HUMBOLDT-UNIVERSITÄT ZU BERLIN
MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT
INSTITUT FÜR INFORMATIK

A Post-Attack Recovery Architecture for Smart Electricity Meters

Eine Architektur zur Kontrollwiederherstellung nach Angriffen auf Smart Metering in Stromnetzen

Masterarbeit

zur Erlangung des akademischen Grades
Master of Science (M. Sc.)

eingereicht von: Jan Sebastian Götte
geboren am: [redacted]
geboren in: [redacted]

Gutachter/innen: Prof. Dr. Björn Scheuermann
Prof. Dr.-Ing. Eckhard Grass

eingereicht am: .... verteidigt am: ....

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und noch nicht für andere Prüfungen eingereicht habe. Sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, sind als solche kenntlich gemacht. Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Berlin, den 6. Juli 2020 ....

This work is licensed under a Creative-Commons "Attribution-ShareAlike 4.0 International" license. The full text of the license can be found at:

`https://creativecommons.org/licenses/by-sa/4.0/`

For alternative licensing options, source files, questions or comments please contact the author at `masterarbeit@jaseg.de`.

This is version `v0.7-0-gdc98ce4-dirty` generated on July 6, 2020. The git repository can be found at:

`https://git.jaseg.de/master-thesis.git`

Contents

1 **Introduction** **4**

2 **Fundamentals** **7**
- 2.1 Structure and operation of the electrical grid 7
  - 2.1.1 Structure of the electrical grid 7
  - 2.1.2 Operational concerns 11
- 2.2 Smart meter technology 12
  - 2.2.1 Smart metering and Human-Computer Interaction 14
  - 2.2.2 Common components 14
  - 2.2.3 Cryptographic coprocessors 16
  - 2.2.4 Physical structure and installation 16
- 2.3 Regulatory frameworks around the world 17
  - 2.3.1 International standards 17
  - 2.3.2 The regulatory situation in selected countries 17
  - 2.3.3 Common themes 20
- 2.4 Security in smart distribution grids 22
  - 2.4.1 Privacy in the smart grid 23
  - 2.4.2 Smart grid components as embedded devices 23
  - 2.4.3 The state of the art in embedded security 23
  - 2.4.4 Attack avenues in the smart grid 25
  - 2.4.5 Practical threats 27
  - 2.4.6 Conclusion or, why we are doomed 28

3 **Restoring endpoint safety in an age of smart devices** **30**
- 3.1 The theory of endpoint safety 31
  - 3.1.1 Attack characteristics 31
  - 3.1.2 Overall structural system security 32
  - 3.1.3 Complex microcontroller firmware 33
  - 3.1.4 Modern microcontroller hardware 33
  - 3.1.5 Safety vs. security: Opting for restoration instead of prevention 34
  - 3.1.6 Technical outline of a safety reset system 34
- 3.2 Communication channels on the grid 35
  - 3.2.1 Power line communication (PLC) systems and their use . 36
  - 3.2.2 Landline and wireless IP-based systems 36
  - 3.2.3 Short-range wireless systems 36
  - 3.2.4 Frequency modulation as a communication channel . . . 37

3.3 From grid frequency to a reliable communication channel 41
  3.3.1 Channel properties 41
  3.3.2 Modulation and its parameters 42
  3.3.3 Error-correcting codes 43
  3.3.4 Cryptographic security 44

4 Practical implementation 51
4.1 Data collection for channel validation 51
  4.1.1 Grid frequency estimation 51
  4.1.2 Frequency sensor hardware design 53
  4.1.3 Clock accuracy considerations 54
  4.1.4 Firmware implementation 55
  4.1.5 Frequency sensor measurement results 56
4.2 Channel simulation and parameter validation 57
  4.2.1 Sensitivity as a function of sequence length 64
  4.2.2 Sensitivity versus peak detection threshold factor 65
  4.2.3 Chip duration and bandwidth 66
4.3 Implementation of a demonstrator unit 67
  4.3.1 Selecting a smart meter for demonstration purposes . . . 67
  4.3.2 Firmware implementation 73
4.4 Grid frequency modulation emulation 73
4.5 Experimental results 74
4.6 Lessons learned 75

5 Future work 78
5.1 Precise grid characterization 78
5.2 Technical standardization 79
5.3 Regulatory adoption 79
5.4 Zones of trust 80

6 Conclusion 82

Bibliography 83

A Frequency sensor schematics 93

B Demonstrator firmware symbol size map 97

Chapter 1

Introduction

In the power grid, as in many other engineered systems, we can observe an ongoing diffusion of information systems into industrial control systems. Automation of these control systems has already been practiced for the better part of a century. Throughout the 20th century this automation was mostly limited to core components of the grid. Generators in power stations are computer-controlled according to electromechanical and economic models. Switching in substations is automated to allow for fast failure recovery. Human operators are still vital to these systems, but their tasks have shifted from pure operation to engineering, maintenance and surveillance[30, 4].

With the turn of the century came a large-scale trend in power systems to move from a model of centralized generation, built around massive large-scale fossil and nuclear power plants, towards a more heterogenous model of smaller-scale generators working together. In this new model large-scale fossil power plants still serve a major role, but two new factors come into play. One is the advance of renewable energies. The large-scale use of wind and solar power in particular from a current standpoint seems unavoidable for our continued existence on this planet. For the electrical grid these systems constitute a significant challenge. Fossil-fueled power plants can be controlled in a precise and quick way to match energy consumption. This tracking of consumption with production is vital to the stability of the grid. Renewable energies such as wind and solar power do not provide the same degree of controllability, and they introduce a larger degree of uncertainty due to the unpredictability of the forces of nature[30].

Along with this change in dynamic behavior, renewable energies have brought forth the advance of distributed generation. In distributed generation end-customers that previously only consumed energy have started to feed energy into the grid from small solar installations on their property. Distributed generation is a chance for customers to gain autonomy and shift from a purely passive role to being active participants of the electricity market[30].

To match this new landscape of decentralized generation and unpredictable renewable resources the utility industry has had to adapt itself in major ways. One aspect of this adaptation that is particularly visible to ordinary people is the computerization of end-user energy metering. Despite the widespread use of industrial control systems inside the electrical grid and the far-reaching diffusion of computers into people's everyday lives the energy meter has long been one of the last remnants of an offline, analog time. Until the 2010s many households were still served through electromechanical Ferraris-style meters that have their origin in the late 19th century[14, 121, 54]. Today under the umbrella term *Smart Metering* the shift towards fully computerized, often networked meters is well underway. The roll out of these *Smart Meters* has not been very smooth overall with some countries severely lagging behind. As a safety-critical technology, smart metering technology is usually standardized on a per-country basis. This leads to an inhomogenous landscape with–in some instances–wildly incompatible systems. Often vendors only serve a single country or have separate models of a meter for each country. This complex standardization landscape and market situation has led to a proliferation of highly complex, custom-coded microcontroller firmware. The complexity and scale of this–often network-connected–firmware makes for a ripe substrate for bugs to surface.

A remotely exploitable flaw inside a smart meter's firmware[1] could have consequences ranging from impaired billing functionality to an existential threat to grid stability[5, 4]. In a country where meters commonly include disconnect switches for purposes such as prepaid tariffs a coördinated attack could at worst cause widespread activation of grid safety systems by repeatedly connecting and disconnecting megawatts of load capacity in just the wrong moments[130].

Mitigation of these attacks through firmware security measures is unlikely to yield satisfactory results. The enormous complexity of smart meter firmware makes firmware security extremely labor-intensive. The diverse standardization landscape makes a coördinated, comprehensive response unlikely.

In this thesis, instead of focusing on the very hard task of improving firmware security we introduce a pragmatic solution to the–in our opinion likely–scenario of a large-scale compromise of smart meter firmware. In our proposal the components of the smart meter that are threatened by remote compromise are equipped with a physically separate *safety reset controller* that listens for a reset command transmitted through the electrical grid's frequency and on reception forcibly resets the smart meter's entire firmware to a known-good state. Our safety reset controller receives commands through Direct Sequence Spread Spectrum (DSSS) modulation carried out on grid frequency through a large controllable load such as an aluminum smelter. After forward error correction and cryptographic verification it re-flashes the meter's main microcontroller over the standard JTAG interface.

In this thesis, starting from a high level architecture we have carried out extensive simulations of our proposal's performance under real-world conditions. Based on these simulations we implemented an end-to-end prototype of our

---

[1] There are several smart metering architectures that ascribe different roles to the component called *smart meter*. Coarsely divided into two camps these are systems where all metering and communication functions reside within one physical unit and systems where metering and communication functions are separated into two units called the *smart meter* and the *smart meter gateway*[115]. An example for the former are setups in the USA, an example of the latter is the setup in Germany. For clarity, in this introductory chapter we use *smart meter* to describe the entire system at the customer premises including both the meter and a potential gateway.

proposed safety reset controller as part of a realistic smart meter demonstrator. Finally we experimentally validated our results and we will conclude with an outline of further steps towards a practical implementation.

Chapter 2

Fundamentals

2.1 Structure and operation of the electrical grid

Since this thesis is filed under *computer science* we will provide a very brief overview of some basic concepts of modern power grids.

2.1.1 Structure of the electrical grid

The electrical grid is composed of a large number of systems such as distribution systems, power stations and substations interconnected by long transmission lines. Mostly due to ohmic losses[1] the efficiency of transmission of electricity through long transmission lines increases with the square of voltage[29, 27]. In practice economic considerations take into account a reduction of the considerable transmission losses (about 6 % in case of Germany[34]) as well as the cost of equipment such as additional transformers and the cost increase for the increased voltage rating of components such as transmission lines. Overall these considerations have led to a hierarchical structure where large amounts of energy are transmitted over very long distances (up to thousands of kilometers) at very high voltages (upwards of 200 kV) and voltages get lower the closer one gets to end-customer premises. In Germany at the local level a substation will distribute 10 kV to 30 kV to large industrial consumers and small transformer substations which converting this to the 400 V three-phase AC households are usually hooked up with[29].

Transmission lines, bus bars and tie lines

The number one component of the electrical grid are transmission lines. Short transmission lines that tightly couple parts of a substation are called *bus bars*. Transmission lines that couple otherwise independent grid segments are called

---

[1] Power dissipation of a resistor of resistance $R[\Omega]$ given current $I[A]$ is $P_{\text{loss}}[W] = U_{\text{drop}} \cdot I = I^2 \cdot R$. Fixing power $P_{\text{transmitted}}[W] = U_{\text{line}} \cdot I$ this yields a dependency on line voltage $U_{\text{line}}[V]$ of $P_{\text{loss}} = \left(\frac{P_{\text{transmitted}}}{U_{\text{line}}}\right)^2 \cdot R$. Thus, ignoring other losses a 2× increase in transmission voltage halves current and cuts ohmic losses to a quarter. In practice the economics are much more complicated due to the cost of better insulation for higher-voltage parts and the cost of power factor compensation.

tie lines. A tie line often connects grid segments operated by two different operators e.g. across a country border.

In mathematical analysis *short* transmission lines can be approximated as a simple lumped-component RLC[2] circuit. In longer lines the effect of wave propagation along the line has to be taken into consideration. In the lumped model the transmission line is represented by a circuit of one or two inductors, one or two capacitors and some resistors. This representation simplifies analysis. For *long* transmission lines above 50 km (cable) or 250 km (overhead lines) this approximation breaks down and wave propagation along the line's length has to be taken into account. The resulting model is what RF engineering calls a transmission line and models the line's parasitics[3] as being uniformly distributed along the length of the line. To approximate this model in lumped-element evaluations the line is represented as a long chain of small lumped-component RLC sections. This complex structure makes simulation and analysis more difficult in comparison to short lines[29].

Almost all transmission lines used in the transmission and distribution grid use three-phase alternating current (AC). Long-distance overland lines are usually implemented as overhead lines due to their low cost and ease of maintenance. Underground cables are much more expensive because of their insulation and are only used when overhead lines cannot be used for reasons such as safety or aesthetics. In specialized applications such as long, high-power undersea cables high-voltage DC (HVDC) is used. In HVDC converter stations at both ends of the line convert between three-phase AC and the line's DC voltage. These converter stations are controlled electronically and do not exhibit any of the mechanical inertia that is characteristic for rotating generators in a power plant. Since HVDC re-synthesizes three-phase AC from DC at the receiving end of the line it can be used to couple non-synchronous grids. This allows for additional degrees of control over the transmission of power compared to a regular transmission line. These technical benefits are offset by high initial cost (mostly due to the converter stations) leading to HVDC being used in specific situations only[30].

