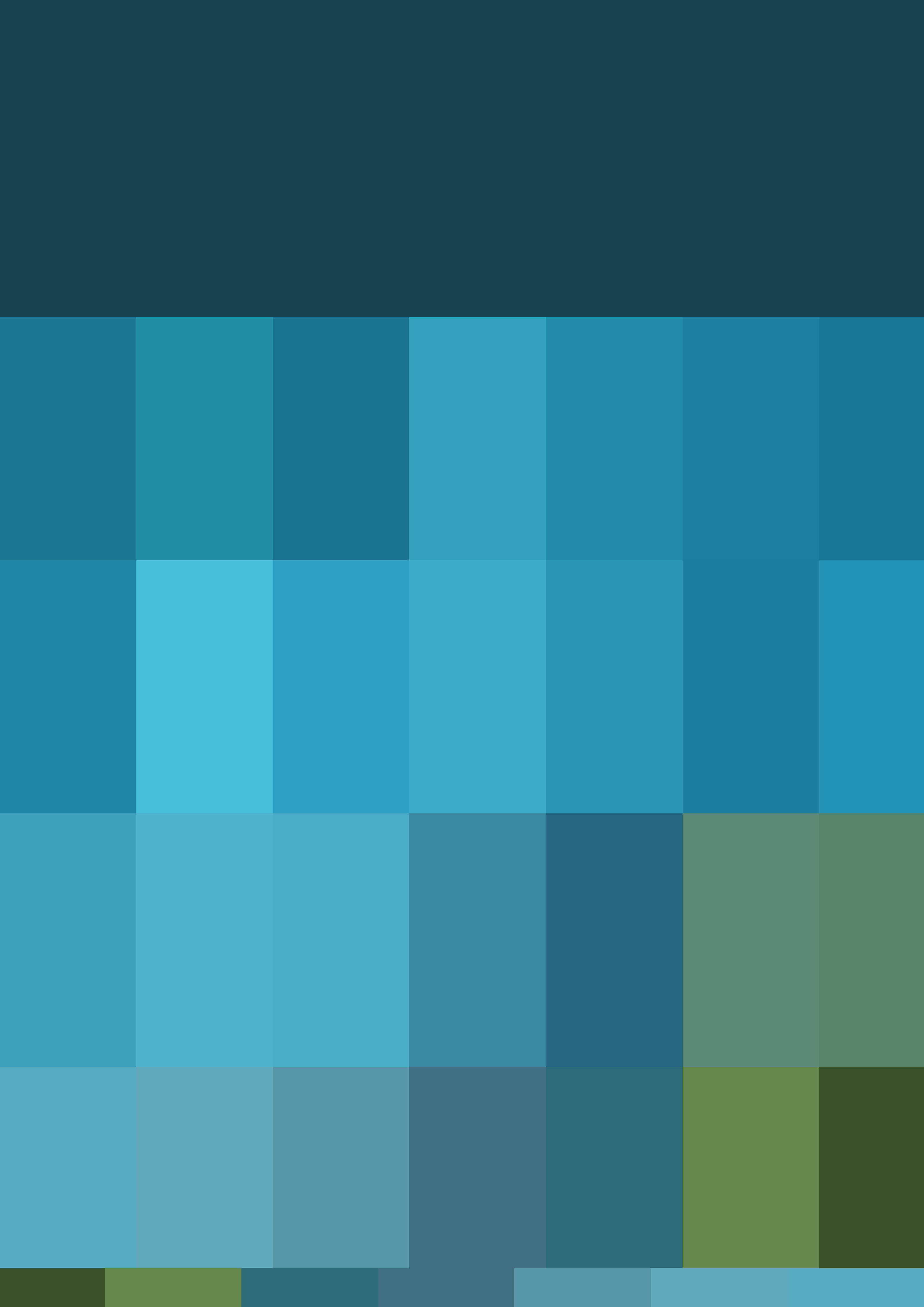
FULL VERSION OF THE WHITE PAPER

1 INTRODUCTION

This White Paper contains suggestions for improving the information technology supply chain with regard to both enhancing the security and safety of products and to making the supply chains fairer and more resilient. Problems that are taken into consideration range from the malicious floods of messages straining the Internet to eavesdropping on business secrets and to kill switches in infrastructures, no matter whether they originate from imperfect software or hardware.

This document is written from the perspective of those who do not live in a country which is part of the supply chains that are currently dominant. It is based on two efforts. Fraunhofer SIT, Karlsruhe Institute of Technology (KIT) and Télécom ParisTech held a workshop January 12, 2017, and its results are reflected in this paper. Furthermore and in cooperation with these partners, KIT interviewed 17 experts from industry and research in 2015 and 2016.

A broad field of technical issues is presented first, highlighting various problems from day-to-day attacks to the rarest stealthy ones which need to be anticipated. The options for developing solutions are subsequently sketched, from creating open source hardware that supports better software mechanisms, to investor actions and legislative policies. Action points of measures to stimulate research and to encourage readers to participate in solving the issues have been prepared as well.



2 PROBLEMS:

DIGITAL SOUVEREIGNTY, INFORMATION SECURITY, SAFETY AND SOCIAL PRODUCT

Highlighted in this section are selected problems related to security, safety and social product, the latter an issue related to the other categories but also an objective in itself.

SECURITY – LIVING IN LA LA LAND?

At first glance, the task of securing the entire information technology supply chain seems impossible. Every device or program in use may already have been compromised or be flawed. Confidentiality, integrity and availability (*CIA*) may consequently already be undermined. As less and less hardware is being designed and built inhouse, the trustworthiness of hardware has also become an important topic (cf. Karri, Koushanfar 2014), in addition to the trustworthiness of software. As to flaws, one could try to iron them out by means of verification. This would lead attackers to target tools used earlier in the supply chain, e.g. the software used for verification or some not yet verified tool somewhere in the long supply chain such as a computer used in designing a tool used in a semiconductor fabrication plant (fab). Even if the safety of components were proven, dishonest employees or burglars could modify them, e.g. integrated circuits (chips). In particular, one would most likely not be able to guarantee ex ante that a whole system is secure.

As it is not possible to achieve a perfect solution to the problem in one step, and as a complete fast redesign of IT would likely be deemed to be too expensive, a comprehensive and stepwise reduction in risk should be considered, with the sequence of steps depending on the values to be protected. Therefore the following issues will be dealt with:

What are the critical risks?
What could be done to mitigate them?

Some alarming examples of weaknesses, backdoors, Trojans or plans for attacks are:

- Insecure components of the Internet of Things (IoT): In 2016, two massive IoT botnets made it to the headlines. The IoT botnet Mirai – composed of compromised IoT devices such as insecure routers, surveillance cameras, digital video recorders and the like – powered a large distributed denial of service (DDoS) attack, spawning 620 Gbps traffic against the website of the security journalist

Brian Krebs. A second botnet consisting of a network of over 152,000 IoT devices, including compromised CCTV cameras and personal video recorders, was used to launch a record-breaking DDoS attack whose peak traffic exceeded 1 Tbps on the France-based hosting provider OVH (Arbor Networks 2016, Schneier 2016, Khandelwal 2016a, Khandelwal 2016b, Wired 2017). This did massive damage to the victims, but may also reduce the use of the Internet, as users may refrain from using it for important or time-critical activities. In addition, there have been reports about secret efforts to use DDoS attacks against the whole Internet: "Someone is learning how to take down the Internet" (Schneier 2016).

- In 2010 the Stuxnet virus caused substantial damage to an Iranian SCADA/PLC system where it reportedly ruined almost one fifth of Iran's nuclear centrifuges. Apparently, the US and Israel were involved (Kelley 2013). This made it clear that industrial systems are at risk.
- As early as 1999, two officers of the Chinese army, Liang and Wang, wrote about weapons such as "computer logic bombs, network viruses" (Liang, Wang 1999; cf. the former US presidential advisor Clarke, cf. Clarke/Knake 2010). Trojan horses have reportedly been inserted into products of companies such as Huawai (cf. New York Times 2015). Note that the first larger attacks on NATO information technology took place after the bombing of the Chinese embassy in Belgrade in 1999 (Healey, Jordan 2014). Remember also the attack on Estonia in 2007, taking Estonia offline, which was likely conducted by Russia (Goetz et al. 2009).
- The US is known for having placed backdoors in foreign computer systems. As Clarke wrote (2010): "The President should... be required to approve personally the emplacement of logic bombs in other nations' networks, as well as approve the creation of trapdoors on a class of politically sensitive targets. Because logic bombs are a demonstration of hostile intent, the President alone should be the one who decides that he or she wants to run the destabilizing risks associated with their placement" (Clarke, Knake 2010). The Snowden documents revealed that systems from HP, Dell and Cisco (e.g. routers in the Internet backbone) were modified in order to conduct attacks such as on the key management of smartcard provider Gemalto or on the phone calls managed by Belgacom. Such attacks were partially conducted by using the System Management Modes of processors and partially by modifying the firmware (cf. Appelbaum 2013). Note that "computer network exploitation" implants (CNE in NSA parlance) are said to exist in tens of thousands of servers (Snowden 2015a). Meanwhile, more weaknesses may have been implemented, by whatever nation state, and be used and abused by malicious insiders or knowledgeable criminals. The recent hardware bugs Meltdown and Spectre in Intel, AMD and ARM processors may be used to eavesdrop on any kind of data, in PCs as well as on servers (Lipp et al. 2018, Kocher et al. 2018).
- There is speculation that some chips used in a Syrian military radar system contained a backdoor to the effect that it was possible in 2007 to switch it off and on again (Adee 2008; Matt 2007; Mitra, Wong 2015).
- Some modern CPUs include extensive built-in hardware and firmware for

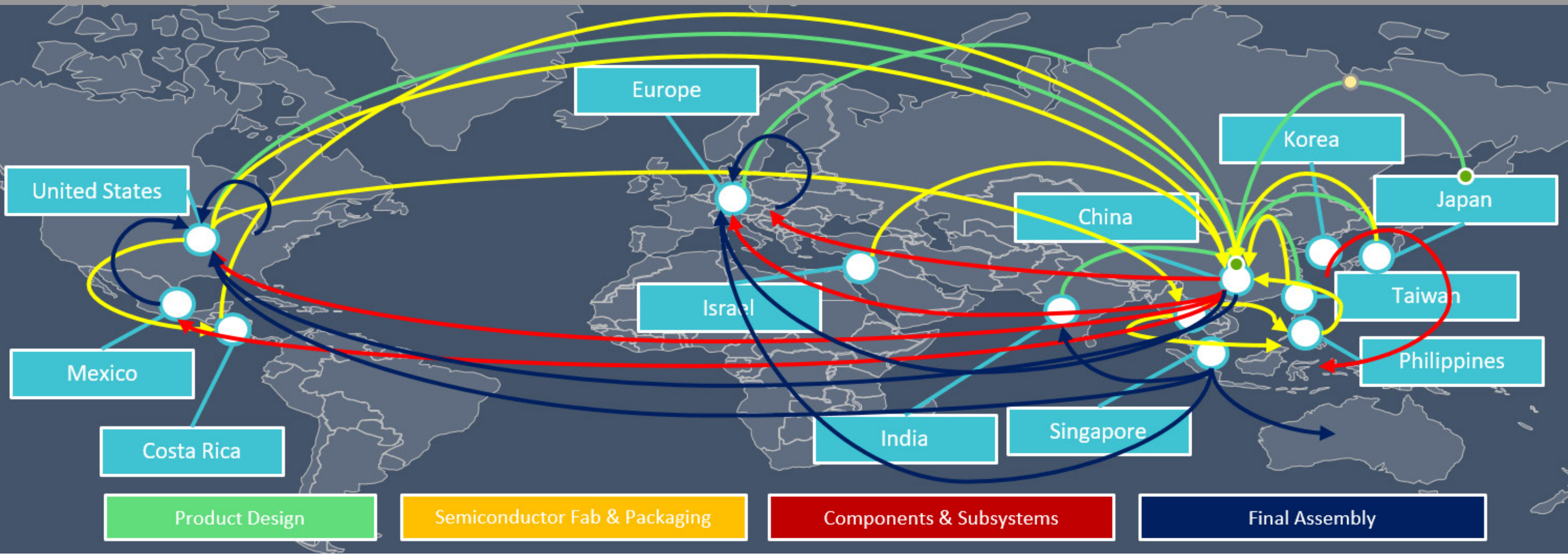


Fig. 8: Supply chain of a chip for a combat aircraft. Source: IDC Manufacturing Insights, according to Leef (2014).

remote out-of-band management like Intel‘s Active Management Technology (AMT; cf. Wikipedia 2017b). Because AMT allows access to the CPU below the operating system level, it constitutes a major entry point for remote attacks. AMT is dangerous not only because it has full access to the CPU hardware at a very low level, but also because its code is secret and proprietary, only available to some US citizens (Vandewege et al. 2014; Appelbaum 2013) and can therefore not be thoroughly reviewed by a third party. With considerable effort, it is sometimes possible to identify Intel's unpublished instructions. These hidden instructions can be abused for espionage or sabotage (cf. Cimpalu 2017, Domas 2017).

- Backdoors have also been placed in the firmware of hard disk drives (Techpowerup 2015, Appelbaum 2013).
- Russian hackers produced instructions for infecting PC UEFI firmware (Oleksiuk 2015). Affected PCs cannot even be protected by a perfectly secure operating system.
- State-organized industrial espionage is an important issue. It has been observed by industry that "data ... are ... ferried away to a remote location" (Dalton 2009). A long time ago, the case of the German company Enercon was in the media, which claimed that US competitors eavesdropped on secrets for building wind turbines (Schulzki-Haddouti 1998). French agents eavesdropped on Siemens' high-speed train prices so that French GEC Alsthom was able to undercut the prices of trains for South Korea (Spiegel 2004). As one of the experts interviewed for this White Paper concluded: "All intellectual property is available to whoever wants to take it" (cf. Washington Post 2017).
- Ransomware is using zero-day attacks, including known and unpatched vulnerabilities, e.g. WannaCry.
- The generation of electricity in Ukraine was interrupted during an armed conflict, possibly by Russia (ICS-CERT 2015), using "operation-specific malicious firmware updates", according to Zetter (2016).
- An FPGA chip for military use, sold by Actel, has been shown to contain a built-in Trojan horse (Skorobogatov/Woods 2012).
- Last but not least it must be added that hard-to-detect hardware Trojans (Lin et al. 2009) have been designed as proof-of-

> concept, e.g. by Becker et al. (2014) using small changes in the dopant level utilised during semiconductor production, and by Yang et al. (2016) using small capacitor effects of the circuits. Imagine that such a small change were made in the area of the random number generator; this would ease attacks on cryptography significantly. Sometimes, however, industry managers seem to think that Trojans in hardware are not a major problem. As Wally Rhines reports, the CEO of the EDA company Mentor said: "I've gone to meetings and heard that [embedded Trojans] don't seem to be a problem. If they are around, our customers who buy design tools don't seem particularly troubled about it. But then if I tell this to guys at the NSA, they roll their eyes like I came out of La La Land" (Techdesignforums 2017).

All of these vulnerabilities exist in the whole supply chain, which is global in character. Figure 8 shows how the components of a single chip travel through the world, according to an analysis conducted for the US armed forces (Leef 2014).

One more argument has to be put forward regarding the need for action. At the end of the first crypto debate, users managed to be allowed to use strong encryption, with the effect that nation states performed counter-attacks such as attacking the computers at the end of communications or compromising anonymisation nodes (Appelbaum 2013). A comprehensive solution should therefore take this into account. Now assuming one used secure software and operating systems one day, both criminals and nation states would most likely place backdoors in lower levels. Assuming the higher levels were secured by using some "cybersecurity fortification", as mentioned by DARPA (Networkworld 2017), cf. their "unhackable" drone or the "Merkelphone" (New Scientist 2015; Golem 2014), it will become attractive for attackers to target lower layers or the tools used for designing and manufacturing them. What may have been judged as being paranoid ahead of the Snowden revelations has now simply become logical.

Very detailed lists of vulnerabilities exist, e.g. Tehranipoor (2009), Karri/Koushanfar (2014) or Vishik et al. (2017). The above bullet points thus just show a small selection of things which have happened or which have been considered as possibly becoming real in the near future. At the same time the list does address, however, the fact that the whole supply chain needs to be regarded, including the influence of nation states. This is not sufficiently taken into consideration in the three documents mentioned above.

As to defence, the "Head of Information Superiority" of the European Defence Agency, Michael Sieber has stated: "Among EU member states, it's hilarious: they claim digital sovereignty but they rely mostly on Chinese hardware, on US American software, and they need a famous Russian to reveal the vulnerabilities" (Sieber, according to Guerreschi 2016). Taking into account that backdoors and Trojan horses created by powerful actors such as nation states are not only dangerous as such, but might also be abused by criminals, a non-compromised world of IT components and systems would be desirable.

SAFETY

As IT components are increasingly being used in critical infrastructures, such as power grids or health systems, in traffic systems, and with industrial robots, flaws or attacks can have negative effects on our safety as well

- *In 1941, the German Konrad Zuse built the world's first freely programmable computer.*
- *In 1964 his company was taken over by the Swiss company BBC. By 1967, 251 Zuse computers had been built. Subsequently, BBC did not want to invest sufficiently to compete with IBM.*
- *„Pas une entreprise de la taille que vous proposez": so said a Swiss banker to Heinz Waldburger, who had marketed, in 1980, the world's first 100 PCs with a graphical user interface and asked to have a larger factory funded. Comparable products by Apple were not marketed until 1983 (Lisa 1983, Macintosh 1984).*
- *„National Champions … (form) cartels", the EU Commission wrote, in 1991, about videotex (Minitel, Bildschirmtext, etc.). A Post Office-approved modem then cost, in Germany, DM 1,950, while a better, illegal one cost only a fraction of this amount. Partially, pricing of content was per page, i.e. the service was expensive as well.*
- *"You should not cannibalise your SMS revenues": a typical statement of a manager of a mobile operator, around the year 2002, when considering the use of Internet technologies on 2G or 3G networks. At the time, what later became the iPhone services, such as email, photos, music or videos, were already known from Japan.*

Box 2: Self-created decay of European IT. The box shows that European investors did not want to finance or market key innovations, such as the computer, the PC with a graphical user interface, and the fixed or mobile Internet (Spiegel 1966, Wikipedia 2017i, Furger 1993, Kommission 1991, Weber/Scuka 2016).

(cf. Ford 2013). Safety will also be reduced if counterfeit chips are used, e.g. mislabelled ones (cf. Miyamoto et al. 2017). Furthermore, chips can be fake and not provide a proper cryptographic implementation or safety feature; they would have a lower die size, making them cheaper to manufacture. Chips may also be counterfeited from old chips (cf. Hughitt 2010, Abesamis/Leblanc 2015). Consider a "kill switch" in an energy or tap water system and take the attack on Ukraine as an example. The US literature points out that it is "very plausible" that a "European chip maker" already has embedded "kill switches" (Xiao 2016), though without naming details. An interviewed expert even talked about the possibility of "hardware Trojans blocking each other". How many dormant hardware Trojans are there already?

It has also been shown that flaws can be used to attack cars. Numerous car hacks have been reported in the media and the academic press, those on the Jeep (Greenberg 2015) and on Tesla (Solon 2016) being among the most prominent. For both Jeep and Tesla vehicles, the hackers were able to gain complete control of the car including steering, brakes and transmission.

SOCIAL PRODUCT

Income – wages and profits – can be influenced negatively by a lack of sovereignty in IT. For instance, chip designs can be stolen. Our interviews revealed that millions of copies of proprietary European chips have been produced illegally in China. Furthermore, the supply chain might become interrupted, e.g. by a catastrophe. Strike waves are imaginable, as are trade wars or fully blown ones – as one expert put it: "Perhaps North Korea will nuke China some day." Or perhaps South Korea?
The concentration of IT in China and the US

also means that the range of available skills will be reduced and local, trustworthy specialists have become hard to find. In general, a more even distribution of the added value created with IT would be of great importance, including the benefits from having local know-how in terms of employment and sovereignty. More competition on every level, including the geographical aspect, would be good for having more resilience, e.g. in case of an armed conflict or a strike wave. This may appear to be contradictory to economic principles. Of course, it would not be cost efficient if every country ran its own semiconductor fabrication plants (fabs) with its own EDA (electronic design automation) tools or used their own operating systems. However, more competition is possible. One could for instance have some fabs with local staff in Europe, Southeast Asia or in Latin America. Furthermore, the current fairly closed circles of producers of software, hardware and design tools can be regarded as an obstacle to cheap and quick innovation, such as by smaller companies.

With regard to Europe, it is not long ago that Europe believed it was leading in mobile technologies. For Europe the loss of revenues to US companies is particularly negative. Had European investors behaved more competitively, the world might still have a significant third centre. Note that European experts did anticipate markets and foresee issues. Since other world regions might also wish to think of actively participating in global competition, the causes of the European disaster are briefly reviewed in Box 2, showing that US domination has not been the inevitable outcome of the last 30 years.

3 TECHNICAL OPTIONS FOR SOLUTIONS

In this section, a number of technical options for solving the security and safety issues are reviewed. First, it is shown that software-only approaches cannot be properly designed as long as the hardware is a black box for the developer. Subsequently, open hardware is discussed as an approach to a solution. Later, the issues of dealing with closed systems and with the tools in the supply chain are reviewed.

PROBLEMS OF SOFTWARE-ONLY APPROACHES

First, two misconceptions are addressed. One originates from practitioners who believe that software can control the hardware. This is not the case. One can imagine that there is some software on some chip which is designed to differentiate between legitimate and illegitimate input. This means that this piece of hardware checks a criterion. The hardware could, however, be designed in a way that if a particular piece of input is sent to it, it does something unwanted by the user, such as go into a different mode or allow access to unsigned information. This demonstrates that software cannot control the hardware.

Another standard reasoning is that the suggestion be made to the operator of a system, e.g. of a critical infrastructure, that it invests in security up to the point that the remaining risks do not justify additional investments. "We know we can't achieve 100 percent security, so we have to make costbenefit decisions based on risk" (Horne 2016). "(T)he challenges ... to secure their legacy ICS and SCADA systems... it can be difficult ... to justify significant security budgets" (Massacci et al. 2016). First, some of the threats discussed above, such as flaws in large systems and backdoors inserted by nation states, cannot be erased by an update. Prior to 2009, nobody found it profitable to protect against highly sophisticated attacks such as Stuxnet or CNEs. Second, it may be impossible to design an update to an imperfect system to ward off a large threat, especially one which has not been experienced before. A new approach with far better components would be needed. In particular such updates will not be developed if the necessary components are not available. The update also will not take place if the cost posed by an attack were borne by others, such as the remainder of the economy. Just to be clear: updating running software still makes sense, as does doing research to improve it; this is however simply not enough if infrastructures, lives, or whole economies are at risk.

There is a lack of transparency regarding what is in a chip, making it impossible to properly specify security aspects of software: “Nobody tells them what is in their chips”, as one expert put it in an interview about industrial customers. Karri and Koushanfar wrote: “Since the 1990s, there has been a steady trend away from inhouse integrated circuit (IC) design and fabrication toward outsourcing various aspects of design, fabrication, testing, and packaging of ICs. The emergence of such a globalized, horizontal semiconductor business model created hitherto unknown security and trust concerns in the ICs and the information systems (rooted in these ICs)” (Karri, Koushanfar 2014). No matter how much effort one puts into securing software, it will be useless if one cannot sufficiently trust the underlying hardware it is running on. Ultimately, products that are based on untrusted hardware cannot be guaranteed to behave as their users and stakeholders expect them to. They might, for example, spy on their environment, cease to work, or start misbehaving upon a certain secret trigger. The severity of such a compromise depends on how much harm the particular product is able to cause to its environment. It tends to be much larger for critical infrastructures (e.g., power and water supplies, but also Internet backbones), but may also be significant for massdeployed IoT devices, especially if many devices start misbehaving simultaneously, e.g., as part of a DDoS attack or as an act of mass surveillance.

A related issue is data remanence. “Data remanence is the residual representation of digital data that remains after attempts have been made to remove or erase the data” (Wikipedia 2017c). Although data remanence is typically associated with non-volatile storage like Flash, given the complexity of modern CPUs it is unclear for how long critical data remains, e.g., in the processor cache.

THE NEED FOR SECURING SYSTEMS HARDWARE-UP

While the severity of a potential attack on a particular system usually can be determined reliably enough, it is much harder to estimate the likelihood of an attack based on a hardware vulnerability. First of all, due to the great complexity of modern ICs with up to 10bn transistors, the presence of a vulnerability is very hard to determine. In addition to unintentional flaws introduced in the course of the design process, reports indicate that chip manufacturers, partly on behalf of nation states, have also deliberately introduced backdoors and hidden functionality. Still, the mere presence of a vulnerability does not imply a risk unless there is non-zero probability for an exploit. Unintentional exploits for deliberately introduced vulnerabilities will be rather unlikely, assuming that those behind the vulnerability attempt to hide them well. The main motivations for intentional attacks include industrial espionage and political interests (so-called national security interests). Attacks based on these motivations have been reported a number of times, and their frequency can be expected to increase with the growing risk of geopolitical instability.

Hence, based on its particular risk profile, products must be made sufficiently trustworthy and secure from the hardware up (Karri 2014). Unfortunately, making systems trustworthy hardware-up turns out to be very challenging in practice.

For one thing, the sheer complexity of modern chips greatly impedes a comprehensive understanding of the chips‘ behaviour in various modes and states. Secondly, from the user‘s point of view, commercially available chips are essentially black boxes whose internals are only known to the manufacturer. This combination causes numerous security-related concerns.

For instance, the ability of a customer, even an industrial one, to test chips is limited because of their complexity and because the technical details of micro-architecture features are kept secret by the semiconductor manufacturers. This directly translates into an increased risk of zero-day hardware bugs. Moreover, extensive debug and tracing support is a potential source of security vulnerabilities as many CPUs are known to have undocumented test modes and features. It is not unlikely that attackers can use these features as backdoors to obtain root access to devices in the field.

With the current complexity of modern integrated circuits it is economically impossible to correctly evaluate the risk posed by backdoors, side channel and fault attacks. Even with detailed knowledge of the Register Transfer Level (RTL) design, which models a circuit and full information about routing and placing of the components, and a complete set of photomasks, the detection of unintentional modifications is highly challenging. In practice all this information is, as already stated, hidden from a customer. Simple leakage models that mimic some intermediate value(s) of a cryptographic algorithm or its data-dependent timing behaviour hardly capture the leakage exhibited by micro-architectural features like pipelining, caching, branch prediction, etc. (cf. the Meltdown and Spectre attacks). Thus, the effectiveness of side channel and fault analysis evaluation becomes questionable.

Lesson learned from Meltdown and Spectre?

With the recent Meltdown and Spectre flaws, we have another crucial security vulnerability at hardware design level, which may have been around in some form since early 90s. Both attacks are results of hardware performance optimizations of modern pipelined microprocessors architectures to improve performance and utilization of computer resources. While Meltdown exploits a flaw in the Out-of-Order Execution (OoOE) found in many processors (Mitre 2017a), Spectre violates memory isolation boundaries by utilizing speculative execution and indirect branch prediction (Mitre 2017b, c). Unauthorized extraction of information is gathered via a micro-architectural side-channel attack where a sufficiently smart task controlled by an adversary can extract sensitive information of other tasks by observing its own performance. If successfully exploited, an attacker can obtain any data stored in memory. Meltdown affects almost every Intel processor since 1995. Spectre affects e.g. Intel, AMD, POWER and ARM processors, extending its reach to pretty much every

Fig. 9: HiFive Unleashed board with the 64bit RISC-V Freedom U540, which is a Linux-capable, multi-core processor up to 1.5 GHz. (Source: https://www.sifive.com/products/hifive-unleashed/)

device from classic PCs to the Internet of Things. Both vulnerabilities are operating system agnostic. Patching without slowing down every CPU is almost impossible without a redesign.

Intel already knew about these specific vulnerabilities several months before a team of researchers had contacted them. But more crucial, the industry has probably widely ignored this problem for decades, or at least has accepted such risks for a better performance of their chips. CPUs were developed by a closed members-only committee, making all the same mistakes as before. And, as these vulnerabilities have shown, this is still the way things are done. Maybe Meltdown and Spectre vulnerabilities are a source of inspiration for allowing new directions on secure and trustworthy ICs (e.g. incomplete evaluations reducing the likelihood of exploits, cf. Heiser, Yarom 2018). A widely accepted open nature of ICs might allow significant advantage in preventing hardware vulnerabilities and help provide alternatives to current closed-source chip design.

OPEN SOURCE HARDWARE

One major path to minimize risks consists of using open source hardware, borrowing from experience with internationally produced open-source software. "Today, 95% of Internet servers run on Linux open system. Approx. 85% of all the smartphones run the open source Android operating system" (TechTime 2016). Hence the open source hardware path should be seriously considered for the IoT, in the automotive sector, for industrial applications, and for components in critical infrastructures, such as for electricity or water, or in the health sector. Perhaps some of the disadvantageous decisions taken in Europe during the last 50 years (referred to in Box 2) would not have been taken if investors and engineers had had open source components in their hands to design computers and communications systems as found advantageous by them, at a certain point in time, e.g. for setting up the mobile Internet already in the 1990s.

Open hardware, sometimes referred to as FOSH (Free and Open-Source Hardware) "consists of physical artefacts of technology designed and offered by the open design movement... The ... hardware design... [including hardware description language source code and layout data is] released under free/libre terms. The original sharer gains feedback and potentially improvements on the design from the FOSH community" (Wikipedia 2017d).

A prominent example of open hardware is the RISC-V ISA (Reduced Instruction Set

Computing; Instruction Set Architecture) based on the established RISC principles. The RISC-V ISA is freely available under a Berkeley Software Distribution license which allows anyone to design, manufacture and sell RISC-V chips and software (RISC-V Foundation 2017; for the motivation of the creators, see Asanovic 2014). The first RISC-V chip has become available in the form of the Freedom E310 produced by SiFive, being the first industrial open source SoC (System-on-Chip; SiFive 2017, see Fig. 7; Huang (2017) claims that parts are still closed). The E310 chip is based on the "Rocket" CPU design of the RISC-V ISA. In 2018, SiFive made the Freedom U540 chip available, which is a Linux-capable, multi-core processor with a speed of up to 1.5 GHz, and the HiFive Unleashed development board (Fig. 9).

The RISC-V project originated in the Computer Science Division at the University of California, Berkeley, in 2010, but many contributors are volunteers and industry workers otherwise unaffiliated with the university. A partial list of organizations supporting the RISC-V foundation includes AMD, BAE, Google, Hewlett Packard Enterprise (HPE), Huawei, IBM, ICT, Lattice Semiconductor, Microsemi, Micron, Microsoft, Nvidia, NXP, Oracle, Qualcomm, Rambus Cryptography Research, and Western Digital (Wikipedia 2017e); a version running Linux is under preparation (Corbet 2018). Note that RISC-V does not mandate to make implementation designs public. In addition to RISC-V, there is also the security-oriented Cheri architecture which could be taken into consideration (http://www.cl.cam.ac.uk/research/security/ctsrd/), and an initiative for using open processors in automotive applications (Open Processor Foundation's J-cores, cf. Open Processor Foundation 2017).

Commercial chip vendors such as ARM Holdings and MIPS Technologies charge substantial license fees for the use of their ISA and the related patents. Many design advances are completely proprietary and never described, not even to customers. Moreover, the secrecy interferes with legitimate public educational use, security auditing, and the development of low-cost or free software compilers and operating systems.

By contrast, open hardware is transparent, free of royalties and licensing costs, and has several other advantages. First, it eliminates major risks during the sourcing process when the semiconductor suppliers are selected. Since all design artefacts are freely available, it becomes possible to switch from one semiconductor manufacturer to another at any time and no upfront arrangements regarding second or third source are needed. Second, the immediate consequence of freely available design artefacts is an unprecedented availability of the hardware itself, which has obvious advantages for smaller companies. At any time, anyone having the appropriate manufacturing capabilities can be contracted to jump in and manufacture the hardware. Third, open hardware implies a collaborative, community-driven approach to every design step from developing concepts over implementing and evaluating them to fixing the bugs. Community-based effort permits the latest technology from all the technical areas of hardware design to be integrated, e.g. switching parts off to save energy, which is something a single company can hardly achieve on its own.