Generators

Traditionally all generators in the power grid were synchronous machines. A synchronous machine is a generator whose copper coils are wound and connected in such a way that during normal operation its rotation is synchronous with the grid frequency. Grid frequency and generator rotation speed are bidirectionally electromechanically coupled. If a generator's angle of rotation would lag behind the grid it would receive electrical energy from the grid and convert it into mechanical energy, acting as a motor–When the machine leads it acts as a generator and is braked. Small deviations between rotational speed and grid frequency will be absorbed by the electromechanical coupling between both. Maintaining optimal synchronization over time is the task of complex control systems inside power stations' speed governors[27, 29].

---

[2] Resistor-inductor-capacitor.

[3] Stray capacitance, ohmic resistance and stray inductance.

Nowadays besides traditional rotating generators the grid also contains a large amount of electronically controlled inverters. These inverters are used in photovoltaic installations and other setups where either DC or non-synchronous AC is to be fed into the grid. Setups like these behave differently to rotating generators. In particular *inertia* in these setups is either absent or a software parameter. This potentially reduces their overload capacity compared to rotating generators. The fundamentally different nature of electronically controlled inverters has to be taken into account in planning and regulation[30].

Switchgear

In the electrical grid switches perform various roles. The ones a computer scientist would recognize are used for routing electricity between transmission lines and transformers and can be classified into ones that can be switched under load (called load switches) and ones that can not (called disconnectors). The latter are used to ensure parts of the network are free from voltage e.g. during maintenance. The former are used to re-route flows of electrical currents. A major difference in their construction is that in contrast to disconnectors load switches have built-in components that extinguish the high-power arc discharge that forms when the circuit is interrupted under load[4]. Beyond this there are circuit breakers. Circuit breakers are safety devices that even under failure conditions can still switch at several times the circuit's nominal current. They are activated automatically on conditions such as overcurrent or overvoltage. Finally, fuses can be considered non-resettable switches. The fuse in a computer power supply is barely more than a glass tube with some wire in it that is designed to melt at the designated current. In energy systems fuses are often much more complex devices that in some cases utilize explosives to quickly and decisively open the circuit and extinguish the resulting arc discharge[98, 29, 27].

Transformers

Along with transmission lines transformers are one of the main components most people will be thinking of when talking about the electrical grid. Transformers connect grid segments at different voltage levels with one another. In the distribution grid transformers are used to provide standard end-user voltage levels to the customer (e.g. 230/400V in Europe) from a 10 kV to 25 kV feeder. In places that use overhead wiring to connect customer households this is the role of the pole-mounted gray devices the size of a small refrigerator that are characteristic for these systems. Transformers can also be used to convert between buses without a fourth neutral conductor and buses with one.

Transformers are large and heavy devices consisting of thick copper wire or copper foil windings arranged around a core made from thin stacked, insulated iron sheets. The entire core sits within a large metal enclosure that is filled with liquid (usually a specialized oil) for both cooling and electrical insulation. This cooling liquid is cooled by radiator fins on the transformer enclosure itself

---

[4] While an arc discharge is considered a fault condition in most low-voltage systems including computers, in energy systems it is often part of normal operation.

or an external heat exchanger. Depending on the design cooling may rely on natural convection within the cooling liquid or on electrical pumps[29, 27].

Transformers come in a large variety of coil and wiring configurations. There exist autotransformers where the secondary is part of the primary (or vice-versa) that are used to translate between voltage levels without galvanic isolation at lower cost. Transformers used in parts of the electrical grid often have several taps and include *tap changers*. A tap changer is a system of mechanical switches that can be used to switch between several discrete transformer ratios to adjust secondary voltage under load[27]. Tap changers are used in the distribution grid to maintain the specified voltage tolerances at the customer's connection.

Instrument transformers

While operating on the exact same physical principles instrument transformers are very different from regular transformers in an energy system. Instrument transformers are specialized low-power transformers that are used as transducers to measure voltage or current at very high voltages. They are part of the control and protection systems of substations[29].

Chokes

Chokes are large inductors. In power grid applications their construction is similar to the construction of a transformer with the exception that they only have a single winding on the core. They are used for a variety of purposes. A frequent use is as a series inductor on one of the phases or the neutral connection to limit transient fault currents. In addition to this inductors are also used to tune LC circuits. One such use are Petersen coils, large inductors in series with the earth connection at a transformer's star point that are used to quickly extinguish arcs between phase and ground on a transmission line. The Petersen coil forms a parrallel LC resonant circuit with the transmission line's earth capacitance. Tuning this circuit through adjusting the Petersen coil reduces earth fault current to a level low enough to quickly extinguish the arc[27].

Power factor correction

Power factor is a power engineering term that is used to describe how close the current waveform of a load is to that of a purely resistive load. Given sinusoidal input voltage $V(t) = V_{\text{pk}} \sin(\omega_{\text{nom}} t)$ with $\omega_{\text{nom}} = 2\pi f_{\text{nom}} = 2\pi \cdot 50\,\text{Hz}$ being the nominal angular frequency, the current waveform of a resistor with resistance $R\,[\Omega]$ according to Ohm's law would be $I(t) = \frac{V(t)}{R} = \frac{1}{R} V_{\text{pk}} \sin(\omega_{\text{nom}} t)$. In this case voltage and current are perfectly in phase, i.e. the current at time t is linear in voltage at constant factor $\frac{1}{R}$.

In contrast to this idealized scenario reality provides us with two common issues: One, the load may be reactive. This means its current waveform is an ideal sinusoid, but there is a phase difference between mains voltage and load current like so: $I(t) = \frac{V(t)}{R} = \frac{1}{|Z|} V_{\text{pk}} \sin(\omega_{\text{nom}} t + \varphi)$. Z is the load's complex impedance combining inductive, capacitive and resistive components and φ is the phase difference between the resulting current waveform and the mains voltage waveform. Examples of such loads are motors and the inductive ballasts in old fluorescent lighting fixtures.

The second potential issue are loads with a non-sinusoidal current waveform. There are many classes of these but the most common one are the switching-mode power supplies (SMPS) used in most modern electronic devices.. Most SMPS have an input stage consisting of a bridge rectifier followed by a capacitor that provide high-voltage DC power to the following switch-mode convert circuit. This rectifier-capacitor input stage under normal load draws a high current only at the very peak of the input voltage sinusoid and draws almost zero current for most of the period.

These two cases are measured by *displacement power factor* and *distortion power factor* that when combined yield the overall true power factor. The power factor is a key quantity in the design and operation of the power grid. As a variable in the operation of electrical grids it is also referred to as VAR after its is unit Volt-Ampère Reactive. A high power factor (close to 1.0, i.e. an in-phase sinusoidal current waveform) yields lowest transmission and generation losses. If reactive power generation and consumption are mismatched and power factor is low, high currents develop that lead to high transmission losses. For this reason grids include circuits to compensate reactive power imbalances[29]. These circuits can be as simple as inductors or capacitors connected to a power line but often can be switched to adapt to changing load conditions. Static var compensators are particularly fast-acting reactive power compensation devices whose purpose is to maintain a constant bus voltage[108].

Loads

Lastly, there is the loads that the electrical grid serves. Loads range from mains-powered indicator lights in devices such as light switches or power strips weighing in at mere Milliwatts to large smelters in industrial metal production that can consume a fraction of a gigawatt all on their own.

2.1.2 Operational concerns

Modelling the electrical grid

Modelling performs an important role in the engineering of a reliable power infrastructure. The grid is a complex, highly dynamic system. To maintain operational parameters such as voltage, grid frequency and currents inside their specified ranges complex control systems are necessary. To design and parametrize such control systems simulations are a valuable tool. Using model calculations the effects of control systems on operational variables such as transmission efficiency or generation losses can be estimated. Model simulations can be used to identify structural issues such as potential points of congestion. The same models can then be used to engineer solutions to such issues, e.g. by simulating the effect of a new transmission line.

There are several aspects under which the grid or parts of the grid can be simulated. There are static analysis methods such as modal analysis that yield information on problematic electromechanical oscillations by computing the eigenvalues of a large system of differential equations describing the collective behavior of all components of the grid. Modal analysis is one example of simulations used in grid planning. Modal analysis is used in decisions to install additional stabilization systems in a particular location. In contrast to static analysis, transient simulations calculate an approximation of the time-domain behavior of some variable of interest under a given model. Transient simulations are used e.g. in the design of control systems. Finally, power flow equations describe the flow of electrical energy throughout the network from generator to load. Numerical solutions these equations are used to optimize control parameters to increase overall efficiency.

2.2 Smart meter technology

Smart meters were a concept pushed by utility companies throughout the early 21st century. Smart metering is one component of the larger societal shift towards digitally interconnected technology. Old analog meters required that service personnel physically come to read the meter. *Smart* meters automatically transmit their readings through modern technologies. Utility companies were very interested in this move not only because of the cost savings for meter reading personnel: An always-connected meter also allows several entirely new use cases that have not been possible before. One often-cited one is utilizing the new high-resolution load data to improve load forecasting to allow for greater generation efficiency. Computerizing the meter also allows for new fee models where electricity cost is no longer fixed over time but adapts to market conditions. Models such as prepayment electricity plans where the customer is automatically disconnected until they pay their bill are significantly aided by a fully electronic system that can be controlled and monitored remotely[4]. A remotely controllable disconnect switch can also be used to coerce customers in situations where that was not previously economically possible[5]. Figure 2.1 shows a schema of a smart metering installation in a typical household[115].

To the customer the utility of a smart meter is largely limited to the convenience of being able to read it without going to their basement. In the long term it is said that there will be second-order savings to the customer since electricity prices adapting to the market situation along with this convenience will lead them to consume less electricity and to consume it in a way that is more amenable to utilities, both leading to reduced cost[14, 24, 4].

Traditional Ferraris counters with their distinctive rotating aluminum disc are simple electromechanical devices. Since they do not include any semiconductors or other high technology that might be prone to failure a cheap Ferraris-style meter can last decades. In contrast to this, smart meters are complex high technology. They are vastly more expensive to develop in the first place since they require the development and integration of large amounts

---

[5] The Swiss association of electrical utility companies in Section 7.2 Paragraph (2)a of their 2010 white paper on the introduction of smart metering[15] cynically writes that remotely controllable disconnect switches "lead a new tenant to swiftly register" with the utility company. This white paper completely vanished from their website some time after publication, but the internet archive has a copy.

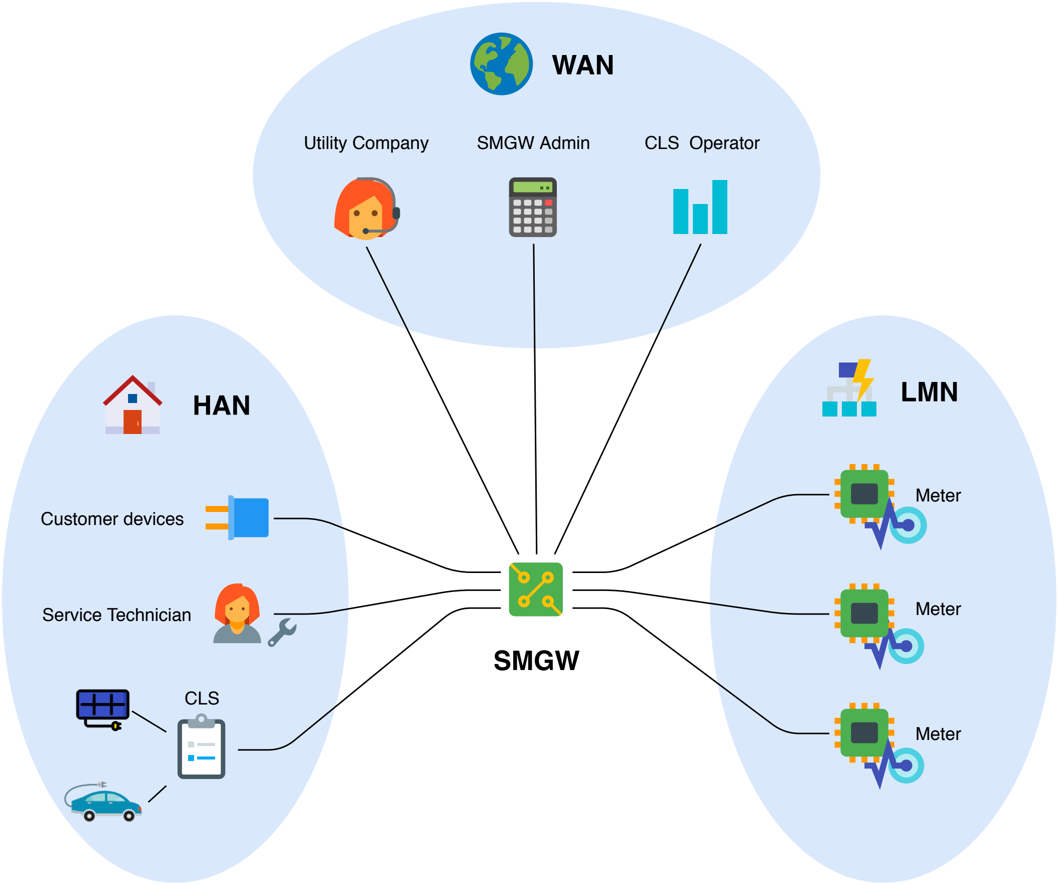


Figure 2.1: A typical usage scenario of a smart metering system in a typical home. This diagram shows a gateway connected to multiple smart meters through its local metrological network (LMN) and a multitude of devices on the customer's home area network (HAN). A solar inverter and an electric car are connected through a controllable local systems (CLS) adaptor.

of complex, custom firmware. Once deployed, their lifetime is limited by this complexity. Complex semiconductor devices tend to fail, and firmware that needs to communicate with the outside world tends to not age well[12]. This combination of higher unit cost and lower expected lifetime leads to increased costs per household. This cost is usually shared between utility and customer.