In addition to the above advantages, open hardware is a promising approach for addressing the security-related issues identified in Section 2 resulting from the lack of trustworthy hardware. As a means for a solution, open hardware opens up the opportunity to tap the trustworthiness through transparency paradigm for hardware. Transparency, in

Fig 10: The polder model: cooperation despite differences (Sijstermans 2017)

turn, allows users to place justified trust in the hardware being used and enables comprehensive evolutionary improvements to be made.

As an example, a completely open specification of the hardware down to micro-architectural features makes it possible to build comprehensive test suites enabling exhaustive testing by all users (which would create business for specialists). Moreover, formal methods can be applied to further facilitate the trustworthiness of the hardware.

One of the major advantages of open source software is that the entire developer community around a FOSS project (free and open-source software project) – theoretically, anyone around the world – can review the source code. It has been repeatedly shown (Pfleeger et al. 2015) that code review is an effective means for finding bugs and security vulnerabilities. This needs to be done, as the latest vulnerabilities of SSL have demonstrated, e.g. the Heartbleed bug. Nation states may even try to insert subtle weaknesses, as the weakening of a random number generator has shown (cf. Schneier 2007). Analogous to FOSS, open hardware can greatly benefit from a community-driven effort to quickly find and fix any security-related bugs. At the same time it would be a kind of "breeding" for more experts.

Conjecturally, security-related hardware bugs can be found more easily than (due to the many-eyes principle) in closed hardware, and published analogously to vulnerabilities found in open-source software. While, in contrast to software, it is impossible to fix hardware bugs (unless one uses complex, expensive mechanisms), being aware of them typically allows programmers to modify the software to avoid a specific hardware feature. All in all, open hardware will significantly reduce the risk of unknown hardware bugs that can be exploited by malicious parties and will strengthen the know-how needed to fix these bugs.

Moreover, knowledge of the hardware down to micro-architecture features allows a side channel evaluation to be performed using much more precise leakage models than with proprietary hardware. It also reduces the total evaluation cost. Having access to the source code of the target hardware, e.g. its RTL description, allows a simulation-based assessment to be conducted as a first step of the evaluation. While this is not a replacement for final evaluation of the silicon, certain vulnerabilities from side channels and faults can be identified early in the hardware development process.

A potentially very significant example of using free and open source hardware for

Nvidia, a leading manufacturer of graphics processing units (GPUs), with revenue of US$ 6.91 billion in 2016, provides one of the first major examples of the commercial advantages that result when the ideas and techniques of open source hardware are utilized in practice. A modern GPU contains up to thousands of so-called streaming processors, which can be used for the fast rendering of graphics or for advanced parallel computing. So far, Nvidia has used a special control processor, the Falcon chip (Fast Logic Controller, cf. Koscielnicki 2016), for setting up and controlling these streaming processors. Due to problems of speed and scalability, the Falcon chip will be replaced in the next generations of their graphics cards. To achieve this, Nvidia has evaluated several commercially available CPU architectures. All of the suitable architectures did not meet their specifications for this application and even some ARM architectures do not match them because of their size (larger than 0.1 mm^2 on the die, cf. NVIDIA 2016) and the cost of the IP (cf. Xie 2016). RISC-V, however, can be implemented sufficiently small and is free. Moreover, Nvidia will have more flexibility with RISC-V because the company can modify and improve the RISC-V architecture without regard to any fees and patents. The "NV RISC-V" architecture will be used in production in 2018 (Sijstermans 2017). Frans Sijstermans of Nvidia describes the reasons for this migration to RISC-V:

> *"The flexibility in RISC-V where you can have a lot of different variations but keep it very clean… For example, if you want something very, very small, we can make a compact version... On the higher end, we develop quite big CPUs as well… It becomes at least an option with RISC-V, with the scalability it has. I want to keep control of the architecture… You don't have a real option to take an ARM core and turn it into something different that is better for me. You get what you get.*
>
> *On the quality side…, open source gets a lot of checks and balances… We get the best people in academia looking at the architecture, we get a lot of feedback from people in the industry. In the longer term, you will see that the quality of RISC-V surpasses the quality of the ISAs that we could have gotten from somebody else.*
>
> *We are happy to take everything we can take. Why would we also contribute? I am from the Netherlands. We have what is called a polder model. Polders are the reclaimed land from the sea. We've done that since the middle ages. At that time there was no real central government. People had to decide themselves: Hey, we're going to make a polder. Often there were fights between different villages... But they had to cooperate to keep the dikes and channels in place, otherwise the land would submerge…. And with RISC-V, you could say you are helping your competitors, why would you do that? Your competitor can just take. They don't pay the money you paid. That is actually a wrong thinking… Everybody will benefit if we all work together, that is a good thing. The other thing is we can influence the direction of RISC-V… The security work – I am not sure it would have existed if we had not said security is one of the things we really need to address early on." (Sijstermans 2017)*

These quotations show that a positive mood in communities can emerge which has the potential to improve quality and to iron out errors, creating a "commons" of free and good components. It shows the power to reduce the business of ARM and others. Obviously this reasoning will hold for other chip manufacturers and areas such as security solutions as well and therefore provides strong evidence that open-source hardware has the potential to change much of the current business models in chip manufacturing.

Box 3: Nvidia using the cheap and flexible RISC-V ISA.

products, the example of Nvidia, is shown in Box 3.

Western Digital has also announced plans to migrate its processors to RISC-V, and intend to ship two billion cores annually. Martin Fink of Western Digital described their motivation:

"The open source movement has demonstrated to the world that innovation is maximized with a large community working toward a common goal. For that reason, we are providing all of our RISC-V logic work to the community. We also encourage open collaboration among all industry participants, including our customers and partners, to help amplify and accelerate our efforts. Together we can drive data-focused innovation and ensure that RISC-V becomes the next Linux success story." (Western Digital 2017)

Another important aspect of using open hardware is that it will remain freely available for decades, forming a competitor to any closed system. This constitutes an aspect of great potential benefit in industries producing longlasting goods, such as in the automotive field.

Finally, the transparency of open hardware allows addressing the aforementioned issues related to data remanence, remote management modes, complex debugging functionality, hardware backdoors and hardware Trojans, fake chips and kill switches. The world would benefit from having an ever growing set of open, proven, and stable reusable components, not only, e.g., OS kernels, but also IP cores for the design of hardware and entire chips (reusable "intellectual property" designs). The approach embodied by open hardware would also increase the competence and availability of computer scientists and electrical engineers in all the countries participating in its implementation.

While auditing in early design stages, e.g. the analysis and validation of an RTL description, is straightforward and can be accomplished using existing EDA tools, such as simulators, this is not the case for manufacturing stages, in particular not for the final silicon. This will be addressed below.

TAMPER-RESISTANT HARDWARE

Security hardware such as smart cards, Trusted Platform Modules or Secure Elements could be used to provide a root of trust for securely booting devices and for loading only authentic applications and updates. This type of hardware is kept as simple as possible to minimize the risk of security vulnerabilities and costs caused by complexity. In theory, it would even be possible to pay the manufacturer for disclosing the RTL level design. However, the benefits of using such dedicated security hardware are limited:

- Software authenticated using security hardware is not necessarily free of vulnerabilities.
- No assurance for other hardware components is provided.

For cheap devices as in the IoT, the current costs of such modules often are prohibitive. For other markets, open tamper-resistant modules would be beneficial.

SECURING CLOSED SOURCE HARDWARE

The mainstream chips currently used in smartphones, PCs and servers are very complex. Yet a few approaches to make them more secure can be listed:

- Have them checked by trustworthy

institutions. As follows from the complexity, this is very difficult. Certification as such would not necessarily mean that they are free of design errors or have proven security characteristics. The issue of the standards for certification constitutes a topic of its own, as the discussion of Common Criteria standardization has shown. Certification could perhaps be enforced with regard to the components used in critical infrastructures.

- A variant of such a type of checks is planned in China. The government intends to conduct a review of foreign products: "The cybersecurity review shall be conducted for network products and services and their supply chains by the combination of businesses' commitments with social supervision, the combination of third parties' evaluation with the government's continuous regulation, and the combination of laboratory testing, on-site inspection, online monitoring and review of background information… The supply chain security risks associated with manufacturing, testing, delivery, and technical support of products or key components" are to be included (Covington 2017).
- Alternatively, access to confidential designs can be licensed, as in the case of China gaining access to AMD‘s processor designs (New York Times 2017).
- Have parts of them, e.g. intellectual property (IP) cores, designed by trusted suppliers, e.g. local ones, or have components verified. Love et al. (2017) describe how to prove the trustworthiness of such IP. Bhunia et al. (2017) describe certain countermeasures in great detail, but address only low-risk attacks, such as attacks on a private user or a DRM holder. Nowhere do they spell out how to arrive at a trustworthy supply chain, including the computers and software used in the design and production of hardware.
- Have chips tested for undocumented instructions. It appears this can be done for x86 hardware ("Sandsifter", cf. Domas 2017a, b). It could also be done for ARM-based hardware. This would help against attackers using existing backdoors, but actually it cramps systems. An attacker could place a hidden instruction in a more clandestine way, e.g. in a product update. A first instruction would put the processor into a different mode and would not be noticeable during any subsequent regular operation. Only if a second secret instruction were sent, would the Trojan become active. Such a procedure can be made arbitrarily complex and, for instance, after several rounds of instructions, lead to an effect such us the weakening of a random number generator.
- Against hidden instructions, a sentinel chip could be used to control the processor, allowing only intended or publicly specified instructions to pass. This may reduce the speed and could be used where applicable. Again, if a legitimate instruction were followed by a confidential one, the sentinel chip would not protect. A sequence of three commands could put the CPU into a special mode, such as:

  ```
  OP 8000932823808209382,9823098320
  OP 982393289328023832,232903802808
  OP 2312923808320832,23890230980943
  ```

- Buying the designer or manufacturer offers some control, as Softbank did with ARM (The Register 2017).
- China develops processors of their own: Triolo et al. (2017) report that China is

"SYSTEMS SHOULD BE MADE TRUSTWORTHY
AND SECURE HARDWARE UP."

RAMESH KARRI, FARINAZ KOUSHANFAR
UNIVERSITY OF CALIFORNIA AT SAN DIEGO/RICE UNIVERSITY

"developing an independent, domestic technological base for the hardware and software… [forming] a solid cyberspace security shield". Among other issues, China is looking into the "potential threats from EDA tools" (Qiu et al. 2016) and develops tools of their own (cf. Hu 2010, Li 2013, Empyrian 2017). Also Russia develops processors locally (Tass 2014, Baikal Electronics 2017, EETimes 2015). Furthermore, India is looking into developing their own processors: "We don‘t know really whether the processor we are getting from outside is trustworthy. Is it secure?" asked V. Kamakoti of the Indian Institute of Technology, according to Sharma (2017) who raises the question: "An ARM killer from IIT, Madras?" (cf. Merritt 2016).

- Even the U.S. government has discussed "to reintroduce production for the full vertical stack into the United States", starting with 5G telecommunications equipment, according to a report based on leaked documents, which the Brookings Institution found "excellent" (Axios 2018, Brookings 2018, Washington Post 2018).
- Last but not least one could aim a sufficiently powerful set of open chips, to be used in laptops or servers. Early, incomplete attempts can be seen in the Novena laptop (Huang 2018). The recent Boom implementation of RISC-V appears to be fast (Celio et al. 2017) and could be a starting point. If sufficient public or industrial support will emerge, the success of Linux and Android could be repeated. Subsequently, Microsoft could even make a port of Windows.

For some further approaches, such as using dual hardware or enforcing security by employing legal instruments, see below.

OPEN DEVELOPMENT TOOLS

The tools used in the production of hardware might be compromised, for example a hardware description language (HDL) such as Verilog and VHDL, used to produce the RTL design and the netlist of components and nodes. Other EDA tools are used to enable and control the processes in the fab and might be compromised as well. As a reader of the German website Heise Online put it, such tools could be a "Stuxnet with GUI".

As of writing in 2017, capable EDA tools are not available to open source developers because of their prohibitively high license fees (e.g. € 80,000 per month) or they are bound to vendor-specific hardware (FPGA). A few are however emerging, such as yosys, Arachnepnr and ice¬storm, forming a complete tool chain for Lattice FPGAs. Also, QFlow can already be used to create ASICs.

This problem leads to the non-availability of EDA knowhow, which poses problems to the industry. Taking into account that the kernel itself is not the only the remarkable outcome of Linux, but economically very important nowadays is that UNIX knowhow has become widespread. Similarly, more skills in using open EDA tools would be beneficial, too.

Attackers might address these levels if they cannot attack the higher levels. For instance, they might insert a stealthy Trojan, like the capacitive or dopant level ones mentioned earlier. Countermeasures would be:

- Design and use existing open design tools, e.g. yosys for the RTL synthesis of Lattice FPGAs (Wolf 2017a).
- Control EDA tools such as those produced by Mentor Graphics, which has been purchased by Siemens (Heise Online 2017).
- Creation of new EDA tools on all levels using open source techniques and licenses (cf. Tim Edward's Open Circuit Design Initiative, Edwards 2017).

Still, this approach does not necessarily work against an insider in the fab. The computers used in the process might also have been compromised.

Also, emerging open EDA tools may not be as efficient as the tools available to current chip manufacturers for optimising layouts. Yet, this could change over the years. Who would have thought 20 years ago that the majority of smartphones runs on an open operating system? Again, working with smaller chips such as ASICs would be the way to start.

CODE VERIFICATION

Verification of the code used in software, hardware and the tools could be very useful. Note that the US armed forces use proven components, making equipment such as helicopters more unhackable (cf. Fig. 6); furthermore, there are attempts to apply the approach to the Internet of Things (Secelements 2018). These developments are based on use of the seL4 operating system kernel, which protects against crashes, does not perform unsafe operations, can be used to isolate data, and enforces integrity. Its design, including the proofs, cost less than $ 4m (Heiser 2016, cf. Klein et al. 2018). A related microkernel has been deployed in billions of mobile devices, including in the secure enclaves of iPhones (Heiser 2013). Systems which are not proven may be broken, as the fiasco with the Fiasco microkernel demonstrated (Peter et al. 2014). Similar steps would be beneficial for the software used in the hardware design, e.g. consisting of open, preferably verified or even proven components. Languages should be used which reduce the number of vulnerabilities in the compiled code, e.g. Rust instead of C++. Note the need to consider whether the machine which produces the proofs is working correctly. A backdoor might work in both the machines the code has been programmed on and on the machine which verifies the code, so Thompson's problem of invisible backdoors can only be overcome if trust emerges over time into all the components of all the systems of the supply chain (Thompson 1984).

As proofs are expensive – though not as expensive as traditional means to secure code, Heiser would argue (2016) – for many applications it will also make sense to use open but unproven components, which may be programmed better in an open source environment, thus leading to more security. Furthermore, the more people are used to creating proofs, the cheaper these might become.

Regarding RISC-V, work has already begun on formal verification and there are ambitions to make the proofs amenable to modular refinement (Arvind 2016, Wolf 2017b). Other work aims at making it easier to prove code (Chlipala 2017, cf. https://deepspec.org/main). Again, these are big tasks, but who would have thought of the power of open systems 20 years ago?

IC RESILIENCE

It has been shown to be theoretically possible to design chips in a way that prevents an adversary from being able to insert a Trojan horse even if he can modify a certain fraction of the gates and wires. This procedure is based on proofs that can be checked probabilistically (Seifert, Bayer 2015). It could help even against the stealthiest modifications of chips as identified by Becker et al. (2014) and Yang et al. (2016).

VERIFICATION OF CHIPS

The existing methods for verifying hardware are based on deprocessing the chip by using techniques such as chemicalmechanical polishing and wet chemical etching to remove the package of the chip, thereby uncovering the silicon and the metal layers. Each interconnect layer and the active layer are then imaged at high resolution with a scanning electron microscope, making it possible to distinguish every physical feature. Finally, image processing algorithms are applied to these high-resolution images to determine the actual functionality implemented in the silicon (Thomas 2017, Starbug/Nohl 2008, Phillips 2008, Grand 2011, Thomas 2015). For what enthusiasts can do to reverse a chip design, see http://visual6502.org/welcome.html.

The main disadvantage of the existing methods is that they are inherently invasive. As a result, only random samples of a batch can be audited. Because the chips can no longer be used after being audited and the technical effort is costly, in practice only a small number of chips are audited. If the production batch is reasonably large, the probability of a malicious chip slipping through the auditing process would be rather high. In addition, randomly sampling the chips rather than auditing of the entire batch raises the question of how these samples are selected and how the selection processes itself can be protected against manipulation. Even if only a small percentage of the chips were manipulated, this would allow for attacks with far-reaching consequences.

Other approaches discussed in the literature to verify the proper working of chips are to measure electricity consumption (Becker et al. 2010) or to use physically unclonable functions (PUF; cf. Francq 2013).

CONTROLLING FABS

A fab can manipulate the production, for example by modifying the netlist, e.g. after "tapein". Such insider risks can be reduced if the production is carried out under controlled conditions, e.g. locally. This makes it possible, if required, for work to be conducted by nationals who are subject to national legislation. They can be held responsible, even if a local (national) entity requests them to do something illegal. Such an approach does not protect against bribery or extortion (cf. the statement of Peter Laackmann of Infineon, Zeit Online 2015), but could be a much safer framework than relying on employees in a foreign country and the trustworthiness of their local intelligence services. The US IBM Trusted Foundry (McCormack 2016) operated in such a setting. As a representative of Bosch put it, when justifying investment in a new fab of their own: "For us it is important that the key technologies are in our own hands and that we are not dependent

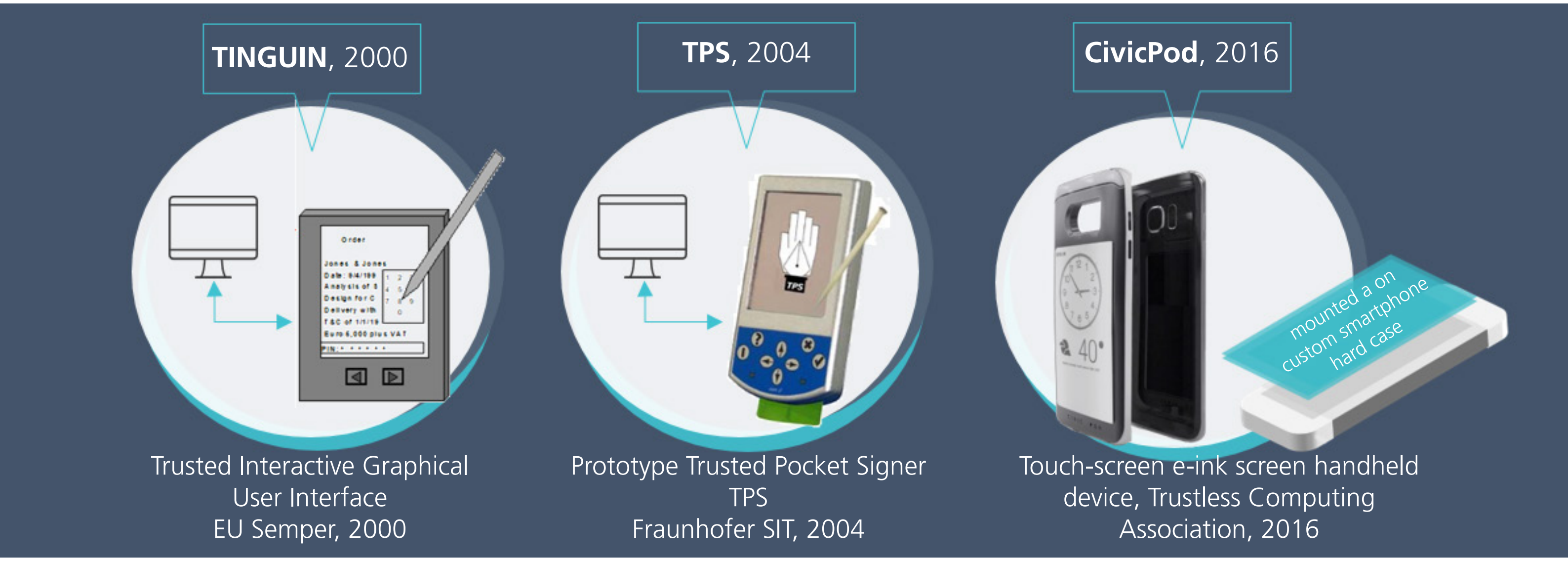

TINGUIN, 2000
TPS, 2004
CivicPod, 2016
mounted a on custom smartphone hard case
Trusted Interactive Graphical User Interface EU Semper, 2000
Prototype Trusted Pocket Signer TPS Fraunhofer SIT, 2004
Touch-screen e-ink screen handheld device, Trustless Computing Association, 2016

Fig. 11a: Proposal for a trusted interactive graphical user interface (TINGUIN) developed in the EU project SEMPER (cf. Lacoste et al. 2000, Weber 2000).

Fig. 11b: Prototype of a trusted interactive user interface (TPS) developed by Fraunhofer SIT and SRC (2004).

Fig. 11c: Proposal for a trustworthy display to be clipped onto a smartphone developed by Trustless.AI, Guerreschi (2016) (https://www.openmediacluster.com/trustless-computing-consortium/)

on suppliers“ (Dirk Hoheisel, according to Reutlinger Nachrichten 2017). Again, the software used in a fab, e.g. for mask creation, could be either made open, or controlled by trustworthy institutions.

From a security point of view, the statement “real men have fabs”, attributed to AMD’s Jerry Sanders, is not outdated, but economies of scale need to be taken into account. An approach based on local production cannot be applied to 200 countries for reason of cost-efficiency. However, it is imaginable that components could be sourced from a small number of countries, e.g. from different continents, so that this risk would be curtailed. For critical components, governments could make procurement from diverse countries mandatory so that not all devices would be affected if the facilities in one of them were not trustworthy. One could even compare the chips produced by different fabs. While the logic of pursuing economies of scale may lead to monopolies, having at least three world centres of competence on every field for reasons of security and safety would provide some resilience and would also lead to competition (and might require measures against collusion). Local production and the local availability of skills would contribute to local revenue. Of course, since it would be important for there to be a level playing field, intergovernmental agreements should, for instance, make sure that standards for worker safety and for the environment are similar in all such global centres.

Guerreschi has proposed having trustworthy foundries where users have access and can witness the processes. Technicians and such witnesses would publicly and completely document the process. This may not help against bribed or blackmailed insiders, but it is noted as a related proposal (Guerreschi 2017).

Trustworthy fabs would also help prevent manipulations in the fab and in the supply chain, such as modifying chips, as discussed above (Hughitt 2010, Abesamis/Leblanc 2015).

SPLIT PRODUCTION OF HARDWARE

“Split manufacturing” refers to conducting expensive steps of chip production

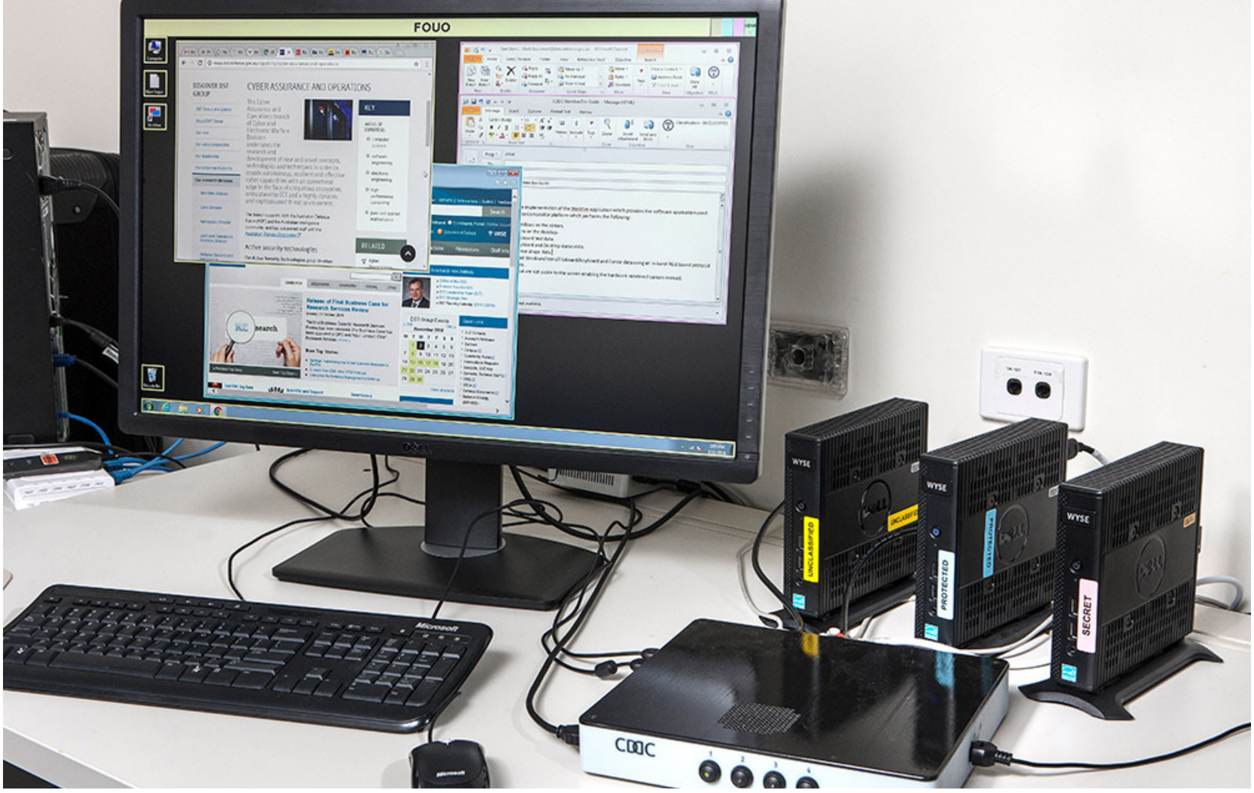


Fig. 12: Cross Domain Desktop Compositor. In this example designed for use in the armed forces, unclassified, protected and secret information is kept separate while managed from the same user interface. This approach could be used to handle legacy systems while migrating towards a new, more secure one (https://research.csiro.au/tsblog/cddc-wins-iawards/).

in a fab with low wages and conducting the steps which are more sensitive locally, in a more controlled environment. This may increase costs (Mitra et al. 2015, Xiao et al. 2016).

USE BOTH TRUSTWORTHY AND NON-TRUSTWORTHY HARDWARE

Computers which have been produced independently of the mainstream supply chains could be used to control transactions on the mainstream devices or to handle sensitive information, such as business secrets or banking. An open or verified device with its own means of user input and output could handle information in a secure way, such as by making a trustworthy visual check possible or by keeping key data isolated from the mainstream device. A basic design is to produce a smartcard reader with its own display, for instance for use in banking or for securing digital signatures, or a wallet with an observer that uses a smartcard which is trusted by the bank and an observing computer checking what is being communicated on behalf of the consumer (according to Chaum 1992). Another possibility for managing private wealth, for instance, is to produce a small trustworthy device to be used in addition to a mainstream device (cf. Fig. 11a, Fig. 11b), or a thin device to be attached to a smartphone, either using a shared screen or one of its own (Trustless 2017, cf. Fig. 11c). Should flexible phones become mainstream, a trustworthy device could also be flexible, or be worn like a wrist-watch.

A related, but different approach is traditionally used in the armed forces by using two machines, which may be of the same design, but could also be different, for example one with a mainstream operating system and one with Linux. This kind of “Adidas” computing is of course inconvenient. It would be much easier to use the same keyboard, video display and mouse for managing several physical computers. Australian Data61 developed a switch handling this securely, the Cross Domain Desktop Compositor (Trustworthy Systems 2017, Fig. 12). It can be used to run new, more secure hard- and software parallel to legacy systems, e.g. in industry, finance and infrastructure operations, where one may wish to continue using large legacy software packages, e.g. some CAD application, Excel, or SAP. Several hardware systems could be integrated in one physical casing, and diodes could indicate which one is active. As an alternative to diodes, one could use unsealed images to mark an active, trustworthy window. A user-selected image would be unsealed (decrypted) only if the window is in a trustworthy state, e.g. if booted correctly. Otherwise it would be very easy for the malware in the legacy system to visually mimic the trustworthy system. Last but not least, one could design secure channels for transmitting data, such as plain vanilla text or calendar information in both directions, or even executables, e.g. from a secure environment to an insecure one.

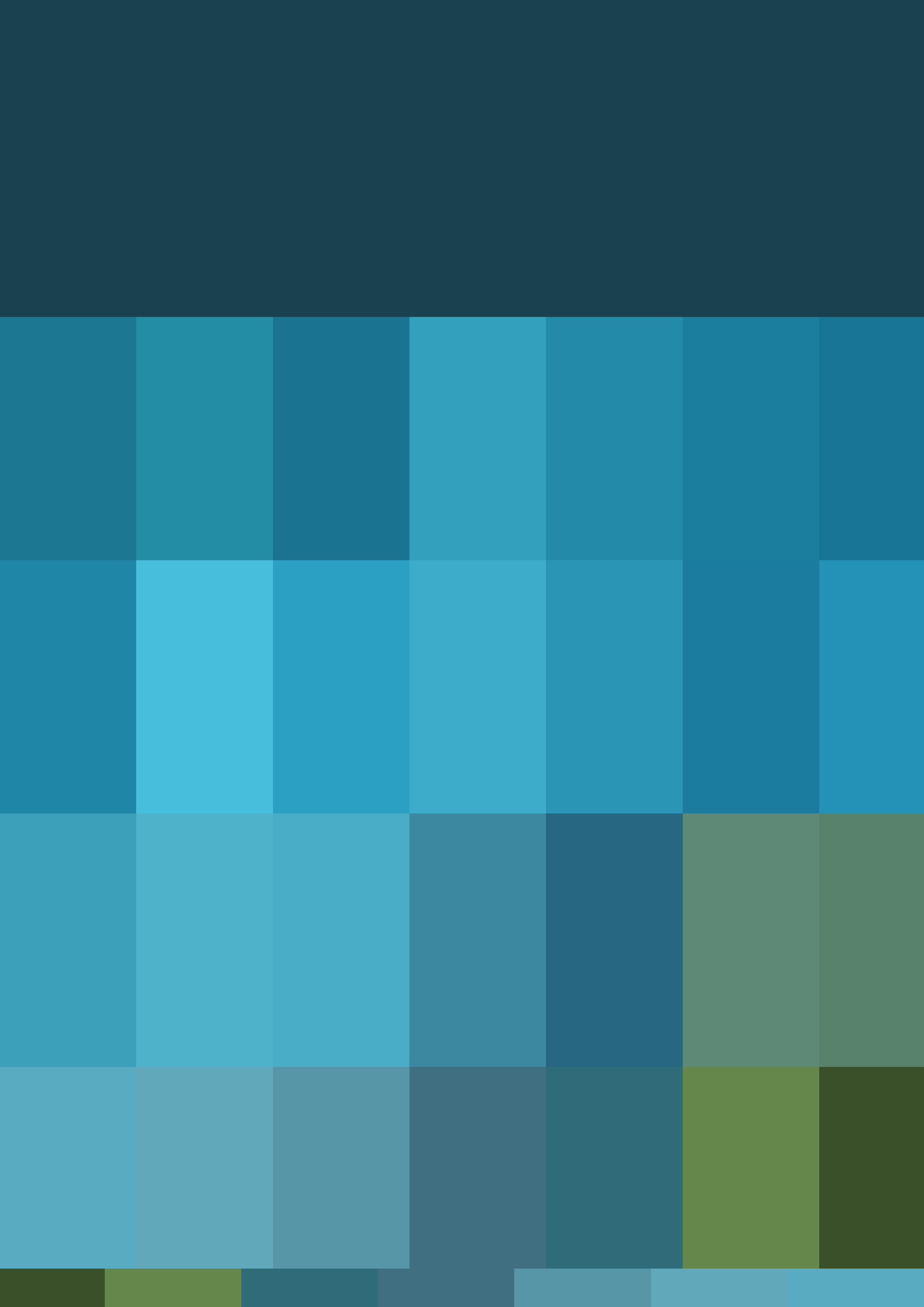
In yet another approach, once one had several, independent sources for a device, one could use several of them and compare their input and output through yet another, but trustworthy device. This could be a viable approach for routers or firewalls (Achenbach et al. 2014, Feldmann et al. 2016).