As part of its smart metering rollout the German government in 2013 had a study conducted on the economies of smart meter installations. This study came to the conclusion that for the majority of households computerizing an existing Ferraris meter is uneconomical. For larger consumers or new installations the higher cost of installation over time is expected to be offset by the resulting savings in electricity cost[24].

2.2.1 Smart metering and Human-Computer Interaction

A fundamental aspect in realizing many of the cost and energy savings promised by the smart metering revolution is that it requires a paradigm shift in consumer interaction. Previously most consumers would only confront their energy use when they receive their monthly or yearly electricity bill. A large part of the cost savings smart meters promise over traditional metering infrastructure[6] critically depend on the consumer regularly interacting with the meter through an in-home display or app, then changing their behavior. We live in an era where our attention is already highly contested. A myriad of apps and platforms compete for our attention through our smart phones and other devices. Introducing an entirely new service exerting cognitive pressure into this already complex battleground is a large endeavour. On the one hand it is not clear how this new service would compete with everything else. On the other hand if it does manage to capture our attention and lead us to modify our behavior, what are the side effects? For instance an in-home display might increase financial anxiety in economically disadvantaged customers.

Human Computer Interaction research has touched the topic of smart metering several times and has many insights to offer for technologists[105, 107, 87, 28, 47]. An issue pointed out in [107] is that at least in some countries consumers fundamentally distrust their utility companies. This trust issue is exacerbated by smart meters being unilaterally forced onto consumers by utility companies. Much of the success of smart metering's ubiquitous promises of energy savings depends on consumer coöperation. Here, the aforementioned trust issue calls into question smart metering's chances of long-term success.

As [105] pointed out smart metering developments could benefit greatly from early involvement of HCI research. A systematic analysis of non-technical aspects can prevent issues such as privacy implications initially being overlooked in the dutch deployment[31]. It is not clear that current standardization practice encompasses an in-depth consideration of the role of consumers in the socio-technological environment posed by this new technology. Standardization is often narrowly focused on technological aspects with little input beyond the occassional public consultation at the time the new standards are being implemented into law. This corporate-driven approach to technological progress being forced through national standardization bears a risk of failing to meet its advertised consumer benefits.

2.2.2 Common components

Smart meters usually are built around an off-the-shelf microcontroller (microcontroller unit, MCU). Some meters use specialized smart metering system-on-chips (SoCs)[36] while others use standard microcontrollers with core metering functions implemented in external circuitry (cf. Section 4.3.1 where we detail the meter in our demonstration setup). Specialized SoCs usually contain a segment LCD driver along with some high-resolution analog-to-digital converters for

---

[6] We are excluding savings from Demand-Side Response (DSR) implemented through smart meters here: Traditional ripple control systems already allowed for these[40], and due to the added cost of high-power relays many smart meters do not include such features.

the actual measurement functions. In many smart meter designs the metering SoC is connected to another full-featured SoC acting as the modem. At a casual glance this might seem to be a security measure, but it is be more likely that this is done to ease integration of one metering platform with several different communication stacks (e.g. proprietary sub-gigahertz wireless, power line communication (PLC) or Ethernet). In these architectures there is a clear line of functional demarcation between the metering SoC and the modem. As evidenced by over-the-air software update functionality (see e.g. [65]) this does not however extend to an actual security boundary.

Energy usage is calculated by measuring both voltage and current at high resolution and then integrating the measurements. Current measurements are usually made with either a current transformer or a shunt in a four-wire configuration. Voltage is measured by dividing input AC down with a resistor chain. Both are integrated digitally using the MCU's time base as a reference.

Whereas legacy electromechanical energy meters only provided a display of aggregate energy use through a decimal counter as well as an indirect indication of power through a rotating wheel one of the selling points of smart meters is their ability to calculate advanced statistics on energy use. These statistics are supposed to help customers better target energy conservation measures[24].

Smart meters can perform additional functions in addition to pure measurement and data aggregation. One is to serve as a gateway between the utility company's control systems and large controllable loads in the consumer's household for Demand-Side Management (DSM)[14]. In DSM the utility company can control when exactly a high-power device such as a water storage heater is switched on. To the customer the precise timing does not matter since the storage heater is set so that it has enough hot water in its reservoir at all times. The utility company however can use this degree of control to reduce load variations during peak times. The efficiency gains realized with this system translate into lower electricity prices for DSM-enabled loads for the customer. Traditionally DSM was realized on a local level using ripple control systems. In ripple control control data is coded by modulating a carrier at a low frequency such as 400 Hz on top of the regular mains voltage. These systems require high-power transmitters at tens of kilowatts and still can only bridge regional distances[40].

Another important additional function is that some smart meters can be used to remotely disconnect consumer households with outstanding bills. Using euphemisms such as *utility revenue protection*[70] or *reducing nontechnical losses*[16] while cynically claiming *Consumer Empowerment*[70] these systems allow an utility company to remotely disconnect a customer at any time[5]. Whereas before smart metering this required either additional hardware or an expensive site visit by a qualified technician smart meters have ushered in an era of frictionless control[7].

---

[7] Note that in some countries such as the UK non-networked mechanical prepayment meters did exist. In such systems the user inserts coins into a coin slot that activates a disconnect switch at the household's main electricity connection. These systems were non-networked and did not allow for remote control. A disadvantage of such systems compared to modern *smart* systems are the high cost of the coin acceptor and the overhead of site visits required to empty the coin box[4].

2.2.3 Cryptographic coprocessors

Just like in legacy electricity meters in smart meters physical security is still a key component of the overall system design. Since in both types of meter cost depends on physical quantities being measured at the customer premises customers can save cost in case they are able to falsify the meter's measurements without being detected[4]. For this reason both types of meters employ countermeasures against physical intrusion. Compared to high-risk devices such as card payment processing terminals or ATMs the tamper proofing used in smart meters is only basic[4]. Common measures include sealing the case by irreversibly ultrasonically welding the front and back plastic shells together or the use of security seals on the lid covering the input and output screw terminals. The common low-tech attack of using magnets to saturate the current transformer's ferrite cores is detected using hall sensors[4, 3, 67, 60, 41]. German smart metering standards specify the use of a smartcard-like security module to provide transport encryption and other cryptographic services[21, 20]. During our literature review we did not find many references to similar requirements in other national standards, though this does not mean that individual manufacturers do not use smartcards for engineering reasons or due to pressure from utilities. The limited documentation on meter internals that we did find such as [36, 10, 68] suggests where no such regulation exists manufacturers and utilities likely choose to forego such advanced measures and instead settle on simple software implementations.

2.2.4 Physical structure and installation

Smart meters are installed like traditional electricity meters. In Japan this means they are usually installed on an exterior wall and need to be resistant against weather and extreme environmental conditions (direct sunlight, high temperature, high humidity). In Germany the meter is always installed either indoors or in an outdoor utility closet that is sealed to keep out the weather. In most countries the meter is connected through large integrated screw terminals. In the US meters compliant with the domestic ANSI C12 standard are round and plug into a large socket that is wired into the house or apartment's electrical connection.

Modern smart meters are usually made with plastic cases. Ferraris meters often used cases stamped from sheet metal with glass windows on them. Smart meters now look much more like other modern electronic devices. A common construction style is to separate the case into front and back halves with both clipped or ultrasonically welded together. Ultrasonic welding gives a robust, airtight interface that cannot easily be separated and reconnected without leaving visible traces, which helps with tamper evidence properties. As an industry-standard process common in various consumer goods ultrasonic welding is a cheap and accessible technology[41, 36].

Communication interfaces sometimes are brought out through regular electromechanical connectors but often also are optical interfaces. A popular style here is to use a regular UART connected to an LED/phototransistor optocoupler mounted on the side of the case. The user interface is usually limited to an LCD display. For cost and ingress protection smart meters rarely use mechanical buttons. Some smart meters use a phototransistor mounted behind the faceplate that can be activated with a flashlight as a crude contact-less input device[41].

All meters provide several options for security seals to be installed to detect opening of the meter or access to its terminal block. The shape and type of these security seals varies. Factory-installed seals are used to detect tampering of the meter itself while seals made by the utility during meter installation are used to guard the meter's terminal block and detect attempts at by-passing[32].

2.3 Regulatory frameworks around the world

Smart metering regulation varies from country to country as it is tightly coupled to the overall regulation of the electrical grid. The standardization of the physical form factor and metrological parameters of a meter is usually separate from the standardization of its *smart* functionality. Most countries base the standard for their meters' outwards-facing communication interface on a family of standards unified under the IEC as DLMS/COSEM. Employing this base protocol ountry-specific standardization only covers which precise variant of it is spoken and what features are supported.

2.3.1 International standards

The family of standards one encounters most in smart metering applications are IEC 62056 specifying the Device Language Message Specification (DLMS) and the Companion Specification for Electronic Metering (COSEM). DLMS/-COSEM are application-layer standards describing a request/response schema similar to HTTP. DLMS/COSEM are mapped onto a multitude of wire protocols. They can be spoken over TCP/IP or mapped onto low-speed UART serial interfaces [110, 115]. Besides DLMS/COSEM there are a multitude of standards usually specifying how DLMS/COSEM are to be applied.

DLMS/COSEM show some amount of feature creep. They do not adhere to the age-old systems design adage that a tool should *do one thing and do it well*. Instead they try to capture the convex hull of all possible applications. This led to a complicated design that requires extensive additional specification and testing to maintain interoperability. In particular in the area of transport security it becomes evident that the IEC as an electrical engineering standards body stretched their area of expertise where resorting to established standard protocols would have led to a better outcome[128]. Compared to industry-standard transport security the IEC standards provide a simplistic key management framework based on a static shared key with unlimited lifetime and provide sub-optimal transport security properties (e.g. lack of forward-secrecy)[74, 110].

2.3.2 The regulatory situation in selected countries

In this section we will give an overview of the situation in a number of countries. This list of countries is not representative and notably does not include any developing countries and is geographically biased. We selected these countries for illustration only and based our selection in a large part on the availability of information in a language we can read. We will conclude this section with a summary of common themes.

Germany

Germany standardized smart metering on a national level. Apart from the calibration standards applying to any type of meter smart meters are covered by a set of communications and security standards developed by the German Federal Office for Information Security (BSI). Germany mandates smart meter installations for newly constructed buildings and during major renovations but does not require most legacy residential installations to be upgraded. This is a consequence of a 2013 cost-benefit analysis that found these upgrades to be uneconomical for the majority of residential customers[24, 22, 59, 16].

The German standards strictly separate between metering and communication functions. Both are split into separate devices, the *meter* and the *gateway* (called *smart meter gateway* in full and often abbreviated *SMGW*). One or several meters connect to a gateway through a COSEM-derived protocol. The communication interface between meter and gateway can optionally be physically unidirectional. An unidirectional interface eliminates any possibility of meter firmware compromise. The gateway contains a cryptographic security module similar to a smartcard[89] that is entrusted with signing of measurements and maintaining an authenticated and encrypted communication channel with its authorities. Security of the system is certified according to a Common Criteria process.