SECURING SOFTWARE

In any case, it will be necessary to secure the software on top of any old or new type of hardware. This can be addressed as discussed in the literature. One approach is to use virtualisation to separate trustworthy from non-trustworthy code. The virtual machine monitor used to manage this would make sure that malware cannot spread from one compartment to another. This would allow using legacy code, isolating new code and even isolating malware, e.g. when surfing the WWW. Sealed (encrypted) images could be used to show to the user whether or not trustworthy code is running, as unsealed images would only be displayed with trustworthy applications. The virtual machine monitor could be open or even proven, in principle like in the above mentioned unmanned helicopter (cf. Heiser, who proposed this for use in government servers, already in 2013). Of course, this would only work if the underlying hardware has no flaws such as Meltdown or Spectre.

LAW ENFORCEMENT

Analogous to law enforcement, the superior approach would be to observe the suspects, as opposed to compromising entire classes of devices, which would put business secrets and infrastructures at risk, as has been argued by Pfitzmann (2008) and Snowden (2015b). Compromising devices run by suspects could be similar to what is performed in Figure 1. However, it can also be left to political processes to determine whether to make it mandatory to compromise an entire class of end-user devices in preparation for employing state Trojans. It could also be argued that having a backdoor, e.g. in smartphones, would be preferable to having an unknown number of vulnerabilities somewhere in the supply chain of every other chip.



4 ECONOMIC AND POLITICAL OPTIONS

In this section, a number of economical and political options for solving the security, safety and economic issues are reviewed.

ECONOMIC OPTIONS

This dimension of finding a solution is on a different level. The first economic option is that investors conclude that the various risks, for example of eavesdropping on business secrets or of infrastructure outages, justify action. They could also judge that regional diversity of sources would be good to prevent a shortage of chips. The second option is that investors conclude that open approaches are not only more secure, but also cheaper than closed ones. Note the recent case of Google supporting the replacement of the proprietary and heavily criticized Intel Management Engine with open firmware. Google employee Ron Minnich developed the open replacement NERF (Non-Extensible Reduced Firmware), with the help of Cisco, Horizon Computing and Two Sigma (Minnich 2017).

Thirdly, innovative companies might have ideas for new products that utilize IT, which would justify the local or in-house design of soft- and hardware. Openness could greatly ease fast innovation in industries. As to the cost of proofs, one could think of governments supporting the development of a set of components which hardly change, e.g. operating system kernels and random number generators, or making their use mandatory for certain applications (see below).

INDUSTRIAL POLICY

It appears that the governments of some countries have developed industrial policies to support their national interests. A technologically unrelated, but successful example is Japan in the nineteenth century, when the Meiji government tried to work against Japan becoming a colony of European or "Western" countries, by improving their competences (Tsunoda et al. 1964). More recently, South Korea had a policy to develop IT, an important part of which was to build, initially, non-standard CDMA phones. China can be named as well and has an ongoing project to spend some $ 200bn in semiconductor technology (McCormack 2016). That may sound huge, but it is only equivalent to Apple's profits in two recent years – a market Europeans did not want to develop, as analysed above. Since China's customers all over the world will have to pay for the $ 200bn by buying Chinese products, this investment is supposed to pay for itself in a relatively short time. It is worth noting that the Emirates of Abu Dhabi bought IBM's US Trusted Foundry (McCormack 2016). Reportedly, they also bought Globalfoundries and a share of ARM (The Register 2017, Wikipedia 2017f). Such investments also appear to represent an important strategy, one that should be taken into account. In these cases, both related industrial policies and considerable funds are present.

On a different level, countries, or the European Union, could publicly declare an open source strategy, expanding on the success of open source software by covering the entire supply chain, fund related research projects, or set up related projects to create more competition with closed source companies (cf., for a nucleus, European Commission 2017).

LEGAL MEANS

Last but not least, legal means can be used to create demand for more secure or locally produced components. A voluntary approach would be to put related clauses into private procurement contracts. The buyer may, however, find that the seller is just a distributor without the means to control the supply chain. Public institutions could create more demand for more secure solutions via their procurements, e.g. for the armed forces or for other government purposes. Already in 2013, Heiser had proposed the "mandatory use of secure virtualization technology in egovernment servers within… 5 years… Only … accessible from terminals with certified secure communications components", referring to seL4 (Heiser 2013).

Finally, legislation could be put in place demanding more secure or local solutions, e.g. for critical infrastructures (cf. Schneier 2017):

"Many new technologies have led to the formation of new government regulatory agencies. Trains did, cars did, airplanes did… We need government to ensure companies follow good security practices: testing, patching, secure defaults - and we need to be able to hold companies liable when they fail to do these things… We need to enforce transparency in design."

Waidner et al. argue similarly: Solutions should be verifiable in labs, at least if their function is critical, and minimum standards should be enforced by governments: "Regulatory measures [such as] the obligation to be transparent vis-à-vis selected, credible, national test laboratories." Furthermore, insecurity in design should lead to liability (Waidner et al. 2013).

Legislation might also make it mandatory to buy local components, e.g. up to a certain share. Making it mandatory for IT users to purchase insurance could also lead to more secure products.

CONCLUDING REMARK

Overall, proposals such as using open source components, producing mandatory security standards or establishing liability rules have the potential to shake the markets. Existing companies based on using closed components may lobby against the new approach and at the same time secure their systems. Start-ups might sell products with unprecedented levels of security. The disruptive nature of such a transition will, of course, only be known after the disappearance of an old technology.

5 ACTION POINTS

In this section, thirteen research issues and other action items for industry, governments and the public are listed. It would certainly be good if the number decreased in the future.

ACTION POINT #1

LEARN MORE ABOUT THE THREATS IN ORDER TO FINE-TUNE ACTIONS

Some threats discussed in Section 2 have resulted in such large damages that action is justified. Other threats are still new and attacks have occurred rarely or primarily in regions of war. However, international conflicts may worsen, and more countries might become victim of rare or new types of attacks. Also, new types of attacks have been thought of, as the cases of dopant-level or capacitive Trojans have shown (Becker et al. 2014, Yang et al. 2016).

Furthermore, the better the upper layers of the supply chain are secured, the more likely it is that lower layers will be attacked. Remember that encryption with sufficiently long keys, which was very useful for e-commerce and other applications, led to compromises on the end points.

To be sure, measures such as legislation or the establishment of more local fabs come at a cost. How much investment is justified in which path? To be better able to answer this question, more information about the threats should be gathered from specialists, insiders and investors, including those whose assets are in danger because of lost business secrets or the interruption of production. If possible, such information should be published to gain the support of market players and to create demand.

ACTION POINT #2

MAKE MORE COMPONENTS TRANSPARENT OR OPEN SOURCE

For an open source hardware approach, the necessary components in a first step would include good tools for programming FPGAs, to be followed in a second step by ASICs (application specific integrated circuits). As of 2017 on github, the software project version control repository, there are 593 CPU designs in Verilog and 328 in VHDL. Moreover, there are 2,234 projects in Verilog and 2,272 in VHDL. This shows that there are already open source hardware efforts, but it is very likely that only a minority of them are of practical relevance. One of the authors of this paper designed a very small, but still rather fast processor using an open-source HDL synthesis tool to demonstrate the possibilities of this approach (Reith 2017, see https://github.com/SteffenReith/J1Sc under doc/misc). One vendor suggested using RISC-V in an FPGA in order to increase the stability and certifiability (Marena, Gerstl 2017). For the IoT, the Pulp Platform (http://www.pulp-platform.org/) is an early example

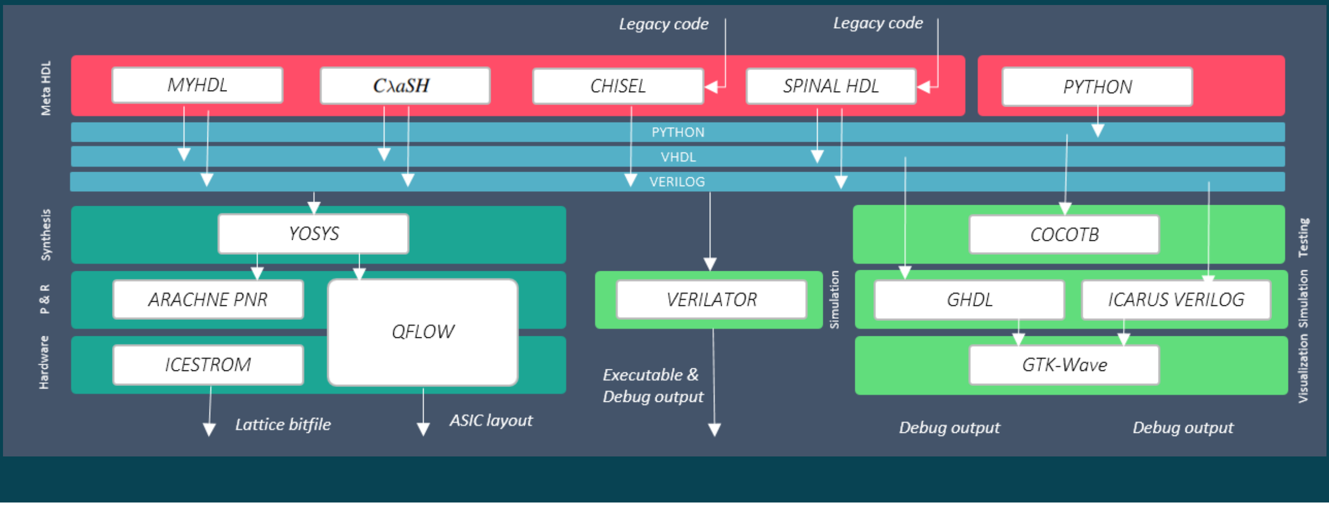


Fig. 13: An overview of some current open source EDA tools. The tools need to be improved, e.g. with better place and route.

of IoT open source hardware (cf. http://iis-projects.ee.ethz.ch/index.php/PULP). Furthermore, developments of the security-oriented Cheri architecture should be taken into account (http://www.cl.cam.ac.uk/research/security/ctsrd/). Even open designs for parallel computing have been produced, i.e. the Adapteva 1024 core 64-bit processor (DARPA 2017, Adapteva 2013, Olofsson 2017; see Fig. 13 for an overview of the emerging open EDA tools). To kick off the process of generating all the necessary components one could start by building an open product, based on an open CPU. This is something start-ups, industry or governments could do as a seed, to demonstrate that it is possible to pursue this path and to increase thrust.

It would also make sense to publish non-free sources as well, or to have them verified by trustworthy institutions.

ACTION POINT #3

INVESTIGATE AND DEVELOP SUCCESSFUL FOSH-BASED BUSINESS ECOSYSTEMS

An issue to be investigated from a business perspective is how to build more successful open hardware-based business ecosystems. In particular, research is needed to determine what successful business models look like in such ecosystems, what kind of businesses are a good fit for open hardware ecosystems, and what has to be done to facilitate the growth of these ecosystems. Some initial information about open hardware business models can be found in Mathilde (2014), Tincq/Benichou (2014), Cicero (2013), Thompson (2008), P2P Foundation (2017 a, b), Zimmermann (2017), Niezen (2013), Ferreira (2008) and Saenz (2010), describing concepts such as selling hardware or providing services. See Box 5 with regard to a possible market.

A large number of skilled developers are of particular importance for fighting attacks on an open source supply chain.

ACTION POINT #4

IMPROVE FOSH BUG FIXING THROUGH VOLUNTARY MEANS OR INCENTIVES

Open source code is not by itself error free. Making code public helps in scrutinizing approaches and learning from errors (cf. the case of S2N of Amazon as a replacement for OpenSSL, Wikipedia 2017a). More modern methods should be supported, such as static analysis or fuzzing techniques. Formal verification, e.g. based on a theorem prover, could help, similar to the approach applied to create seL4 (cf. Klein et al. 2010, New Scientist 2015). However, it is unclear what incentives must be put in place and what infrastructure is needed to facilitate fast security fixes and, in general, a vibrant contribution culture.

What needs to be done to ensure that open hardware always has the latest, best, and most secure features? It seems that the RISC-V community is the first to perform formal verifications, as was mentioned above (Arvind 2016, Wolf 2017b).

One could thus conduct research on mechanisms to facilitate verification, ranging from making it mandatory to encouraging companies, foundations or governments to support it.

ACTION POINT #5

RESEARCH ON REDUCING THE COSTS OF CERTIFICATION

Research on reducing the costs of certification, in particular for formally verified components, would be beneficial, such as on:

- Methods for reducing the costs of certification and verification. Open components might be more suitable for certification as the costs could be distributed among developers.
- Means for educating more computer scientists to conduct such procedures.
- Incentives to make such verifications more common.
- A sensible, stable set of verified components usable in security-critical applications.

ACTION POINT #6

EXTEND FOSH BY ADDING SECURITY PERIPHERALS

FOSH hardware can be extended by plugging additional boards, called shields, into the PCB, as the Arduino project has demonstrated, for example to get mobile network support or to connect an LCD or a camera (see https://www.arduino.cc/ and Box 4). For open hardware-based solutions to become established in security applications, a key question is how to make the hardware easily extensible by adding security peripherals, such as tamper-resistant chips.

Sometimes it will be desirable to have such a tamper-resistant module. For instance, a car manufacturer or a government may not want car users to modify the software in their cars. Or with PCs, one may want protection against mimicry attacks encouraging one to click on “ok” in order to get something promising, while in reality an illegitimate update to a security procedure would get installed. Here, a non-overwriteable, tamper-resistant module would verify the code signatures. Still, the design of such a module could be open. Sensors could be placed around the chip, much like the physical membranes around HSMs, but perhaps in a cheaper way (cf. Reith 2016; Schimmel, Hennig 2014 for novel methods of shielding) and the results from the Tampres project for improved tamper resistance could be taken into account (http://www.tampres.eu/).

Combining tamper resistance with open source might sound counterintuitive, especially considering that over the past two decades or so, most silicon manufacturers have been integrating the entire functionality into a single system-on-chip (SoC). However, integrating

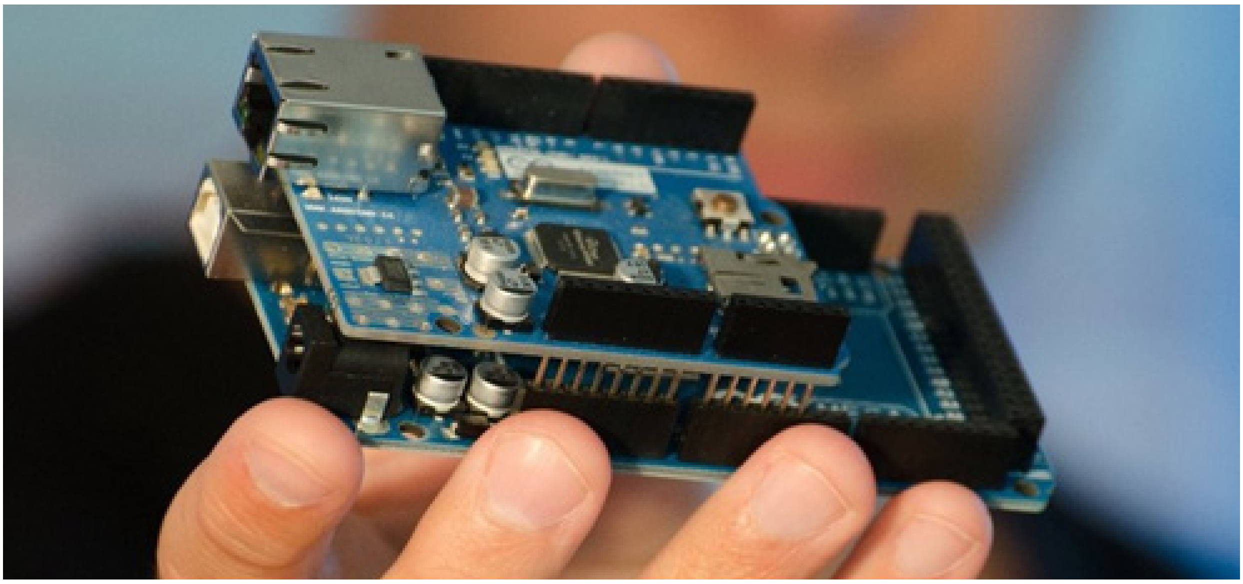


Fig 14: Arduino microcontroller platform with shield

Arduino is arguably one of the most successful open hardware projects to date (Niezen 2013). Arduino is an open-source electronics platform based on easy-to-use hardware and software. The project started in 2005 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators (Wikipedia 2017g). Arduino is open and free in the sense that all designs for the mainboard, the shields and the software are open. Some components such as CPUs or communications controllers are closed source (http://wiki.p2pfoundation.net/Open_Source_Hardware_Business_Models). Arduino comes with an easy to use developer software (the integrated development environment IDE). Arduino sold 700,000 official boards between 2006 and 2013 (Thompson 2008, Wikipedia 2017g), not to mention the copies produced by other companies.