The German specification does not include any support for disconnect switches as they are common in some other countries outside of demand-side management. It only does not prohibit the installation of one behind the smart meter installation. This makes it theoretically possible for a utility company to still install a disconnect switch to disconnect a customer, but this would be a spearate installation from the smart meter. In Germany there are significant barriers that have to be met before a utility company may cut power to a household[114]. The elision of a disconnect switch means attacks on German meters will be limited in influence to billing irregularities and attacks using DSM equipment such as water storage heaters that represent only a fraction of overall load.

The Netherlands

The Netherlands were early to take initiative to roll out smart metering after its recognition by the European Commission in 2006[31, 118]. After overcoming political issuses the Netherlands were above the European median in 2018, having replaced almost half of all meters[31, 113]. Dutch smart meters are standardized by a consortium of distribution system operators. They integrate gateway and metrology functions into one device. The utility-facing interface is a IEC DLMS/COSEM-based interface over cellular radio such as GPRS or LTE[7]. Like e.g. the German standard, the Dutch standard precisely specifies all communication interfaces of the meter[39]. Another parallel is that the Dutch standard also does not cover any functionality for remotely disconnecting a household. This absence of a disconnect switch limits attacks on Dutch smart meters, too to causing billing irregularities.

The UK

The UK is currently undergoing a smart metering rollout. Meters in the UK are nationally standardized to provide both Zigbee ZSE-based and IEC DLMS/COSEM connectivity. UK smart metering specifications are shared between electrical and gas meters. Different to other countries' specifications the UK national specifications require electrical meters to have an integrated disconnect switch and gas meters to have an integrated valve. In Northern Ireland most consumers use prepaid electricity contracts[4]. Prepayment and credit functionality are also specified in the UK's national smart metering standard, as is remote firmware update functionality[124]. Outside communications in these standards is performed through a gateway (there called *communications hub*) that can be shared between several meters [123, 124, 122, 16, 110]. The combination of both gas and electricity metering into one family of standards and the exceptionally large set of *required* features make the UK regulations the maximalist option among the regulations in this section. The mandatory inclusion of both disconnect switches and remote connectivity up to remote firmware update make it an interesting attack target[5].

Italy

Italy was among the first countries to legally mandate the widespread installation of smart meters in households. Italy in 2006 and 2007 by law set a starting date for the rollout in 2008[16]. The Italian electricity market was recently privatized. While the wholesale market and transmission network privatization has advanced the vast majority of retail customers continued to use the incumbent distribution system operator ENEL as their supplier[113]. This dominant position allowed ENEL to orchestrate the large-scale rollout of smart meters in Italy. Almost every meter in Italy had been replaced by a smart meter by 2018[113]. An unique feature of the Italian smart metering infrastructure is that it relies on Power Line Communication (PLC) to bridge distances between meters and cellular radio gateways[58].

Japan

Japan is currently rolling out smart metering infrastructure. Compared to other countries in Japan significant standardization effort has been spent on smart home integration[2, 110, 16]. Japan has domestic standards under its Japanese Industrial Standards organization (JIS) that determine metrology and physical dimensions. Tokyo utility company TEPCO is currently rolling out a deployment that is based on the IEC DLMS/COSEM standards suite for remote meter reading in conjuction with the Japanese ECHONET home-area network protocol. Smart meters are connected to TEPCO's backend systems through the customer's internet connection, sub-gigahertz radio based on 802.15.4 framing, regular landline internet or PLC[66, 110].

A unique point in the Japanese utility metering landscape is that the current practice is monthly manual readings. In Japan residential utility meters are usually mounted outside the building on an exterior wall and every month someone with a mirror on a long stick will come and read the meter. The meter reader then makes a thermal paper print-out of the updated utility bill and puts it into the resident's post box. This practice gives consumers good control over their consumption but does incur significant personnel overhead.

The USA

In the USA the rollout of smart meters has been promoted by law as early as 2005. The US electricity market is highly complex with states having significant authority to decide on their own policies[16]. Originally different from the IEC standards used in large fraction of the rest of the world the USA developed their own domestic set of standards for smart meters under the Americal National Standards Institute (ANSI)[110]. Today ANSI is converging with the IEC on the protcol layer. An obvious feature of ANSI-standard meters is that they are round and plug into a wall-mounted socket while IEC devices are usually rectangular and connected directly to the mains wiring through large screw terminals[36].

2.3.3 Common themes

Researching the current situation around the world for the above sections we were able to distill some common themes. First, smart metering is slowly advancing on a global scale and despite significant reservations from privacy-conscious people and consumer advocates it seems it is here to stay. Still, there are some notable exceptions of countries that have decided to scale-back an ongoing rollout effort after subsequent analysis showed economical or other issues[8].

The introduction of smart metering

The smart meter rollout is largely driven by utility companies. Utility companies field a variety of arguments for the rollout. The most prominent argument is a general increase in energy-efficiency along with a reduction of emissions. This argument is based on the estimation that smart metering will increase private customers' awareness of their own consumption and this will lead them to reduce their consumption. The second highly popular argument for smart metering is that it is necessary for the widespread adoption of renewable energies. This argument again builds on the trend towards green energy to rationalize smart metering. Interestingly this argument is often formulated as an inevitability instead of a choice.

Academic reception of smart metering is dyed with an almost unanimous enthusiasm. In particular smart meter communication infrastructure has received

---

[8] cf. the Netherlands and Germany

a large amount of research attention[40, 58, 69, 86, 90, 131, 5, 4]. Outside of human-computer interaction claims that smart meters will reduce customer energy consumption have often been uncritically accepted.

Standardization and reality of smart devices

Regulators, utilities and academics meet in their enthusiasm on the issue of smart home integration of smart metering. A feature of many concepts is that the meter acts as the centerpiece of a modern, fully integrated smart home[7, 52, 19, 1]. The smart meter serves as a communication hub between a new class of grid-aware loads and the utility company's control center. Large (usually thermal) loads such as dishwashers, refrigerators and air conditioners are expected to intelligently adapt their heating/cooling cycles to better match the grid's supply. A frequent scenario is one in which the meter bills the customer using near-real time pricing, and supplies large loads in the customer's household with this pricing information. These loads then intelligently schedule their operation to minimize cost[110]. At the time between 2000 and 2005 when smart metering proposals were first advanced this vision might have been an effect of the *law of the instrument*[72, 4]. Back then outside of specialty applications household devices were not usually networked[92]. Smart meters at the time may have seemed to be the obvious choice for a smart home communications hub.

From today's perspective, this idea is obviously outdated. Smart *things* now have found their way into many homes. Only these things are directly interconnected through the internet–foregoing the home-area network (HAN) technologies anticipated by smart metering pioneers. The simple reason for this is that nowadays anyone has Wifi, and Wifi transceivers have become inexpensive enough to disappear in the bill of materials (BOM) cost of a large home device such as a washing machine. Smart meters are usually situated in the basement–physically far away from most of one's devices. This makes connecting them to said devices awkward and connecting them via the local Wifi lends the question why the smart devices should not simply use the internet directly.

Connecting things to a smart meter through a local bus is academically appealing. It promises cost-savings from a simpler physical layer (such as ZigBee instead of Wifi) and it neatly separates concerns into home infrastructure and the regular internet. Communication between smart meter and devices never leaves the house. This promises tolerance to utility backend systems breaking. It also physically keeps communication inside the house, bypassing the utility's eyes improving both customer privacy and agency. The presently popular model of a device as simple as a light bulb proxying its every action through a manufacturer's servers somewhere on the public internet is in stark contrast to this scenario. Alas, the reason that this model is as popular is that in most cases it simply works. Device manufacturers integrate one of many off-the-shelf Wifi modules. The resulting device will work anywhere on earth[9]. A HAN-connected device would have several variants with different modems for different standards.

---

[9] For some places channel assignments may have to be updated. This is a configuration-level change and in some devices can be done by the end-user during provisioning.

Some might work across countries, but some might not. And in some countries there might not even be a standard for smart grid HANs.

Looking at the situation like this begs the question why this realization has not yet found its way into mainstream acceptance by smart metering implementors. The customer-facing functionality promised through smart meters would be simple to implement as part of a now-standard *Internet of Things* application. An in-home display that shows real time energy consumption and cost statistics would simply be an Android tablet fetching summarized data from the utility's billing backend. Custom hardware for this purposes seems anachronistic today. Demand-side response by large loads would be as simple as an HTTPS request with a token identifying the customer's contract that returns the electricity price the meter is currently charging along with a recommendation to switch on or off. It seems the smart home has already arrived while smart metering is still getting off the starting blocks[4].

2.4 Security in smart distribution grids

The smart grid in practice is nothing more or less than an aggregation of embedded control and measurement devices that are part of a large control system. This implies that all the same security concerns that apply to embedded systems in general also apply to most components of a smart grid. Where programmers have been struggling for decades now with input validation[85], the same potential issue raises security concerns in smart grid scenarios as well[93, 84]. Only, in smart grid we have two complicating factors present: Many components are embedded systems, and as such inherently hard to update. Also, the smart grid and its control algorithms act as a large (partially-)distributed system making problems such as input validation or authentication harder[11] and adding a host of distributed systems problems on top[81].

Given that the electrical grid is essential infrastructure in our modern civilization, these problems amount to significant issues in practice. Attacks on the electrical grid may have grave consequences[5, 84] while the long maintenance cycles of various components make the system slow to adapt. Thus, components for the smart grid need to be built to a much higher standard of security than most consumer devices to ensure they live up to well-funded attackers even decades down the road. This requirement intensifies the challenges of embedded security and distributed systems security among others that are inherent in any modern complex technological system. The safety-critical nature of the modern smart metering ecosystem in particular was quickly recognized by security experts[5].

A point we will not consider in much depth in this work is theft of electricity. An incentive for the introduction of smart metering that is frequently cited in utility industry publications outside of a general public's view is the reduction of electricity theft[32]. Academic publications tend to either focus on other benefits such as generation efficiency gains through better forecasting or rationalize the consumer-unfriendly aspects of smart metering with "enormous social benefits"[101]. They do not usually point out the economical incentive such *revenue protection* mechanisms provide[5, 4].

2.4.1 Privacy in the smart grid

A serious issue in smart metering setups is customer privacy. Even though the meter “only” collects aggregate energy consumption of a whole household this data is highly sensitive[95]. This counterintuitive fact was initially overlooked in smart meter deployments leading to outrage, delays and reduced features[31]. The root cause of this problem is that given sufficient timing resolution these aggregate measurements contain ample entropy. Through disaggregation algorithms individual loads can be identified and through pattern matching even complex usage patterns can be discerned with alarming accuracy[57]. Similar privacy issues arise in many other areas of modern life through pervasive tracking and surveillance[133]. What makes the case of smart metering worse is that even the fig leaf of consent such practices often hide behind does not apply here. If a citizen does not consent to Google’s privacy policy Google says they can choose not to use their service. In today’s world this may not be a free choice thereby invalidating this argument but it is at least technically possible. Smart metering on the other hand is mandated by law and depending on the law a customer unwilling to accept the accompanying privacy violation may not be able to evade it[23].

2.4.2 Smart grid components as embedded devices

A fundamental challenge in smart grid implementations is the central role smart electricity meters play. Smart meters are used both for highly-granular load measurement and (in some countries) load switching[132]. Smart electricity meters are effectively consumer devices. They are built down to a certain price point that is measured by the burden it puts on consumers. The cost of a smart meter is ultimately limited by it being a major factor in the economies of a smart meter rollout[24]. Cost requirements preclude some hardware features such as the use of a standard hardened software environment on a high powered embedded system (such as a hypervirtualized embedded linux setup) that would both increase resilience against attacks and simplify updates. Combined with the small market sizes in smart grid deploymentsthis results in a high cost pressure on the software development process for smart electricity meters. Most vendors of smart electricity meters only serve a handful of markets. A large fraction of smart meter development cost lies in the meter’s software. Landis+Gyr, a large manufacturer that makes most of its revenue from utility meters in their 2019 annual report write that they 36 % of their total R&D budget on embedded software (firmware) while spending only 24 % on hardware R&D[82, 83]. There exist multiple competing standards applicable to various parts of a smart electricity meter and most countries have their own certification regimen[117]. This complexity creates a large development burden for new market entrants[125].

2.4.3 The state of the art in embedded security

Embedded software security generally is much harder than security of higher-level systems. This is due to a combination of the unique constraints of embedded devices: Among others they are hard to update and usually produced in small quantities. They also lack capabilities compared to full computers. Processing power is limited and memory protection functions are spartan. Even well-funded companies continue to have trouble securing their embedded systems. A spectacular example of this difficulty is the recently-exposed flaw in Apple's iPhone SoC first-stage ROM bootloader[10], that allows a full compromise of any iPhone before the iPhone X. iPhone 8, one of the affected models, was still being manufactured and sold by Apple until April 2020. In another instance in 2016 researchers found multiple flaws in the secure-world firmware used by Samsung in their mobile phone SoCs. The flaws they found were both severe architectural flaws such as secret user input being passed through untrusted userspace processes without any protection and shocking cryptographic flaws such as CVE-2016-1919[11][71]. And Samsung is not the only large multinational corporation having trouble securing their secure world firmware implementation. In 2014 researchers found an embarrassing integer overflow flaw in the low-level code handling untrusted input in Qualcomm's QSEE firmware[109]. For an overview of ARM TrustZone including a survey of academic work and past security vulnerabilities of TrustZone-based firmware see [106].

For their mass-market phones these companies have R&D budgets that dwarf some countries' national budgets. If even they have trouble securing their secure embedded software stacks, what is a smart meter manufacturer to do? If a standard as in case of the German one requires IP gateways to speak TLS, a protocol that is notoriously tricky to implement correctly[53], the manufacturer is short on options to secure their product.

Since thorough formal verification of code is not yet within reach for either large-scale software development or code heavy in side-effects such as embedded firmware or industrial control software[100] the two most effective measures for embedded security are reducing the amount of code on one hand, and labor-intensively reviewing and testing this code on the other hand. A smart meter manufacturer does not have a say in the former since it is bound by the official regulations it has to comply with, and will likely not have sufficient resources for the latter. We are left with an impasse: Manufacturers in this field

---

[10] Modern system-on-chips integrate one or several CPUs with a multitude of peripherals, from memory and DMA controllers over 3D graphics accelerators down to general-purpose IO modules for controlling things like indicator LEDs. Most SoCs boot from one of several boot devices such as flash memory, Ethernet or USB according to a configuration set by pin-strapping configuration IOs or through write-only fuse bits.

Physically, one of the processing cores of the SoC (usually one of the main CPU cores) is connected such that it is taken out of reset before all other devices, and is tasked with enabling and configuring all other peripherals of the SoC. In order to run later intialization code or more advanced bootloaders, this core on startup runs a very small piece of code hard-burned into the SoC in the factory. This ROM loader initializes the most basic peripherals such as internal SRAM memory and selects a boot device for the next bootloader stage.

Apple's ROM loader measures only a few hundred bytes. It performs authorization checks to ensure only software authorized by Apple is booted. The present flaw allows an attacker to circumvent these checks and boot their own code on a USB-connected iPhone. This compromises Apple's chain of trust from ROM loader to userland right at its root. Since this is a flaw in the factory-programmed first stage read-only boot code of the SoC it cannot be patched in the field.

[11] `http://cve.circl.lu/cve/CVE-2016-1919`

likely do not have the security resources to keep up with complex standards requirements. At the same time they have no option to reduce the scope of their implementation to alleviate the burden on firmware security.

2.4.4 Attack avenues in the smart grid

If we model the smart grid as a control system responding to changes in inputs by regulating outputs, on a very high level we can see two general categories of attacks: Attacks that directly change the state of the outputs, and attacks that try to influence the outputs indirectly by changing the system's view of its inputs. The former would be an attack such as shutting down a power plant to decrease generation capacity[84]. The latter would be an attack such as forging grid frequency measurements where they enter a power plant's control systems to provoke the control systems to oscillate[77, 130, 75].

Communication channel attacks

Communication channel attacks are attacks on the communication links between smart grid components. This could be attacks on IP-connected parts of the core network or attacks on shared busses between smart meters and IP gateways in substations. Generally, these attacks can be mitigated by securing the aforementioned communication links using modern cryptography. IP links can be protected using TLS, and more low-level busses can be protected using more lightweight Noise[103]-based protocols.

Cryptographic security transforms an attackers ability to read and manipulate communication contents into a mere denial of service attack. Thus, in addition to cryptographic security safety under DoS conditions must be ensured for continued system performance under attacks. This safety property is identical with the safety required to withstand random outages of components, such as communication link outages due to physical damage from storms, flooding etc[110]. In general attacks at the meter level are hard to weaponize. Meters primarily serve billing purposes. The use of smart meter data for load forecasting is not yet common practice. Once it is this data will only be used to refine existing forecasting models that are based on aggregate data collected at higher vantage points in the distribution grid. This combination of smart metering data with more trusted aggregate data from sensors within the grid infrastructure limits the potential impact of a data falsification attack on smart meters. It also allows the utility to identify potentially corrupt meter readings and thus detect manipulation above a certain threshold. In order for an attack to have more far-reaching consequences the attacker would need to compromise additional grid infrastructure[75, 77].

Exploiting centralized control systems

The type of smart grid attack most often cited in popular discourse, and to the author's knowledge the only type that has so far been carried out in practice, is a direct attack on centralized control systems. In this attack, computer components of control systems are compromised by the same techniques used to compromise any other kind of computer system such as spearfishing, exploiting insecure services running on internet-exposed ports and using one compromised system to compromise other systems on the same ostensably secure internal network. These attacks are very powerful as they yield the attacker direct control over whatever outputs the compromised control systems are controlling. If an attacker manages to compromise the right set of control computers, they may even be able to cause physical damage[84].

Despite their potentially large impact, these attacks are only moderately interesting from a scientific perspective. For one, their mitigation mostly consists of a straightforward application of decades-old security best practices. Though there is room for the implementation of genuinely new, power systems-specific security systems in this field, the general state of the art is lacking behind other fields of embedded security. From this background low-hanging fruit should take priority[78]. Given political will these systems can readily be fortified. There is only a comparatively small number of them and having a technician drive to every one of them in turn to install a firmware security update is feasible.

Control function exploits

Control function exploits are attacks on the mathematical control loops used by the centralized control system. One example of this type of attack are resonance attacks as described in [130]. In this kind of attack, inputs from peripheral sensors indicating grid load to the centralized control system are carefully modified to cause a disproportionately large oscillation in control system action. This type of attack relies on complex resonance effects that arise when mechanical generators are electrically coupled. These resonances, colloquially called “modes”, are well-studied in power system engineering[108, 56, 45, 30]. Even disregarding modern attack scenarios, for stability electrical grids are designed with measures in place to dampen any resonances inherent to grid structure. These resonances are hard to analyze since they require an accurate grid model and they are unlikely to be noticed under normal operating conditions.

Mitigation of these attacks can be achieved by ensuring unmodified sensor inputs to the control systems in the first place. Carefully designing control systems not to exhibit exploitable behavior such as oscillations is also possible but harder.

Endpoint exploits

The one to us rather interesting attack on smart grid systems is someone exploiting the grid’s endpoint devices such as smart electricity meters. These meters are deployed on a massive scale, with at least one meter per household on average[12]. Once compromised, restoration to an uncompromised state can be difficult if it requires physical access to thousands of devices in hard-to-access locations.

---

[12] Households rarely share a meter but some households may have a separate meter for detached properties such as a detached garage or basement.

By compromising smart electricity meters, an attacker can forge the distributed energy measurements these devices perform. In a best-case scenario, this might only affect billing and lead to customers being under- or over-charged if the attack is not noticed in time. In a less ideal scenario falsified energy measurements reported by these devices could impede the correct operation of centralized control systems.

In some countries such as the UK smart meters have one additional function that is highly useful to an attacker: They contain high-current disconnect switches to disconnect the entire household or business in case electricity bills are left unpaid for a certain period. In countries that use these kinds of systems on a widespread level, the load disconnect switch is controlled by the smart meter's central microcontroller. This allows anyone compromising this microcontroller's firmware to actuate the disconnect switch at will. Given control over a large number of network-connected smart meters, an attacker might thus be able to cause large-scale disruptions of power consumption[5, 116]. Combined with an attack method such as the resonance attack from [130] that was mentioned above, this scenario poses a serious threat to grid stability.

In places where Demand-Side Management (DSM) is common this functionality may be abused in a similar way. In DSM the smart metering system directly controls power to certain devices such as heaters. The utility can remotely control the turn-on and turn-off of these devices to smoothen out the load curve. In exchange the customer is billed a lower price for the energy consumed by these loads. DSM was traditionally done in a federated fashion usually through low-frequency PLC over the distribution grid[40]. Smart metering systems no longer require large, resource-intensive transmitters in substations and bear the potential for a rollout of such technology on a much wider scale than before. This leads to a potentially significant role of DSM systems in the impact calculation of an attack on a smart metering system. DSM does not control as much load capacity as remote disconnect switches do but the attacks cited in the above paragraph still fundamentally apply.

2.4.5 Practical threats

As a highly integrated system the electrical grid is vulnerable to attacks from several angles. One way to classify attacks is by their motivation. Along this axis we found the following motives:

Service disruption. An attack aimed at disrupting service could e.g. aim at causing a blackout. It could also take aim in a more subtle way targeting a degradation of parameters such as power quality (voltage, frequency and waveform). It could target a particular customer, geographic area or all parts of the grid. Possible motivations range from a tennage hacker's boredom to actual cyberwar[26, 84].

Commercial disruption. Simple commercial motives already motivate a wide variety of attacks on grid infrastructure[32]. Though generally mostly harmless from a cypersecurity point of view there are instances where these attacks put the lives of both the attacker and bystanders at grave risk[5]. Such attacks generally aim at the meter itself but a more sophisticated attacker might also target the utility's backend computer bureaucracy.

Data extraction. The smart grid collects large amounts of data on both individual consumers and on an aggregate level. The privacy risk in individual consumer's data is obvious. On the web data collection practices ranging from questionable to flat-out illegal have widely proliferated for various purposes including election manipulation[64]. Assuming criminals in this field would eschew fertile ground such as this due to legal or ethical concerns is optimistic. Taking the risk to individual customer's data out of the equation even aggregate data is still highly attractive to some. Aggregate real-time electricity usage data is a potential source on timely information on matters such as national social events (through TV set energy consumption[57]) or the state of the economy.

A factor to consider in all these cases is that one actor's attacks have the potential to weaken system security overall. An attacker might add new backdoors to gain persistence or they might disable existing mitigations to enable further steps of their attack.

In this paper we will largely concentrate on attacks of the first type because they both have the most serious consequences and the most motivated attackers. Attackers that may want to disrupt service include nation state's cyberwar operations. This type of attacker is both highly skilled and highly funded.

2.4.6 Conclusion or, why we are doomed

We can conclude that a compromise of a large number of smart electricity meters cannot be ruled out. The complexity of network-connected smart meter firmware makes it exceedingly unlikely that it is in fact flawless. Large-scale deployments of these devices sometimes with disconnect relays make them an attractive target for attackers interested in causing grid instability. The attacker model for these devices includes nation states, who have considerable resources at their disposal.

For a reasonable guarantee that no large-scale compromises of hard- and software built today will happen over a span of some decades, we would have to radically simplify its design and limit attack surface. Unfortunately, the complexity of smart electricity meter implementations mostly stems from the large list of requirements these devices have to conform with. Alas, the standards have already been written, political will has been cast into law and changes that reduce scope or functionality have become exceedingly unlikely at this point.

A general observation with smart grid systems of any kind is that they comprise a departure from the federated control structure of yesterday's "dumb" grid and the advent of centralization to an enormous scale. This modern, centralized infrastructure has been carefully designed to defend against malicious actors and all involved parties have an interest in keeping it secure but in centralized systems scaling attacks is inherently easier than in decentralized systems[4]. An attacker can employ centralized control to their advantage. From this perspective the centralization of smart metering control systems–sometimes up to a national level[5, 4]–poses a security risk.

Chapter 3

Restoring endpoint safety in an age of smart devices

As laid out in the previous section we cannot fully rule out a large-scale compromise of smart energy meters at some point in the long-term future. Instead we have to rephrase our claim to security. We cannot rule out exploitation: We have to limit its impact. Assuming that we cannot strip any functionality from smart meters all we can do is to flush out an attacker once they are in. Mitigation replaces prevention.

In a worst-case scenario an attacker would gain unconstrained code execution e.g. by exploiting a flaw in a network protocol implentation. Smart meters use standard microcontrollers that do not have advanced memory protection functions (cf. Section 2.2.2). We can assume the attacker has full control over the main microcontroller given any such flaw. With this control they can actuate the disconnect switch if present. They can transmit data through the device's communication interfaces or use the user interface components such as LEDs and the LCD. Using the self-programming capabilities of flash microcontrollers an attacker could even gain persistency. Note that in systems separating cryptographic functions into some form of cryptographic module[1] we can be optimistic and assume the attacker has not yet compromised this cryptographic co-processor.