But Arduino is not just one device; it is an open-source microcontroller platform that encourages extensions through the concept of so-called shields. Shields are boards that can be plugged on top of the Arduino PCB, thereby significantly extending its capabilities (cf. Fig. 13). Shields can provide wired and wireless communication capabilities, mobile network support (GSM), additional storage, the global positioning system (GPS), liquid crystal display (LCD), camera, motor controls, etc. (Wikipedia 2017g). The different shields follow the same philosophy as the original toolkit: they are easy to mount and cheap to produce.

Box 4: Arduino open-source hardware platform

any new or additional functionality into the chip must be justifiable from an economic perspective: there must be high-volume lead applications for the SoC that require the additional functionality. Some markets, such as the IoT market, on the other hand, are characterized by application diversity, leading to a high market fragmentation with respect to the technical requirements for individual applications and products. With high-volume new IoT applications being rather unlikely (excluding here mobile and automotive applications), the concept of a cheap, extensible open-hardware platform

> “THE PROPOSED SOLUTION HAS HUGE POTENTIAL BENEFITS OVERALL, BUT IN MY OPINION, IS VERY DISRUPTIVE FOR THE ENTRENCHED INTERESTS IN CHIP DESIGN, WHICH FORM AN OLIGOPOLY OF SORTS. THE FUNDS FOR SUCH AN ENDEAVOR ARE DEFINITELY AVAILABLE, ALMOST UNLIMITED.”
>
> **ARNAUD SAFFARI, OAK GLOBAL TECHNOLOGY, USA**

seems much more promising than the SoC paradigm. A thorough investigation is needed to determine precisely which security peripherals are needed for the IoT domain. Publicly funded research projects could kick-off the design of open, tamper-resistant modules.

ACTION POINT #7

EXAMINE IC RESILIENCE

As noted above, it has been shown that it is possible to design chips in a way that an adversary can modify a certain fraction of the gates and wires without being able to insert a Trojan horse. This approach could be examined in practice, to learn about its complexity and its costs (Seifert, Bayer 2015).

ACTION POINT #8

IMPROVE CHIP VERIFICATION

To establish trustworthiness, new methods are needed to conduct security audits of the final silicon in packaged chips. Perhaps hardware could be redesigned to be more resilient against Trojans, with redundant circuitry running in parallel.

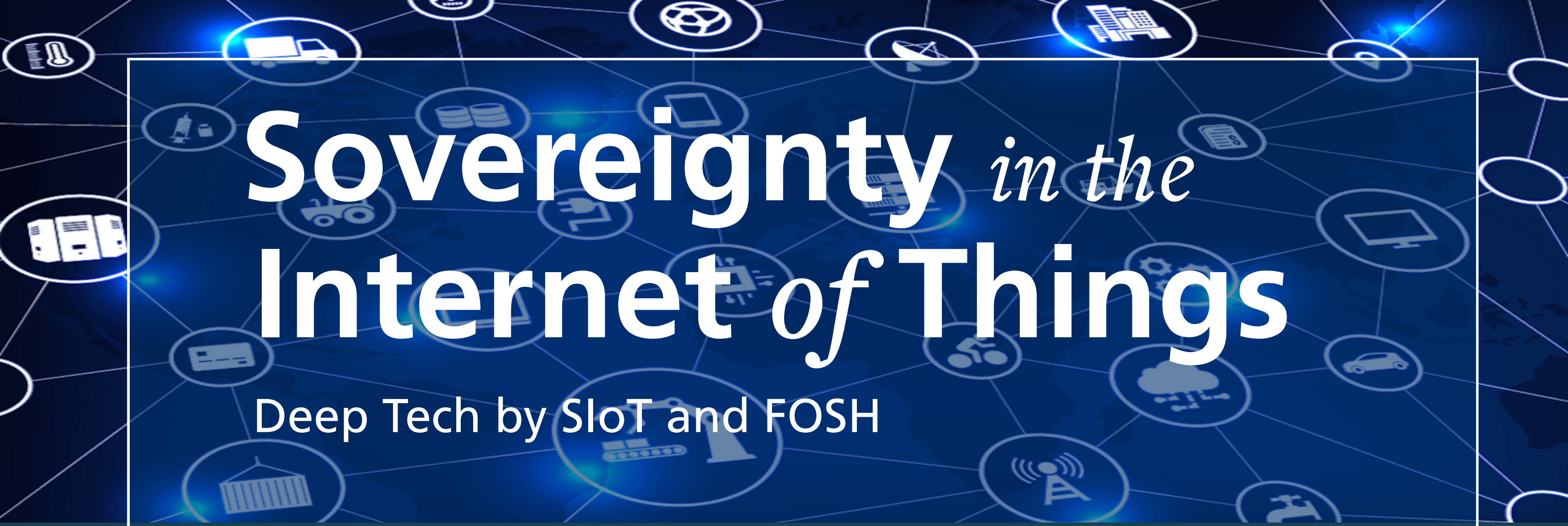
However, a malicious insider in a fab could possibly modify a chip design in some way in a minor part of the chip, causing a severe security-breach. Finally, Trojans which would be remotely triggered might be hard to detect. This problem has been worked on by, for example, the Homer project “Hardware TrOjans: Menaces et robustEsse des ciRcuits intEgrés” (http://en.pole-scs.org/projet/homere?popup=1).

The production of chips which contain evidence of their characteristics and origin has also been a topic of discussion, e.g. by using physically unclonable functions (PUFs). This could help in the effort against relabelled and grey market chips.

ACTION POINT #9

DEVELOP OPEN SOURCE EDA TOOLS

To make the whole supply chain transparent and secure and to reduce costs for open source developers, open EDA tools are essential. Enthusiasts, companies and public institutions could push the process of developing them. Examples are yosys and QFlow (see Fig. 13); also DARPA is discussing the need for a complete infrastructure (Salmon 2017). One could even run them on newly designed computers based on, e.g., RISC-V, seL4 and Linux. First versions might run in a somewhat clumsy way, but lead to trustworthy results.



Sovereignty *in the* Internet *of* Things

Deep Tech by SIoT and FOSH

*The emerging **"Internet of Things" (IoT)** is expected to encompass every aspect of our lives and to generate a paradigm shift towards a hyper-connected society. IoT devices range from sensors and security cameras to vehicles and production machines, and are used in the industry, energy, mobility, smart home, and smart city application domains. The IoT raises major security and privacy concerns. Security weaknesses in these systems not only lead to severe privacy breaches, but also have safety-relevant implications if safety-critical components are attacked. As more and more devices get connected, the number of attack vectors and possibilities for adversaries will grow rapidly. Maintaining privacy requires solid security that is built hardware-up. Bosch (2016) expects there will be 14 billion connected devices by 2022 which will enable new business models and ecosystems.*

*Devices of the Internet of Things could become a large market for sovereign FOSH devices and sovereign Deep Tech that is based on a radical open design and engineering approach. From this premisse, „**Sovereignty in the Internet of Things**" - or for short **SIoT** - has the potential to strengthen the security, safety and trustrworthiness by opening and regaining control over the supply-chain of Cyber-Physical Systems. It has the potential to contribute positively to economic profit in industry as well as the social product of nation states or political unions and associations like the EU, Latin America or ASEAN countries. SIoT might having profound impact on **Deep Tech** in segments including autonomous systems, smart cities, medical devices, clean tech, energy efficiency, smart mobility, smart production and many more developing or emerging application areas in the IoT. SIoT has enabling power and potential to catalyse change.*

Box 5: Sovereignty in the Internet of Things - Deep Tech by SioT and FOSH

One could even develop hardware first on a traditional system, and subsequently redo all the manufacturing steps ("bootstrap") in the new one (remember the approach used by CDDC).

ACTION POINT #10

FIGURE OUT OPTIONS FOR ECONOMIC PROCESSES

Economic processes should be evaluated and discussed with stakeholders, be it con-

fidentially or publicly. A priori, everything is possible, from stealthy open source-based venture capital-funded products, via public-private partnerships, to a new type of Manhattan-, Airbus- or ESA-like project.
In the above quoted US-government document, securing a communication network using local production is called a "moonshot" (Axios 2018). Since some other industries such as agriculture are subsidized for reasons of self-sufficiency, some of the costs for making sure the economy and its infrastructures do not get disrupted would be justified.

ACTION POINT #11

DEVELOP LEGAL PROPOSALS

Legal proposals, from contract clauses to legislation, would ideally be applicable in several countries to create large markets. The concrete wordings need to be worked out and evaluated. In addition, it may become necessary to identify or improve GPL for open-source hardware in a way which is acceptable to industry.

ACTION POINT #12

CREATE CONVINCING DEMONSTRATORS

To create thrust, good demonstrators could show the potential of the new approaches. Examples would protect something valuable against malware, have no backdoors, or at least much fewer surfaces where insiders or attackers could place something unwanted. A good demonstrator would, e.g. show:

- Isolation protects valuable code and data from malware, ultimately to protect lives (e.g. automotive), the social product (e.g. infrastructures) or business secrets.
- The demonstrator would be based on a good operating system, using, e.g., seL4 (cf. Heiser 2013).
- It would use an open processor, e.g. RISC-V.
- The hardware could be implemented using open source EDA tools.
- The hardware would be produced in a trustworthy fab, e.g. a local one.

So one would not start with something complex, such as a smartphone, but with something small, such as an ASIC.

ACTION POINT #13

CREATE A SUPPORTIVE GLOBAL PROCESS

Those interested in pursuing the processes sketched in this White Paper should be supported by a global working group, a think tank refining strategies, updating this White Paper, producing up-to-date supportive materials, organizing discussions, etc. To facilitate this, this text is provided in an editable format at the following URLs:

- http://www.QuattroS-Initiative.org/
- https://www.itas.kit.edu/english/projects_webe17_quattros.php
- https://github.com/SteffenReith/Quattro-S

6 SUMMARY, PLAN, AND OUTLOOK

Recall that a security problem can be anywhere, such as an error due to negligence in design (perhaps caused by controlling costs), a wrong understanding of cryptographic concepts, or an implementation or a backdoor placed on purpose, in any part of the long supply chain. Even if one could control the supply chain, staff that foils the controls could be anywhere, whether they are vulnerable to payments from a foreign nation or suffer from low motivation. Depending on one's risk estimations and the abatement costs, one can choose the countermeasures as described in Section 3.

If one had resilience by having several world centres, it would be more unlikely that all the tools and products of an industry or of an infrastructure could be compromised. Customers would select a trustworthy source. Still, skilful adversaries might undermine several tools or fabs. Also, old flaws might remain. Yet, over the years, the application of the above principles could iron out insecurities. One could actually steer the process, while in the moment, there is no process to systematically combat vulnerabilities. Box 6 provides a high-level overview of how selected, more or less risky problems can be addressed with the approaches presented before.

However, there are severe obstacles. One is the uncertainty about the economics of the open source path. The other is the efficiency of the existing computer industry, considering for example smartphone hardware and the computers used in a fab.

So what could one do? As a start, the above action points can be put in a nutshell to form a first plan:

1. Create global public discussions about the problems and the technical, economic and legislative options for creating a positive mood.
2. Make the use of open, certified or proven components and systems mandatory, e.g. for infrastructures and government use.
3. Produce prototypes of secure, open components, for instance publicly funded ones, addressing key IT problems. For example, a convincing start would comprise a small, open processor produced in a local, trustworthy fab, an open operating system, an open cryptographic toolset, with some proven components, in order to protect something valuable, such as lives, infrastructures or business secrets.
4. Produce plans for the development of free and open components throughout

the entire supply chains, including formally verified and tamper-resistant ones and tools for design automation. This could take the form of public research and industrial projects, addressing consumer products as well as industrial products and arms. Thus a large “commons” of components could be created.

5. Refine the above action points as well as this plan, based on global discussions, surveys of industry and investors, etc.

The authors think this is the world’s first plan addressing security, including problems such as APTs, as well as sovereignty, income and innovation in a comprehensive way and tackling the whole supply chain. It may well be in an early state, but there are hooks to use, such as the voluntary design of open components, the proven designs such as those explored by the armed forces, and the steps taken by industry to secure the supply chain of automotive and industrial goods.

What kind of real world developments can cause optimism that the hooks will be used and improved?

- First signs of success of use of open processors by industry players such as Nvidia and Western Digital.
- Armed forces continue to show interest in improvements.
- The fate of UNIX-producing companies, such as DEC, SCO or IBM, which died or now support Linux.
- The ongoing discussions about increasing product liabilities, as through legislation.
- Since the Snowden revelations, citizens, industry and governments have become increasingly aware of the risks and the issues surrounding the IT industry.
- Investors might see a chance to invest under the new paradigm of openness from the fab to the end user, e.g. with “Made in the EU”.
- Large funds are available, remember the cases of Softbank and Abu Dhabi.
- Last but not least, as the Japanese have shown, one should copy the culture of “gambaremasu”: do not give up, look at solutions, see the efforts made in China and Russia.

The authors intend to push for improvements, encourage others to participate and hope to be able to tick off some action points in the future.

Examples of problems	Damage	Some components of a solution
ATTACKS ON „CIA“		
Malware causing, e.g., DDoS, extortion	Large, cf. Mirai, WannaCry	Open source components, legislation, economics
Disrupting infrastructures	Severe, cf. Ukraine, Stuxnet	Secure entire supply chain
Eavesdropping on business secrets	Large, cf. Gemalto, Belgacom	Secure entire supply chain, split devices
SAFETY PROBLEMS		
Car hacks	Casualties, damage to reputation	Verification of IP cores and higher layers
Fake & mislabelled chips	Defects, e.g. in defence, automotive	Local production
LACK OF INCOME		
Lack of local value creation	Large, cf. Nokia	Control of supply chain, legislation, economics
Theft of chip designs	Large, e.g. in car industry	Local fab, legislation
Supply chain interruption	Large	Local fab, legislation

Box 6: Stylized key problems and solutions.

REFERENCES

Abesamis, C.; Leblanc, M.: NSA Counterfeit Parts Awareness and Inspection. 2015. https://mttc.jpl.nasa.gov/files/NASA%20Counterfeit%20Training_unlimited%20distribution%20handout.pdf

Achenbach, D.; Müller-Quade, J.; Rill, J.: Universally Composable Firewall Architectures using Trusted Hardware. BalkanCryptSec 2014.
http://link.springer.com/chapter/10.1007%2F978-3-319-21356-9_5

Adapteva: Epiphany Architecture Reference. 2013.
http://adapteva.com/docs/epiphany_arch_ref.pdf

Adee, S.: The Hunt for the Kill Switch. Are chip makers building electronic trapdoors in key military hardware? The Pentagon is making its biggest effort yet to find out. May 1, 2008. http://spectrum.ieee.org/semiconductors/design/the-hunt-for-the-kill-switch

Appelbaum, J.: NSA ANT Catalog. 2013.
https://www.eff.org/files/2014/01/06/20131230-appelbaum-nsa_ant_catalog.pdf

Arduino 2017: https://www.arduino.cc/

Arvind: Formal Specification of RISC V Uniprocessor Consistency. 2016.
https://riscv.org/wp-content/uploads/2016/07/Tue0930_160712RISCYspecs.pdf

Asanovic, K.; Patterson, D.: Instruction Sets Should Be Free: The Case For RISC-V. 2014.
https://www2.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-146.pdf

Axios: Scoop: Trump team considers nationalizing 5G network. Jan. 28, 2018.
https://www.axios.com/trump-team-debates-nationalizing-5g-network-f1e92a49-60f2-4e3e-acd4-f3eb03d910ff.html

Baikal Electronics: Baikal-T1 SoC Extended Performance Benchmarking Results are Revealed. 15 August 2017.
https://www.baikalelectronics.com/about/press-center/news/BenchmarksT1-2017/

Becker, G.; Kasper, M.; Paar, Ch.: Side-Channel based Watermarks for IP Protection. Presentation given at COSADE 2010.