With the meter's core microcontroller under attacker control we cannot use this microcontroller to restore control over the system. We have no way of ensuring the attacker does not simply delete a security mechanism we include in the core microcontroller's firmware. Theoretically a secure boot implementation could be used to ensure meters boot into a safe state after temporary power loss but we cannot rely on secure boot being present on every smart meter application controller. Nowadays secure boot is a standard feature in many SoC aimed at smartphones or smart TVs but it is still very uncommon in microcontrollers.

Our solution to this problem is to add another smaller microcontroller to the smart meter design. This microcontroller will contain a small piece of software that receives cryptographically authenticated commands from utility companies. On demand it can reset the meter's core microcontroller to a

---

[1] such as systems used in Germany[17].

known-good state. To reliably flush out an attacker from a compromised core microcontroller we re-program the core microcontroller in its entirety. We propose using JTAG to re-program the core microcontroller with a known-good firmware image read from a sufficiently large SPI flash connected to the reset controller. JTAG is supported by most microcontrollers complex enough to be used in a smart meter design. JTAG programming functionality can be ported to a new microcontroller with relatively little work.

Our solution requires the core mircocontroller's JTAG interface to be activated (i.e. not fused-shut). For our solution to work the core microcontroller firmware must not be able to permanently disable the JTAG interface by itself. In microcontrollers that do not yet provide this functionality this is a minor change that could be added to a custom microcontroller variant at low cost. On most microcontrollers keeping JTAG open should not interfere with code readout protection[2]. Code secrecy should be of no concern[111] here but some manufacturers have strong preferences due to a fear of copyright infringement.

3.1 The theory of endpoint safety

In order to gain anything by adding our reset controller to the smart meter's already complex design we must satisfy two interrelated conditions.

1. *security* means our reset controller itself does not have any remotely exploitable flaws

2. *safety* menas our reset controller will perform its job as intended

Note that our *security* property includes only remote exploitation, and excludes any form of hardware attack. Even though most smart meters provide some level of physical security, we do not wish to make any assumptions on this. In the following section we will elaborate our attacker model and it will become apparent that sufficient physical security to defend against all attackers in our model would be infeasible, and thus we will design our overall system to remain secure even if we assume some number of physically compromised devices.

3.1.1 Attack characteristics

The attacker model the two above conditions must hold under is as follows. We assume three angles of attack: Attacks by the customer themselves, attacks by an insider within the metering systems controlling utility company and lastly attacks from third parties. Examples for these third parties are hobbyist hackers or outside cybercriminals on the one hand, but also other companies participating in the smart grid infrastructure besides the utility company such as intermediary providers of meter-reading services.

Due to the critical nature of the electrical grid, we have to include hostile state actors in our attacker model. When acting directly, these would be

---

[2] Readout protection usually forces a device to erase its program and data memories before allowing JTAG access.

classified as third-party attackers by the above schema, but they can reasonably be expected to be able to assume either of the other two roles as well e.g. through infiltration or bribery. In the generalized attacker model in [48] the authors give a classification of attacker types and provide a nice taxonomy of attacker properties. In their threat/capability rating, criminals are still considered to have higher threat rating than state-sponsored attackers. The New York Times reported in 2016 that some states recruit their hacking personnel in part from cybercriminals. If this report is true, in a worst-case scenario we have to assume a state-sponsored attacker to be the worst of both types. Comparing this against the other attacker types in [48], this state-sponsored attacker is strictly worse than any other type in both variables. We are left with a highly-skilled, very well-funded, highly intentional and motivated attacker.

Based on the above classification of attack angles and our observations on state-sponsored attacks, we can adapt [48] to our problem, yielding the following new attacker types:

1. **Utility company insiders controlled by a state actor.** We can ignore the other internal threats described in [48] since an insider coöperating with a state actor is strictly worse in every respect.
2. **State-sponsored external attackers.** A state actor can directly attack the system through the internet and with proper operations security they do not risk exposure or capture.
3. **Customers controlled by a state actor.** A state actor can very well compromise some customers for their purposes. They might either physically infiltrate the system posing as legitimate customers, or they might simply deceive or bribe existing customers into coöperation.
4. **Regular customers.** A hostile state actor might gain control of some number of customers through means such as voluntary coöperation, bribery or infiltration but this limits the scale of an attack since an attacker has to avoid arousing premature attention. Though regular customers may not have the motivation, skill or resources of a state-sponsored attacker, potentially large numbers of them may try to attack a system out of financial incentives[5, 32]. To allow for this possibility, we consider regular customers separate from state actors posing as customers.

3.1.2 Overall structural system security

Considering overall security, we first introduce the reset authority, a trusted party acting as the single authority for issuing reset commands in our system. In practice this trusted party may be part of the utility company, part of an external regulatory body or a hybrid setup requiring both to coöperate. We assume this party will be designed to be secure against all of the above attacker types. The precise design of this trusted party is out of scope for this work but we will provide some practical suggestions on how to achieve security below in Section 5.3.

Using an asymmetric cryptographic design centered around the reset authority, we rule out all attacks except for denial-of-service attacks on our system by any of the four attacker types. All reset commands in our system originate from the reset authority and are cryptographically secured to provide authentication and tamper detection. Under this model attacks on the electrical grid components between the reset authority and the customer device degrade into denial of service attacks. To ensure the *safety* criterion from Section 3.1 holds we must make sure our cryptography is secure against man-in-the-middle attacks and we must try to harden the system against denial-of-service attacks by the attacker types listed above. Given our attacker model we cannot fully guard against this sort of attack but we can at least choose a communication channel that is resilient under the above model.

Finally, we have to consider the issue of hardware security. We will solve the problem of physical attacks by simply not programming any secret information into devices. This also simplifies hardware production. We consider supply-chain attacks out-of-scope for this work.

3.1.3 Complex microcontroller firmware

The *security* property from 3.1 is in a large part reliant on the security of our reset controller firmware. The best method to increase firmware security is to reduce attack surface by limiting external interfaces as much as possible and by reducing code complexity as much as possible. If we avoid the complexity of most modern microcontroller firmware we gain another benefit beyond implicitly reduced attack surface: If the resulting design is small enough we may even succeed in formal verification of our security properties. Though formal verification tools are not yet suitable for highly complex tasks they are already adequate for small amounts of code and simple interfaces.

3.1.4 Modern microcontroller hardware

Microcontrollers have gained enormously in both performance and efficiency as well as in peripheral support. Alas, these gains have largely been driven by insatiable customer demand for faster, more powerful chips and for the longest time security has not been considered important outside of some specific niches such as smartcards. A few years ago a microcontroller would spend its entire lifetime without ever being exposed to any networks[4]. Though this trend has been reversing with the increasing adoption of internet-of-things things and more advanced security features have started appearing in general-purpose microcontrollers, most still lack even basic functionality found in processors for computers or smartphones.

One of the components lacking from most microcontrollers is strong memory protection or even a memory mapping unit as it is found in all modern computer processors and SoCs for applications such as smartphones. Without an MPU (Memory Protection Unit) or MMU (Memory Management Unit) many memory safety mitigations cannot be implemented. This and the absence of virtualization tools such as ARM's TrustZone make hardening microcontroller firmware a big task. It is very important to ensure memory safety in microcontroller firmware through tools such as defensive coding, extensive testing and formal verification.

In our design we achieve simplicity on two levels: One, we isolate the very complex metering firmware from our reset controller by having both run on separate microcontrollers. Two, we keep the reset controller firmware itself extremely simple to reduce attack surface there. Our protocol only has one message type and no state machine.

3.1.5 Safety vs. security: Opting for restoration instead of prevention

By implementing our reset system as a physically separate microcontroller we sidestep most security issues around the main application microcontroller. There are some simple measures that can be taken to harden its firmware. Implementing industry best practices such as memory protection or stack canaries will harden the system and increase the cost of an attack but it will not yield a system that we can be confident enough in to say it is fully secure. The complexity of the main application controller firmware makes fully securing the system a formidable effort–and one that would have to be repeated by every meter vendor for every one of their code bases.

In contrast to this our reset system does not provide any additional security. Any attack that could occur without it can still occur with it in place. What it provides is a fail-safe mechanism that can quickly immobilize a malicious actor mid-attack. It does this in a way that can be adapted to any meter architecture and any microcontroller platform with low effort since it relies on established standard interfaces such as JTAG and SWD. Concentrating research and development resources on a single platform like this allows for a system that is more economical to implement across device series and across vendors.

Attack resilience in the power grid can benefit from a safety-focused approach. The greater threat such an attack poses is not the temporary denial of service of utility metering functions. Even in a highly integrated smart grid as envisioned by utility companies these measurement functions are used by utility companies to increase efficiency and reduce cost but are not necessary for the grid to function at all. Thus if we can provide mere *safety* with a fail-safe semantic instead of unattainable perfect *security* we have gained resilience against a large class of realistic attack scenarios.

3.1.6 Technical outline of a safety reset system

There are several ways our system could be practically implemented. The most basic way is to add a separate microcontroller connected to the meter's main application MCU and optionally other embedded microcontrollers such as modems. This discrete chip could either be placed on the metering board itself or it could be placed on a separate PCB connected to the programming interface(s) of the metering board. In certain cases the latter might allow its use in otherwise unmodified legacy designs.

The safety reset controller would be a much simpler MCU than the meter's main application controller. Its software can be kept simple leading to low program flash and RAM requirements. Since it does not need to address rich periphery such as external parallel memory, LCDs etc. it can be a physically small, low-pin count device. If the main application controller is supposed to be reset to a full factory image with little or no reduced functionality its firmware image size is certainly too large for the reset controller's embedded flash. Thus a realistic setup would likely use an external SPI flash chip to store this image.

The most likely interfaces to reset the main application controller and possibly other microcontrollers such as modem chips would be the controller's integrated programming port such as JTAG. Parallel high-voltage flash programming has come to be uncommon in modern microcontrollers and most nowadays use some form of a serial interface. There exist a variety of serial programming and debug interfaces but JTAG has grown to be by far the most broadly supported one and has largely displaced vendor-specific debug interfaces except for very small devices.

The kind of microcontroller that would likely be used as the main application controller in a smart meter application will almost certainly support JTAG. These microcontrollers are high pin-count devices since they need to connect to a large set of peripherals such as the LCD and the large program flash makes it likely for a proper debugging interface to be present. The one remaining issue in this coarse technical outline is what communication interface should be used to transmit the trigger command to the reset controller. In the following section we will give an overview on communication interfaces established in energy metering applications and evaluate each of them for our purpose.

3.2 Communication channels on the grid

There is a number of well-established technologies for communication on or along power lines. We can distinguish three basic system categories: Systems using separate wires (such as DSL over landline telephone wiring), wireless radio systems (such as LTE) and *power line communication* (PLC) systems that reüse the existing mains wiring and superimpose data transmissions onto the 50 Hz mains sine[58, 69].

For our scenario, we will ignore short-range communication systems. There exists a large number of *wideband* power line communication systems that are popular with consumers for bridging Ethernet segments between parts of an apartment or house. These systems transmit up to several hundred megabits per second over distances up to several tens of meters[69]. Technologically, these wideband PLC systems are very different from *narrowband* systems used by utilities for load management among other applications and they are not relevant to our analysis.

3.2.1 Power line communication (PLC) systems and their use

In long-distance communications for applications such as load management, PLC systems are attractive since they allow re-using the existing wiring infrastructure and have been used as early as in the 1930s[49]. Narrowband PLC systems are a potentially low-cost solution to the problem of transmitting data at small bandwidth over distances of several hundred meters up to tens of kilometers.

Narrowband PLC systems transmit on the order of Kilobits per second or slower. A common use of this sort of system are *ripple control* systems. These systems superimpose a low-frequency signal at some few hundred Hertz carrier frequency on top of the 50Hz mains sine. This low-frequency signal is used to encode switching commands for non-essential residential or industrial loads. Ripple control systems provide utilities with the ability to actively control demand while promising savings in electricity cost to consumers[40].

In any PLC system there is a strict trade-off between bandwidth, power and distance. Higher bandwidth requires higher power and reduces maximum transmission distance. Where ripple control systems usually use few transmitters to cover the entire grid of a regional distribution utility, higher bandwidth bidirectional systems used for automatic meter reading (AMR) in places such as Italy or France require repeaters within a few hundred meters of a transmitter.

3.2.2 Landline and wireless IP-based systems

Especially in automated meter reading (AMR) infrastructure the cost-benefit trade-off of power line systems does not always work out for utilities. A common alternative in these systems is to use the public internet for communication. Using the public internet has the advantage of low initial investment on the part of the utility company as well as quick commissioning. Disadvantages compared to a PLC system are potentially higher operational costs due to recurring fees to network providers as well as lower reliability. Being integrated into power grid infrastructure, a PLC system's failure modes are highly correlated with the overall grid. Put briefly, if the PLC interface is down, there is a good chance that power is out, too. In contrast general internet services exhibit a multitude of failures that are entirely uncorrelated to power grid stability. For purposes such as meter reading for billing purposes, this stability is sufficient. However for systems that need to hold up in crisis situations such as the recovery system we are contemplating in this thesis, the public internet may not provide sufficient reliability.

3.2.3 Short-range wireless systems

Smart meters contain copious amounts of firmware but still pale in comparison to the complexity of full-scale computers such as smartphones. For short-range communication between a meter and a cellular radio gateway mounted nearby or between a meter and a meter reading operator in a vehicle on the street a protocol such as Wifi (IEEE 802.11) is too complex. Absent widely-used standards in this space proprietary radio protocols grew attractive. These are often based on some standardized lower-level protocol such as ZigBee (IEEE 802.15) but entirely home-grown ones also exist. To the meter manufacturer a proprietary radio protocol has several advantages. It is easy to implement and requires no external certification. It can be customized to its specific application. In addition it provides vendor lock-in to customers sharing infrastructure such as a cellular radio gateway between multiple devices. In other fields a lack of standardization has led to a proliferation of proprietary protocols and a fragmented protocol landscape. This is a large problem since the consumer cannot easily integrate products made by different manufacturers into one system. In advanced metering infrastructure this is unlikely to be a disadvantage since usually there is only one distribution grid operator for an area. Shared resources such as a cellular radio gateway would most likely only be shared within a single building and usually they are all operated by the same provider.

Systems in Europe commonly support Wireless M-Bus, an European standardized protocol[94] that operates on several ISM bands[3]. ZigBee is another popular standard and some vendors additionally support their own proprietary protcols[4].

3.2.4 Frequency modulation as a communication channel

For our system, we chose grid frequency modulation (henceforth GFM) as a low-bandwidth unidirectional broadcast communication channel. Compared to traditional PLC, GFM requires only a small amount of additional equipment, works reliably throughout the grid and is harder to manipulate by a malicious actor.

Grid frequency in Europe's synchronous areas is nominally 50 Hertz, but there are small load-dependent variations from this nominal value. Any device connected to the power grid (or even just within physical proximity of power wiring) can reliably and accurately measure grid frequency at low hardware overhead. By intentionally modifying grid frequency, we can create a very low-bandwidth broadcast communication channel. Grid frequency modulation has only ever been proposed as a communication channel at very small scales in microgrids before[126] and to our knowledge has not yet been considered for large-scale application.

Advantages of using grid frequency for communication are low receiver hardware complexity as well as the fact that a single transmitter can cover an entire synchronous area. Though the transmitter has to be very large and powerful the setup of a single large transmitter faces lower bureaucratic hurdles than integration of hundreds of smaller ones into hundreds of local systems that each have autonomous governance.

---

[3] Frequency bands that can be used for *Industrial, Scientific and Medical* applications by anyone and that do not require obtaining a license for transmitter operation. Manufacturers can use whatever protocol they like on these bands as long as they obtain certification that their transmitters obey certain spectral and power limitations.

[4] For an example see [65].

The frequency dependency of grid frequency

Despite the awesome complexity of large power grids the physics underlying their response to changes in load and generation is surprisingly simple. Individual machines (loads and generators) can be approximated by a small number of differential equations and the entire grid can be modelled by aggregating these approximations into a large system of nonlinear differential equations. Evaluating these systems it has been found that in large power grids small signal steady state changes in generation/consumption power balance cause an approximately linear change in frequency[79, 30, 120, 119]. *Small signal* here describes changes in power balance that are small compared to overall grid power. *Steady state* describes changes over a time frame of multiple waveform cycles as opposed to transient events that only last a few milliseconds.

This approximately linear relationship allows the specification of a coefficient with unit $\mathrm{W\,Hz^{-1}}$ linking power differential ΔP and frequency differential Δf. In this thesis we are using the European power grid as our model system. We are using data provided by ENTSO-E (formerly UCTE), the governing association of European transmission system operators. In our calculations we use data for the continental European synchronous area, the largest synchronous area. $\frac{\Delta P}{\Delta f}$, called *Overall Network Power Frequency Characteristic* by ENTSO-E is around $25\,\mathrm{GW\,Hz^{-1}}$.

We can derive general design parameter for any system utilizing grid frequency as a communication channel from the policies of ENTSO-E[120, 46]. Any such system should stay below a modulation amplitude of 100 mHz which is the threshold defined in the ENTSO-E incidents classification scale for a Scale 0-1 (from “Anomaly” to “Noteworthy Incident” scale) frequency degradation incident[120] in the continental Europe synchronous area.

Control systems coupled to grid frequency

The ENTSO-E Operations Handbook Policy 1 chapter[120] defines the activation threshold of primary control to be 20 mHz. Ideally, a modulation system would stay well below this threshold to avoid fighting the primary control reserve. Modulation line rate should likely be on the order of a few hundred Millibaud. Modulation at these rates would outpace primary control action which is specified by ENTSO-E as acting within between “a few seconds” and 15 s.

Keeping modulation amplitude below this threshold would help to avoid spuriously triggering these control functions. The effective *Network Power Frequency Characteristic* of primary control in the European grid is reported by ENTSO-E at around $20\,\mathrm{GW\,Hz^{-1}}$. This works out to an upper bound on modulation power of $20\,\mathrm{MW\,mHz^{-1}}$.

An outline of practical transmitter implementation

In its most basic form a transmitter for grid frequency modulation would be a very large controllable load connected to the power grid at a suitable vantage point. A spool of wire submerged in a body of cooling liquid such as a small lake along with a thyristor rectifier bank would likely suffice to perform this function during occasional cybersecurity incidents. We can however decrease hardware and maintenance investment even further compared to this rather uncultivated solution by repurposing regular large industrial loads as transmitters in an emergency situation. For some preliminary exploration we went through a list of energy-intensive industries in Europe[42]. The most electricity-intensive industries in this list are primary aluminum and steel production. In primary production raw ore is converted into raw metal for further refinement such as casting, rolling or extrusion. In steelmaking iron is smolten in an electric arc furnace. In aluminum smelting aluminum is electrolytically extracted from alumina. Both processes involve large amounts of electricity with electricity making up 40 % of production costs. Given these circumstances a steel mill or aluminum smelter would be good candidates as transmitters in a grid frequency modulation system.

In aluminum smelting high-voltage mains is transformed, rectified and fed into about 100 series-connected electrolytic cells forming a *potline*. Inside these pots alumina is dissolved in molten cryolite electrolyte at about 1000 °C and electrolysis is performed using a current of tens or hundreds of Kiloampère. The resulting pure aluminum settles at the bottom of the cell and is tapped off for further processing.

Like steelworks, aluminum smelters are operated night and day without interruption. Aside from metallurgical issues the large thermal mass and enormous heating power requirements do not permit power cycling. Due to the high costs of production inefficiencies or interruptions the behavior of aluminum smelters under power outages is a well-characterized phenomenon in the industry. The recent move away from nuclear power and towards renewable energy has lead to an increase in fluctuations of electricity price throughout the day. These electricity price fluctuations have provided enough economic incentive to aluminum smelters to develop techniques to modulate smelter power consumption without affecting cell lifetime or product quality[38, 43]. Power outages of tens of minutes up to two hours reportedly do not cause problems in aluminum potlines and are in fact part of routine operation for purposes such as electrode changes[43, 99].

The power supply system of an aluminum plant is managed through a highly-integrated control system as keeping all cells of a potline under optimal operating conditions is challenging. Modern power supply systems employ large banks of diodes or SCRs[5] to rectify low-voltage AC to DC to be fed into the potline[8]. The potline voltage can be controlled almost continuously through a combination of a tap changer and a transductor. The individual cell voltages can be controlled by changing the anode to cathode distance (ACD) by physically lowering or raising the anode. The potline power supply is connected to the high voltage input and to the potline through isolators and breakers.

In an aluminum smelter most of the power is sunk into resistive losses and the electrolysis process. As such an aluminum smelter does not have any

---

[5] SCRs, also called thyristors, are electronic devices that are often used in high-power switching applications. They are normally-off devices that act like diodes when a current is fed into their control terminal.

significant electromechanical inertia compared to the large rotating machines used in other industries. Depending on the capabilities of the rectifier controls high slew rates are possible, permitting modulation at high[6] data rates.

Avoiding dangerous modes

Modern power systems are complex electromechanical systems. Each component is controlled by several carefully tuned feedback loops to ensure voltage, load and frequency regulation. Multiple components are coupled through transmission lines that themselves exhibit complex dynamic behavior. The overall system is generally stable, but may exhbit instabilities to particular small-signal stimuli[79, 30]. These instabilities, called *modes*, occur when due to mis-tuning of parameters or physical constraints the overall system exhibits oscillation at a particular frequency. [79] separates these modes into four categories:

Local modes where a single power station oscillates in some parameter,

Interarea modes where subsections of the overall grid oscillate with respect to each other due to weak coupling between them,

Control modes caused by imperfectly tuned control systems and

Torsional modes that originate from electromechanical oscillations in the generator itself.

The oscillation frequencies associated with each of these modes are usually between a few tens of Millihertz and a few Hertz[56, 45, 30]. It is hard to predict the particular modes of a power system at the scale of the central European interconnected system. Theoretical analysis and simulation may give rough indications but cannot yield conclusive results. Due to the obvious danger as well as high economical impact due to inefficiencies experimental measurements are infeasible. Modes are highly dependent on the power grid's structure and will change with changes in the power grid over time. For all of these reasons, a grid frequency modulation system must be designed very conservatively without relying on the absence (or presence) of modes at particular frequencies. A concrete design guideline that we can derive from this situation is that the frequency spectrum of any grid frequency modulation system should not exhibit large peaks and should avoid a concentration of spectral energy in small frequency bands.

---

[6] Aluminum smelter rectifiers are *pulse rectifiers*. This means instead of simply rectifying the incoming three-phase voltage they use a special configuration of transformer secondaries and in some cases additional coils to produce a large number of equally spaced phases (e.g. six) from a standard three-phase input. Where a direct-connected three-phase rectifier would draw current in six pulses per mains voltage cycle a pulse rectifier draws current in more, smaller pulses to increase power factor. For example a 12-pulse rectifier will draw current in 12 pulses per cycle. In the best case an SCR pulse rectifier switched at zero crossing should allow 0 % to 100 % load changes from one rectifier pulse to the next, i.e. within a fraction of a single cycle.

Overall system parameters

In conclusion we end up with the following tunable parameters for a grid frequency modulation based on a large controllable load:

Modulation amplitude. Amplitude is proportionally related to modulation power. In a practical setup we might realize a modulation power up to a few hundred MW which would yield a few tens of mHz of frequency amplitude.

Modulation preemphasis and slew-rate control. Preemphasis might be necessary to ensure an adequate Signal-to-Noise ratio (SNR) at the receiver. Slew-rate control and other shaping measures might be necessary to reduce the impact of these sudden load changes on the transmitter's primary function (say, aluminum smelting) and to prevent disturbances to other grid components.

Modulation frequency. For a practical implementation a careful study would be necessary to determine the optimal frequency band for operation. On one hand we need to prevent disturbances to the grid such as the excitation of local or inter-area modes. On the other hand we need to optimize Signal-to-Noise ratio (SNR) and data rate to achieve optimal latency between transmission start and reset completion and to reduce the overall burden on both transmitter and grid.

Further modulation parameters. The modulation itself has numerous parameters that are discussed in Section 3.3.2 below.

3.3 From grid frequency to a reliable communication channel

Based on the physical properties oulined above we will provide the theoretical groundwork for a practical communication system based on grid frequency modulation.

3.3.1 Channel properties

In this section we will explore how we can construct a reliable communication channel from the analog primitive we have outlined in the previous section. Our load control approach to grid frequency modulation leads to a channel with the following properties.

Slow-changing. Accurate grid frequency measurements take several periods of the mains sine wave. Faster sampling rates can be achieved with more complex specialized synchrophasor estimation algorithms but this will result in a trade-off between sampling rate and accuracy[9].