Becker, G.; Regazzoni, F.; Paar, Ch.; Burleson, W.: Stealthy dopant-level hardware Trojans: extended version. J Cryptogr Eng (2014) 4:19–31

Bhunia, S.; Ray, S.; Sur-Kolay, S. (eds.): Fundamentals of IP and SoC Security. Cham 2017.

Boeing: HACMS: High Assurance Cyber Military System. 2015. Cf.
https://www.facebook.com/BoeingCareers/videos/1167433693271186/

Bosch: Robert Bosch GmbH, Bosch Smart Home – Annual Report, 2016: http://annual-report.bosch.com/smarthome2016

Brookings Institution: Building a secure 5G network without nationalization. 2018. https://www.brookings.edu/blog/techtank/2018/01/29/building-a-secure-5g-network-without-nationalization/

Celio, C.; Chiu, P.-F.; Nikolic, B.; Patterson, D.; Asanović, K.: BOOM v2: an open-source out-of-order RISC-V core. 2017. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-157.pdf

Chaum, D.: Achieving Electronic Privacy. Scientific American, August 1992, p. 96-101.

Cicero, S.: The Truth About Open Source Hardware Business Models. 2013. http://www.open-electronics.org/the-truth-about-open-source-hardware-business-models/

Cimpalu, C.: Researchers Find a Way to Disable Much-Hated Intel ME Component Courtesy of the NSA. August 28, 2017. https://www.bleepingcomputer.com/news/hardware/researchers-find-a-way-to-disable-much-hated-intel-me-component-courtesy-of-the-nsa/

Clarke, R.; Knake, R.: Cyber War: The Next Threat to National Security and What to Do About It. New York 2010

Chlipala, A.: Coming Soon: Machine-Checked Mathematical Proofs in Everyday Software and Hardware Development. Chaos Computer Club 2017. https://events.ccc.de/congress/2017/Fahrplan/events/9105.html

Corbet, Jonathan: Is it time for open processors? 2018. https://lwn.net/Articles/743602/

Covington: China Releases Final Regulation on Cybersecurity Review of Network Products and Services. 2017. https://www.cov.com/-/media/files/corporate/publications/2017/05/china_releases_final_regulation_on_cybersecurity_review_of_network_products_and_services.pdf

Dalton, C.: A hypervisor against ferrying away data. Interview by Franco Furger and Arnd Weber. OpenTC Newsletter (2009), http://www.itas.kit.edu/pub/m/2009/webe09b.htm

DARPA: Mr. Andreas Olofsson. 2017. http://www.darpa.mil/staff/mr-andreas-olofsson

Data61: SMACCM. 2017. https://ts.data61.csiro.au/projects/TS/SMACCM/

Domas, C.: Breaking the x86 ISA. Blackhat 2017 (a). https://www.blackhat.com/docs/us-17/thursday/us-17-Domas-Breaking-The-x86-Instruction-Set-wp.pdf

Domas, C.: Sandsifter. 2017 (b). https://github.com/xoreaxeaxeax/sandsifter

Edwards, Tim: 2017. http://opencircuitdesign.com/qflow/welcome.html

EETimes: SoC From Russia With MIPS. 27.5.15. http://www.eetimes.com/document.asp?doc_id=1326694

Empyrian: IC EDA Solution. 2017. http://www.empyrean.com.cn/_english/index.php?m=content&c=index&a=lists&catid=72

European Commission: Open source software strategy. 2017. https://ec.europa.eu/info/european-commissions-open-source-strategy_en#actionplan

Evans, B.: Beginning Arduino Programming: 2011. Apress, New York.

Feldmann, A.; Heyder, P.; Kreutzer, M.: Schmid, S.: Seifert, J.-P.; Shulman, H.; Thimmaraju, K.: Waidner, M.; Sieberg, J.: NetCo: Reliable Routing With Unreliable Routers. IEEE/IFIP Inter-national Conference on Dependable Systems and Networks (DSN) Workshops, 2016. Toulouse

Ferreira, E.: Open Hardware Business Models. 2008. https://timreview.ca/article/136

Ford, D.: Cheney's defibrillator was modified to prevent hacking. 2013.

http://edition.cnn.com/2013/10/20/us/dick-cheney-gupta-interview/

Francq, J.: Hardware Trojans Detection Methods. 2013. http://www.hint-project.eu/downloads/NEW_TRUDEVICE.pdf

Furger, F.: Informatik-Innovationen aus der Schweiz? – Lilith/Diser und Oberon. Zürich 1993

Goetz, J.; Rosenbach, M.; Szandar, A.: War of the Future. National Defense in Cyberspace. 2/11/2009. http://www.spiegel.de/international/germany/war-of-the-future-national-defense-in-cyberspace-a-606987.html

Golem: Simko 3. Telekom dementiert Abschaffung des Merkel-Handys. 8.10.2014. http://www.golem.de/news/simko-3-telekom-dementiert-abschaffung-des-merkelphones-1410-109703.html

Grand, J.: Hardware Reverse Engineering: Access, Analyze, & Defeat. 2011. https://media.blackhat.com/bh-dc-11/Grand/BlackHat_DC_2011_Grand-Workshop.pdf

Greenberg, A.: Hackers Remotely Kill a Jeep on the Highway - With Me in It. 2015. https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway

Greenberg, A.: How An Entire Nation Became Russia‘s Test Lab for Cyberwar. Wired, 20.6.2017. https://www.wired.com/story/russian-hackers-attack-ukraine/

Guerreschi, R.: Trustless. 2016. https://www.slideshare.net/rguerreschi/trustless-slides-dualuse-rd-proposal

Guerreschi, R.: 2017. http://www.openmediacluster.com

Halfacree, G.: (photo) 2017. https://www.flickr.com/photos/120586634@N05/32418933415 32418933415_807a55e6f8_h.jpg

Healey, J.; Jordan, K.: NATO's Cyber Capabilities: Yesterday, Today, and Tomorrow. September 2014. http://www.atlanticcouncil.org/images/publications/NATOs_Cyber_Capabilities.pdf

Heise Online: Siemens erweitert für 4,2 Milliarden Euro sein Industriesoftware-Knowhow. 2017. https://www.heise.de/newsticker/meldung/Siemens-erweitert-fuer-4-2-Milliarden-Euro-sein-Industriesoftware-Knowhow-3672142.html

Heiser, G.: Protecting eGovernment Against Attacks, Sydney 2013 (White Paper). https://www.itas.kit.edu/downloads/projekt/projekt_webe12_cosiso_heiser_paper.pdf

Heiser, G.: Verified software can (and will) be cheaper than buggy stuff. 16.6.2016. https://microkerneldude.wordpress.com/

Heiser, G.; Yarom, Y.: Insecure by design – lessons from the Meltdown and Spectre debacle. Feb. 4, 2018. https://theconversation.com/insecure-by-design-lessons-from-the-meltdown-and-spectre-debacle-90629

Horne, B.: Trust Me. Trust Me Not. IEEE Security & Privacy 2016.

Homer project: Hardware TrOjans: Menaces et robustEsse des ciRcuits intEgrés http://en.pole-scs.org/projet/homere?popup=1.

Hu, W.: A Brief Introduction to CPU Development in China. Beijing 2010. http://www.telecomitalia.com/content/dam/telecomitalia/documents/innovation/it/eventi/progetto_unesco/4.Introduction_to_CPU_development_China.pdf

Huang, A.: Keynote Address: Impedance Matching Expectations Between RISC-V and the Open Hardware Community. Risc-V Shanghai 2017. https://riscv.org/wp-content/uploads/2017/05/Wed1100-impedancematch-huang.pdf, https://www.youtube.com/watch?v=zXwy65d_tu8

Huang, A.: Novena. 2018. https://www.bunniestudios.com/blog/?tag=novena

Hughitt, B.: Counterfeit Electronic Parts. 2010.

https://nepp.nasa.gov/workshops/etw2010/talks/08_Hughitt_Counterfeit%20Electronics%20-%20All%20the%20World%27s%20a%20Fake.pdf

ICS-CERT: Cyber-Attack Against Ukrainian Critical Infrastructure. February 25, 2016. https://ics-cert.us-cert.gov/alerts/IR-ALERT-H-16-056-01

Intercept: The Great SIM Heist. 2015. https://firstlook.org/theintercept/2015/02/19/great-sim-heist/

Jin, Y.: Introduction to Hardware Security. Review. In: Electronics 2015, 4, 763-784

Kang, J.: Freedom E310 & Low-Cost HiFive1 Development Board. 2016. https://riscv.org/wp-content/uploads/2016/12/Tue0930-FE310_HiFive1_v2-Kang-SiFive.pdf

Karri, R.; Koushanfar, F.: Trustworthy Hardware [Scanning the Issue]. In: Proceedings of the IEEE 102 (8), 2014, S. 1123–1125.

Kelley, M.: The Stuxnet Attack On Iran's Nuclear Plant Was 'Far More Dangerous' Than Previously Thought. 2013. http://www.businessinsider.com/stuxnet-was-far-more-dangerous-than-previous-thought-2013-11

Khandelwal, S.: IoT Botnet. 25,000 CCTV Cameras Hacked to Launch DDoS Attack. June, 28, 2016 (a). http://thehackernews.com/2016/06/cctv-camera-hacking.html

Khandelwal, S.: World's Largest 1 Tbps DDoS Attack Launched from 152,000 Hacked Smart Devices. Sept. 27, 2016 (b). http://thehackernews.com/2016/09/ddos-attack-iot.html

Klein, G.; Andronick, J.; Elphinstone, K.; Heiser, G.; Cock, D.; Derrin, P.; Elkaduwe, D.; Engelhardt, K.; Kolanski, R.; Norrish, M.; Sewell, T.; Tuch, H.; Winwood, S. (2010): **seL4: Formal Verification of an Operating-system Kernel.** In: Commun. ACM 53(6), S. 107-115

Klein, G.; Andronick, J.; Kuz, I.; Murray, T.; Heiser, G.; Fernandez, M.: Formally Verified Software in the Real World. To appear in: Communications of the ACM, 2018. https://ts.data61.csiro.au/publications/csiroabstracts/Klein_AKMHF_toappear.abstract.pml

Kocher, P.; Genkin, D.; Gruss, D.; Haas, W.; Hamburg, M.; Lipp, M.; Mangard, S.; Prescher, T.; Schwarz, M.; Yarom, Y.: Spectre Attacks: Exploiting Speculative Execution. 2018. https://spectreattack.com/spectre.pdf

Kommission (1991). Mitteilung der Kommission der Europäischen Gemeinschaft über die Europäische Elektronik- und Informatikindustrie. SEK(91) 565 endg.; Ratsdok. 5653/91. Bundesrat Drucksache 258/91, April 19, 1991. http://dipbt.bundestag.de/doc/brd/1991/D258+91.pdf.

Koscielnicki, M.: nVidia Hardware Documentation, 2016. https://media.readthedocs.org/pdf/envytools/latest/envytools.pdf

Lacoste, G.; Pfitzmann, B.; Steiner, M.; Waidner, M. (eds.): SEMPER - Secure Electronic Marketplace for Europe. Springer Lecture Notes in Computer Science 1854, Springer, 2000

Leaksource 2013: http://leaksource.wordpress.com/2013/12/30/nsas-ant-division-catalog-of-exploits-for-nearly-every-major-software-hardware-firmware/

Leef, S.: Presentation at STARSS Workshop. May 21, 2014. https://www.src.org/calendar/e005440/

Li, X.: Chip Design and EDA in China. 2013. http://www.carch.ac.cn/~lxw/

Liang, Q.; Wang, X.: Unrestricted Warfare. Beijing 1999

Lin, L., Kasper, M., Güneysu, T., Paar, C., Burleson, W.: Trojan side-channels: lightweight hardware Trojans through side-channel engineering. In: Cryptographic hardware and embedded systems- CHES 2009, LNCS, pp. 382–395. Springer, Heidelberg (2009)

Lipp, M.; Schwarz, M.; Gruss, D.; Prescher, T.; Haas, W.; Mangard, s.; Kocher, P.; Genkin, G.; Yarom, Y.; Hamburg, M.: Meltdown. 2018.
https://meltdownattack.com/meltdown.pdf

Love, E., Jin, Y., Makris, Y.: Proof-carrying hardware intellectual property: a pathway to trusted module acquisition. IEEE Trans. Inf. Forensics Secur. 2012, 7(1), 25–40

Marena, T.; Gerstl, S.: RISC-V und sein Potential für den Industriemarkt. 6.7.2017.
http://www.elektronikpraxis.vogel.de/hardwareentwicklung/pld-und-asic/articles/622665/

Massacci, M.; Ruprai, R.; Collinson, M.; Williams, J.: Economic Impacts of Rules- versus Risk-Based Cybersecurity Regulations for Critical Infrastructure Providers. IEEE Security & Privacy 2016.

Mathilde: Open Source Hardware Business Models. 2014.
http://makingsociety.com/2014/12/open-source-hardware-business-model

Matt: What is Suter? October 5, 2007.
http://www.1913intel.com/2007/10/05/what-is-suter/

McCormack, R.: DOD, NSA Enter A New World Order: U.S. Is Now Dependent On Foreign Com-pa-nies For Its Most Sensitive Electronics. May 31, 2016.
http://www.manufacturingnews.com/news/2016/Trusted-Foundry-0531161.html

Mentor Graphics: Siemens to expand its digital industrial leadership with acquisition of Mentor Graphics. 2016.
https://www.mentor.com/company/news/siemens-to-expand-its-digital-industrial-leadership-with-acquisition-of-mentor-graphics

Merritt, R.: India Preps RISC-V Processors. Two 64-bit projects run in parallel. 1/27/2016. https://www.eetimes.com/document.asp?doc_id=1328790

Microsoft 2017: What is the Security Development Lifecycle?
https://www.microsoft.com/en-us/sdl/default.aspx

Minnich, R.: Replace your exploit ridden firmware with Linux. 2017. https://ossna2017.sched.com/event/BCsr/replace-your-exploit-ridden-firmware-with-linux-ronald-minnich-google

Mitra, S.; Wong, H.-S.; Wong, S.: Stopping Hardware Trojans in Their Tracks. IEEE Spectrum 2015.
http://spectrum.ieee.org/semiconductors/design/stopping-hardware-trojans-in-their-tracks

Mitre: CVE-2017-5754 (2017a):
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=2017-5754

Mitre: CVE-2017-5753 (2017b):
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=2017-5753

Mitre: CVE-2017-5715 (2017c):
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=2017-5715

Miyamoto, I.; Holzer, T.; Sarkani, S.: Why a counterfeit risk avoidance strategy fails. Computers & Security 2017, 81-96

Networkworld: DARPA to eliminate "patch & pray" by baking chips with cybersecurity fortification. 10.4.2017. http://www.networkworld.com/article/3188632/security/darpa-to-eliminate-patch-and-pray-by-baking-chips-with-cybersecurity-fortification.html

New Scientist: Unhackable kernel could keep all computers safe from cyberattack. Sept. 16, 2015.
https://www.newscientist.com/article/mg22730392-600-unhackable-kernel-could-keep-all-computers-safe-from-cyberattack-2/

New York Times: New Rules in China Upset Western Tech Companies. Jan 28, 2015.

http://www.nytimes.com/2015/01/29/technology/in-china-new-cybersecurity-rules-perturb-western-tech-companies.html?ref=business

New York Times: China's Technology Ambitions Could Upset the Global Trade Order. 7.11.2017. https://www.nytimes.com/2017/11/07/business/made-in-china-technology-trade.html

Niezen, G.: Ten Commercially Successful Open Source Hardware Projects in 2013. 2013. https://medium.com/@gendor/10-commercially-successful-open-source-hardware-projects-in-2013-515a799daa89#.n0oqwe83c

Niezen 2013, CC-BY FIT Lab Swansea

Oleksiuk, D.: Exploiting UEFI boot script table vulnerability. February 6, 2015. http://blog.cr4.sh/2015/02/exploiting-uefi-boot-script-table.html

Olofsson, A.: What I learned building a parallel processor company from scratch. HIPEAC 2017

Open Processor Foundation 2017: J-cores. http://opf.org/j-core.html

Ortega, A.; Muñiz, S.: Hardware Trojans and Malicious Logic. Presentation at RSA Conference 2014. https://www.rsaconference.com/writable/presentations/file_upload/hta-t07b-hardware-trojans-and-malicious-logic.pdf

P2P Foundation: Open Source Hardware Business Models. 2017 (a). http://wiki.p2pfoundation.net/Open_Source_Hardware_Business_Models

P2P Foundation: Arduino -- Business Model. 2017 (b). http://wiki.p2pfoundation.net/Arduino_-_Business_Model

Peter, M.; Nordholz, J.; Petschick, M.; Danisevskis, J.; Vetter, J.; Seifert, J. –P.: Undermining Isolation through Covert Channels in the Fiasco.OC Microkernel. Cryptology ePrint Archive: Report 2014/984. http://eprint.iacr.org/2014/984

Pfitzmann, A.: Contra Online-Durchsuchung. Informatik Spektrum 31(1): 65-69 (2008)

Pfleeger, Ch.; Pfleeger, S.; Jonathan, M. (2015): Security in Computing: Prentice Hall Professional Technical Reference.

Phillips, E.: Reverse Engineering Silicon Logic. 2008. http://hackaday.com/2008/09/13/reverse-engineering-silicon-logic

Qiu, Yingxin; Li, Xiaowei; Wang, Tiancheng: Property Coverage Analysis Based Trustworthiness Veri-fication for Potential Threats from EDA Tools. 2016. http://ieeexplore.ieee.org/document/7796079/

Reith, St.: New Directions of Hardware Based Cryptographic Modules for Modern Cars. Presentation given at IT Security for Vehicles. Düsseldorf 2016.

Reith, St.: J1Sc - A simple reimplementation of the J1 CPU in Scala using Spinal HDL. 2017. https://github.com/SteffenReith/J1Sc

Reutlinger Nachrichten: Bosch baut Chip-Fabrik für eine Milliarde Euro in Dresden. 19.6.2017. http://www.swp.de/reutlingen/lokales/reutlingen/bosch-baut-_chip-fabrik-fuer-eine-milliarde-euro-in-dresden-15261276.html

RISC-V Foundation: Specifications. 2017. https://riscv.org/specifications/

Saenz, A.: 13 Open Source Hardware Companies Making $1 Million or More. https://singularityhub.com/2010/05/10/13-open-source-hardware-companies-making-1-million-or-more-video/

Salmon, L.: A Perspective on the Role of Open-Source IP In Government Electronic Systems. RISC-V Workshop 2017. https://content.riscv.org/wp-content/uploads/2017/12/Wed-1042-RISCV-Open-Source-LintonSalmon.pdf

Schimmel, O.; Hennig, M.: Kopier- und Manipulationsschutz für eingebettete Systeme. Datenschutz und Datensicherheit 2014, 742-746

Schneier, B.: Did NSA Put a Secret Backdoor in New Encryption Standard? 2007. https://www.schneier.com/essays/archives/2007/11/did_nsa_put_a_secret.html

Schneier, B.: Someone Is Learning How to Take Down the Internet. Crypto-Gram, Sept. 15, 2016

Schneier, B.: Security and the Internet of Things. Crypto-Gram, Feb. 15, 2017

Snowden, E.: @Snowden. Oct. 28, 2015 (b)

Schulzki-Haddouti, Ch.: Die US-Geheimdienste wollen sich Zugang zu verschlüsselten Daten verschaffen – weltweit. In: Die Zeit, 31.12.1998. http://www.zeit.de/1998/39/199839.c_krypto_.xml

Secelements. 2018. http://secelements.com/

Seifert, J.-P.; Bayer, C.: Trojan-Resilient Circuits. In: Pathan, A. (ed.): Securing Cyber-Physical Systems, Boca Raton 2015

Sharma, S.: An ARM killer from IIT, Madras? November 9, 2017. https://factordaily.com/india-chip-design-shakti-iit-madras/

SiFive: Freedom E310. 2017. https://www.sifive.com/products/freedom-e310/

Sijstermans, F.: RISC V at NVIDIA. 6th RISC-V Workshop, Shanghai, May 2017. https://www.youtube.com/watch?v=g6Z_5l69keI

Skorobogatov, S.; Woods, Ch.: Breakthrough Silicon Scanning Discovers Back-door in Military Chip, CHES 2012. https://www.cl.cam.ac.uk/~sps32/ches2012-backdoor.pdf

Snowden, E.: Schlachtfeld Internet. 12.1.2015 (a) in: http://www.ardmediathek.de/tv/Reportage-Dokumentation/Die-Story-im-Ersten-Schlachtfeld-Intern/Das-Erste/Video?documentId=25812360

Snowden, E. : Snowden. Oct. 28, 2015 (b)

Solon, O.: Team of Hackers Take Remote Control of Tesla Model S from 12 Miles Away. 2016. https://www.theguardian.com/technology/2016/sep/20/tesla-model-s-chinese-hack-remote-control-brakes

Spiegel: Computer: Anschluß gesucht. 10.10.1966. http://www.spiegel.de/spiegel/print/d-46414527.html

Spiegel: Feind hört mit. 11.2.2004. http://www.spiegel.de/netzwelt/web/krypto-handys-feind-hoert-mit-a-285618.html

Starbug; Nohl, K.: Hardware Reverse Engineering. 2008. https://events.ccc.de/congress/2008/Fahrplan/attachments/1218_081227.25C3.HardwareReversing.pdf

Sugawara, T.; Suzuki, D.; Fujii, R.; Tawa, S.; Hori, R.; Shiozaki, M.; Fujino, T.: Reversing Stealthy Dopant-Level Circuits. https://eprint.iacr.org/2014/508.pdf

Tass: Russia wants to replace US computer chips with local processors. Moscow, June 19, 2014. http://tass.com/economy/736804

Techdesignforums 2017: http://www.techdesignforums.com/practice/technique/the-wally-rhines-interview-part-two-ai-automotive-security/ May 15, 2017

Techpowerup: NSA Hides Spying Backdoors into Hard Drive Firmware. February 17th 2015. http://www.techpowerup.com/209925/nsa-hides-spying-backdoors-into-hard-drive-firmware.html

TechTime: GSA: The Semiconductor Industry must reinvent itself in order to survive. 2016: http://news.techtime.co.il/2016/03/30/semiconductors-4/

Tehranipoor, M., Koushanfar, F.: A survey of hardware Trojan taxonomy and detection. IEEE Design and Test of Computers, 2009.

The Register: Softbank tears off chunk of ARM, feeds it to hungry Saudis. 8 Mar 2017. https://www.theregister.co.uk/2017/03/08/quarter_arm_sold_saudi_investment_fund_reports/

Thomas, O.: Advanced IC Reverse Engineering Techniques: In Depth Analysis of a Modern Smart Card. 2015. https://www.blackhat.com/docs/us-15/materials/us-15-Thomas-Advanced-IC-Reverse-Engineering-Techniques-In-Depth-Analysis-Of-A-Modern-Smart-Card.pdf

Thomas, O.: Chip Reverse Engineering Process. 2017. http://www.texplained.com/process

Thompson, C.: Build it. Share it. Profit. Can Open Source Hardware Work? 2008. https://www.wired.com/2008/10/ff-openmanufacturing/

Thompson, K.: Reflections on trusting trust. Communications of the ACM 1984

Tincq, B.; Benichou, L.: Open Hardware Business Models. 2014. http://www.slideshare.net/btincq/business-models-for-open-source-hardware

Triolo et al.: China's Strategic Thinking on Building Power in Cyberspace: A Top Party Journal's Timely Explanation Translated. 25.9.2017. https://www.newamerica.org/cybersecurity-initiative/blog/chinas-strategic-thinking-building-power-cyberspace/

Trustless. 2017. https://trustless.ai/

Trustworthy Systems: Cross Domain Desktop Compositor. 2017. h ttp://ts.data61.csiro.au/projects/TS/cddc.pml

Tsunoda, R.; de Bary, W.; Keene, D. (eds.): Sources of Japanese Tradition. Columbia University Press 1964

Vandewege, W.; Garrett, M.; Stallman, R.: „Active Management Technology": The obscure remote control in some Intel hardware. https://www.fsf.org/blogs/community/active-management-technology

Vishik et al.: Hardware Threat Landscape and Good Practice Guide. 2017. https://www.enisa.europa.eu/publications/hardware-threat-landscape

Waidner, M.; Backes, M.; Müller-Quade, J.: Sicherheitstechnik im IT-Bereich. Positionspapier aus Forschungssicht. Darmstadt 2013. https://www.sit.fraunhofer.de/fileadmin/dokumente/studien_und_technical_reports/Positionspapier_IT-Sicherheit_Forschung.pdf

Washington Post: New details emerge about 2014 Russian hack of the State Department: It was 'hand to hand combat'. April 3, 2017. https://www.washingtonpost.com/world/national-security/new-details-emerge-about-2014-russian-hack-of-the-state-department-it-was-hand-to-hand-combat/2017/04/03/d89168e0-124c-11e7-833c-503e1f6394c9_story.html?utm_term=.c49ce85bd407

Washington Post: National Security Council official behind 5G memo leaves White House. Feb. 2, 2018. https://www.washingtonpost.com/news/josh-rogin/wp/2018/02/02/national-security-council-official-behind-5g-memo-leaves-white-house/?utm_term=.c2d86b90524e

Weber, A.: Full Bindingness and Confidentiality. Requirements for Secure Computers, and Design Options. In: Hansen, H.; Bichler, M.; Mahrer, R.: Proceedings of the 8th European Confer-ence on Information Systems, ECIS 2000. A Cyberspace Odyssey, Vienna 2000, 889-896

Weber, A.; Scuka, D.: Comment: Operators at crossroads: market protection or innovation? In: "Telecommunications Policy", Volume 40, Issue 4, April 2016, Pages 368–377

Western Digital: Western Digital to Accelerate the Future of Next-generation Computing Architec-tures for Big-data and Fast Data Environments. 28.11.2017. https://www.wdc.com/about-wd/newsroom/press-room/2017-11-28-western-digital-to-accelerate-the-future-of-next-generation-computing-architectures-for-big-data-and-fast-data-environments.html

Wikipedia: Heartbleed. 2017a. http://en.wikipedia.org/wiki/Heartbleed

Wikipedia: Intel Active Management Technology. 2017b. https://en.wikipedia.org/wiki/Intel_Active_Management_Technology

Wikipedia: Data Remanence. 2017c. https://en.wikipedia.org/wiki/Data_remanence

Wikipedia: Open-source hardware. 2017d. https://en.wikipedia.org/wiki/Open-source_hardware

Wikipedia: RISC-V. 2017e. https://en.wikipedia.org/wiki/RISC-V

Wikipedia: Linus's Law. 2017f. https://en.wikipedia.org/wiki/Linus‘s_Law

Wikipedia: Arduino. 2017g. https://en.wikipedia.org/wiki/Arduino

Wikipedia: Globalfoundries. 2017h. https://de.wikipedia.org/wiki/Globalfoundries

Wikipedia: Zuse KG. 2017i. https://de.wikipedia.org/wiki/Zuse_KG

Wired: How a dorm room Minecraft scam brought down the Internet. Dec. 13, 2017. https://www.wired.com/story/mirai-botnet-minecraft-scam-brought-down-the-internet/

Wolf, C.: Yosys Open Synthesis Suite. 2017a. http://www.clifford.at/yosys/

Wolf, C. (**2017b): End-to-end formal ISA verification of RISC-V processors with riscv-formal.** https://riscv.org/2017/10/7th-risc-v-workshop-agenda/,
http://www.clifford.at/papers/2017/riscv-formal/slides.pdf

Wong, W.: First Open-Source RISC-V Chip Arrives. 2016. http://electronicdesign.com/microcontrollers/first-open-source-risc-v-chip-arrives

Xiao, K., Forte, D., Jin, Y., Karri, R., Bhunia, S., & Tehranipoor, M. (**2016**). **Hardware trojans: Lessons learned after one decade of research.** ACM Transactions on Design Automation of Electron-ic Systems, 22(1), [6]

Xie, J.: NVIDIA RISC-V Story. 4th RISC-V Workshop, 2016. https://riscv.org/wp-content/uploads/2016/07/Tue1100_Nvidia_RISCV_Story_V2.pdf, https://youtu.be/gg1lISJfJIo

Yang, K.; Hicks, M.; Dong, Q.; Austin, T.; Sylvester, D.: A2: Analog Malicious Hardware. 2016 IEEE Symposium on Security and Privacy.
http://www.ieee-security.org/TC/SP2016/papers/0824a018.pdf

Zeit Online: Chaos Computer Club. Anders als erwartet. 27.12.15.
http://www.zeit.de/digital/datenschutz/2015-12/32c3-hardware-trojaner

Zetter, K.: Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid. 03.03.16. https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/

Zimmermann, L.: Business Models for Open Source Hardware. 2017.
http://bloglz.de/business-models-for-open-source-hardware-open-design

ABBREVIATIONS

This list of abbreviations and their explanations has been produced by borrowing from Wikipedia and by using own definitions supposed to fit to the issues discussed in this document.

ASIC: Application-specific integrated circuit.
AMT: Intel‘s Active Management Technology, part of ME. See under ME.
APT: Advanced persistent threat, a sophisticated attack technique using a remote control system.
CDMA: Code-division multiple access, a technique for spectrum sharing using different codes.
CNE: Computer network exploitations, pre-implanted, exploitable characteristics, e.g. code.
Common Criteria: a standard used for security evaluations of different levels.
Compiler: Software which translates given code, typically from source code to machine code.
CPU: Central processing unit, carrying out all instructions.
CPS: Cyber-Physical Systems
DARPA: US Defense Advanced Research Projects Agency.
DDoS: Distributed denial of service attack, e.g. flooding a machine with information.
EDA: Electronic design automation, tools used to convert abstract hardware descriptions into real hardware.
Fab: Semiconductor fabrication plant.
FOSH: Free and open-source hardware.
FOSS: Free and open-source software.
FPGA: Field-programmable gate array, an integrated circuit to be configured after manufacturing.
Gbps: Gigabits per second
GPL: General Public License, gives users the freedom to share or modify software.
GPU: Graphics processing unit.
HDL: Hardware description language.
IC: Integrated circuit.
ICS: Industrial control systems
ISA: Instruction set architecture (for a processor), the formal structure of a CPU and its instructions.

IoT: Internet of Things.
IP core: Intellectual property core, a reusable piece of chip layout design.
IT: Information technology.
LCD: Liquid crystal display.
ME: Intel's Management Engine, allowing remote computer administration even when the machine is asleep and no operating system running.
NERF: Non-extensible reduced firmware, an open, reduced version of UEFI (see under UEFI).
OpenSSL: Open secure sockets layer, a set of software functions to encrypt, decrypt and authenticate data to be transmitted over the internet.
OS: Operating system.
PCB: Printed circuit board.
PLC: Programmable logic controller, a computer controlling a part of a manufacturing process
PUF: Physically uncloneable functions, a stochastic pattern in a chip, usable as a unique ID.
SCADA: Supervisory control and data acquisition (system) for high-level process management
seL4: Secure L4, a proven variant of the L4 operating system kernel; originally Jochen Liedtke's forth version of a kernel.
SoC: System on chip.
RISC-V: "Reduced instruction set computing", fifth version.
RTL: Register transfer level, a high level representation of a circuit, which can be translated automatically into real hardware by using EDA tools.
Tbps: Terabits per second.
Trojan horse: A hidden function in a piece of soft- or hardware, like the mythological wooden horse containing soldiers.
UEFI: Unified extensible firmware interface, an interface between firmware and operating system.
VHDL/Verilog: Hardware description languages which are used to describe the behaviour and/or the structure of a circuit. Such a description can serve as a starting point to build a semiconductor chip.
Zero-day exploit: A weakness in a system, typically in software, which has not yet been exploited and which is only known to the attacker.