Analog. Grid frequency is an analog signal.

Noisy. While stable over long periods of time thanks to power stations' Load-Frequency Control systems[119] there are considerable random short-term variations. Our modulation amplitude is limited by technical and economic constraints so we have to find a system that will work at poor SNRs.

Polarized. Grid frequency measurements have an inherent sense of polarity that we can use in our modulation scheme.

3.3.2 Modulation and its parameters

In this section we will analyze what makes for a good set of parameters for a modulation scheme fitting grid frequency modulation.

As described before the grid's oscillatory modes mean that we should avoid any modulation technique that would concentrate energy in a small bandwidth. Taking this principle to its extreme provides us with a useful pointer towards techniques that might work well: Spread-spectrum techniques. By employing spread-spectrum modulation we can produce close to ideal frequency-domain behavior. Modulation energy is spread almost flatly across the modulation bandwidth[55]. At the same time we achieve modulation gain which increases system sensitivity. This modulation gain potentially allows us to use a weaker stimulus allowing for a further reduction of the probability of disturbance to the overall system. Spread-spectrum techniques also inherently allow us to trade-off receiver sensitivity for data rate. This tunability is a useful parameter in the overall system design.

Spread spectrum covers a whole family of techniques that are comprehensively explained in [55]. [55] divides spread spectrum techniques into the coarse categories of *Direct Sequence Spread Spectrum*, *Frequency Hopping Spread Spectrum* and *Time Hopping Spread Spectrum*.

In [55] a BPSK or similar modulation is assumed underlying the spread-spectrum technique. Our grid frequency modulation channel effectively behaves more like a DC-coupled wire than a traditional radio channel: Any change in excitation will cause a proportional change in the receiver's measurement. Using our FFT-based measurement methodology we get a real-valued signed quantity. In this way grid frequency modulation is similar to a channel using coherent modulation. We can utilize both signal strength and polarity in our modulation.

For our purposes we can discount both Time and Frequency Hopping Spread Spectrum techniques. Time hopping helps to reduce interference between multiple transmitters but does not help with SNR any more than Direct Sequence does since all it does is allowing other transmitters to transmit. Our system is strictly limited to a single transmitter so we do not gain anything through Time Hopping.

Frequency Hopping Spread Spectrum techniques require a carrier. Grid frequency modulation itself is very limited in peak frequency deviation Δf. Frequency hopping could only be implemented as a second modulation on top of GFM, but this would not yield any benefits while increasing system complexity and decreasing data bandwidth.

Direct Sequence Spread Spectrum is the only remaining approach for our application. Direct Sequence Spread Spectrum works by directly modulating a long pseudo-random bit sequence onto the channel. The receiver must know the same pseudo-random bit sequence and continuously calculates the correlation between the received signal and the pseudo-random template sequence mapped from binary $[0, 1]$ to bipolar $[1, -1]$. The pseudo-random sequence has an approximately equal number of 0 and 1 bits. The positive contribution of the $+1$ terms of the correlation template approximately cancel out with the -1 terms when multiplied with an uncorrelated signal such as white Gaussian noise.

By using a family of pseudo-random sequences with low cross-correlation channel capacity can be increased. Either the transmitter can encode data in the choice of sequence or multiple transmitters can use the same channel at once. The longer the pseudo-random sequence, the lower its cross-correlation with noise or other pseudo-random sequences of the same length. Choosing a long sequence we increase modulation gain while decreasing bandwidth. For any given application the sweet spot will be the shortest sequence that is long enough to yield sufficient SNR for subsequent processing layers such as channel coding.

A popular code used in many DSSS systems are Gold codes. A set of Gold codes has small cross-correlations. For some value n a set of Gold codes contains $2^n + 1$ sequences of length $2^n - 1$. Gold codes are generated from two different maximum length sequences generated by linear feedback shift registers (LFSRs). For any bit count n there are certain empirically determined preferred pairs of LFSRs that produce Gold codes with especially good cross-correlation. The $2^n + 1$ gold codes are defined as the XOR sum of both LFSR sequences shifted from 0 to $2^n - 1$ bit as well as the two individual LFSR sequences. Given LFSR sequences `a` and `b` in numpy notation this is `[a, b] + [ a ^ np.roll(b, shift) for shift in len(b) ]`.

In DSSS modulation the individual bits of the DSSS sequence are called *chips*. Chip duration determines modulation bandwidth[55]. In our system we are directly modulating DSSS chips on mains frequency without an underlying modulation such as BPSK as it is commonly used in DSSS systems.

3.3.3 Error-correcting codes

To reduce reception error rate we have to layer channel coding on top of the DSSS modulation. The messages we expect to transmit are at least a few tens of bits long. We are highly constrained in SNR due to limited transmission power and with lower SNR comes higher BER (Bit Error Rate). At a fixed BER, packet error rate grows exponentially with transmission length so for our relatively long transmissions we would realistically get unacceptable error rates.

Error correcting codes are a very broad field with many options for specialization. Since we are implementing only an advanced prototype in this thesis we chose to spend only limited resources on optimization and settled on a basic Reed-Solomon code. We have no doubt that applying a more state-of-the-art code we could gain further improvements in code overhead and decoding speed among others[88]. Since message length in our system limits system response time but we do not have a fixed target we can tolerate some degree of overhead. Decoding speed is of very low concern to us because our data rate is extremely low. We derived our implementation by adapting and optimizing an existing open source decoder that we validated on an open source encoder implementation. We generate test signals using a Python tool on the host.

3.3.4 Cryptographic security

Above the communication base layer elaborated in the previous section we have to layer a cryptographic protocol to ensure system security. We want to avoid a case where a third party could interfere with our system or even subvert this safety system itself for an attack. From a protocol security perspective the system we are looking for can informally be modelled as consisting of three parties: the trusted *transmitter*, one of a large number of untrusted *receivers*, and an *attacker*. These three play according to the following rules:

Access. Both transmitter and attacker can transmit any bit sequence.

Indistinguishability. The receiver receives any transmission by either but cannot distinguish between them.

Kerckhoff's principle. Since the protocol design is public and anyone can get access to an electricity meter the attacker knows anything any receiver might know[73, 104].

Priority. The transmitter is stronger than an attacker and will "win" during simultaneous transmission.

Seeding. Both transmitter and receiver can be seeded out-of-band with some information on each other such as public key fingerprints.

We are not considering situations where an attacker attempts to jam an ongoing transmission. In practice there are several avenues to prevent such attempts. Compromised large loads that are being abused by the attacker can be manually disconnected by the utility. Error-correcting codes can be used to provide resiliency against small-scale disturbances. Finally, the transmitter can be designed to have high enough power to be able to override any likely attacker.

With the above properties in mind our goal is to find a cryptographic primitive that has the following properties:

Authentication. The transmitter can produce a message bit sequence that a certain subset of receivers can identify as being generated by the transmitter. On reception of this sequence, all addressed receivers perform a safety reset.

Unforgeability. The attacker cannot forge a message, i.e. find a bit sequence other than one of the transmitter's previous messages that a receiver would accept. This implies that the attacker also cannot create a new distinct message from a previously transmitted message.

Brevity. The message should be short. Our communication channel is outrageously slow compared to anything else used in modern telecommunications and every bit counts.

On a protocol level we also have to ensure *idempotence*. Our system should have an at-most-once semantic. This means for a given message each receiver either performs exactly one safety reset or none at all, even if the message is re-transmitted by either the transmitter or an attacker. We cannot achieve the ideal exactly-once semantic wit pure protocol gymnastics since we are using an unidirectional lossy communication primitive. A receiver might be offline (e.g. due to a local power outage) and then would not hear the transmission even if our broadcast primitive was reliable. Since there is no back channel, the transmitter has no way of telling when that happens. The practical impact of this can be mitigated by the transmitter repeating the message a number of times.

It follows from the unforgeability requirement that we can trivially reach idempotence at the protocol level by keeping a database of all previous messages and only accepting new messages. By considering this in our cryptographic design we can reduce the storage overhead of this “database”.

Along with the indistinguishability property the access requirement implies that we need a cryptographic signature[81]. However, we have relaxed constraints on this signature compared to standard cryptographic practice[6]. While cryptographic signatures need to work over arbitrary inputs, all we want to “sign” here is the instruction to perform a safety reset. This is the only message we might ever want to transmit so our message space has only one element. The information content of our message thus is 0 bit! All the information we want to transmit is already encoded *in the fact that we are transmitting* and we do not require a further payload to be transmitted: We can omit the entirety of the message and just transmit whatever “signature” we produce[61, 62]. This is useful to conserve transmission bits so our transmission does not take an exceedingly long time over our extremely slow communication channel.

We can modify this construction to allow for a small number of bits of information content in our message (say two or three instead of zero) at no transmission overhead by transmitting the cryptographic signature as usual but simply omitting the message. The message contains only a few bits of information and we are dealing with minutes of transmission time so the receiver can reconstruct the message through brute-force. Though this trade-off between computation and data transmission might seem inelegant it does work for our extremely slow link for up to a few bits of information.

There is an important limitation in the rules of our setup above: The attacker can always record the reset bit sequence the transmitter transmits and replay that same sequence later. Even without cryptography we can trivially prevent an attacker from violating the at-most-once criterion. If every receiver memorizes all bit sequences that have been transmitted so far it can detect replays. With this mitigation by replaying an older authentic transmission an attacker can cause receivers that were offline during the original transmission to reset at a later point. Considering our goal is to reset them in the first place this should not pose a threat to the system’s safety or security.

A possible scenario would be that an attacker first causes enough havoc for authorities to trigger a safety reset. The attacker would record the trigger transmission. We can assume most meters were reset during the attack. Due to this the attacker cannot cause a significant number of additional resets immediately afterwards. However, the attacker could wait several years for a number of new meters to be installed that might not yet have updated firmware that includes the last transmission. This means the attacker could cause them to reset by replaying the original sequence.

A possible mitigation for this risk would be to introduce one bit of information into the trigger message that is ignored by the replay protection mechanism. This *enable* bit would be 1 for the actual reset trigger message. After the attack the transmitter would then perform scheduled transmissions of a "disarm" message that has this bit set to 0. This message informs all new meters and meters that were offline during the original transmission of the original transmission for replay protection without actually performing any further resets.

We could use any of several traditional asymmetric cryptographic primitives to produce these signatures. The comparatively high computational effort required for signature verification would not be an issue. Transmissions take several minutes anyway and we can afford to spend some tens of seconds even in signature verification. Transmission length and by proxy system latency would be determined by the length of the signature. For RSA signature length is the modulus length (i.e. larger than 1000 bit for very basic contemporary security). For elliptic curve-based systems curve length is approximately twice the security level and signature size is twice the curve length because two curve points need to be encoded[4]. For contemporary security this results in more than 300 bit transmission length. We can exploit our unique setting's low message entropy to improve on this by basing our scheme on a cryptographic hash function used as a one-way pseudo-random function (PRF). Hash-based signature schemes date back to the very beginnings of cryptographic signatures[6, 35, 80]. Today, in general applications schemes based on asymmetric cryptography are preferred but hash-based signature systems have their applications in certain use cases. One example of such a scheme is the TESLA scheme[102] that is the basis for navigation message authentication in the European Galileo global navigation satellite system. Here, a system based purely on asymmetric primitives would result in too much computation and communication overhead[63]. In the following sections we will introduce the foundations of hash-based signatures before deriving our authentication scheme.

Lamport signatures

1979, Lamport in [80] introduced a signature scheme that is based only on a one-way function such as a cryptographic hash function. The basic observation is that by choosing a random secret input to a one-way function and publishing the output, one can later prove knowledge of the input simply by publishing it. In the following paragraphs we will describe a construction of a one-time signature scheme based on this observation. The scheme we describe is the one usually called a "Lamport Signature" in modern literature but is slightly different from the variant described in the 1979 paper. For our purposes we can consider both to be equivalent.

Setup. In a Lamport signature, for an n-bit hash function H the signer generates a private key $s = \left(s_{b,i} | b \in \{0,1\}, 0 \leq i < n\right)$ of $2n$ random strings of length n. The signer publishes a public key $p = \Big(p_{b,i} = H\left(s_{b,i}\right), b \in \{0,1\}, 0 \leq i < n\Big)$ that is simply the list of hashes of each of the random strings that make up the private key.

Signing. To sign a message m, the signer publishes the signature $\sigma = \left(\sigma_i = k_{H(m)_i,i}\right)$ where $H(m)_i$ is the i-th bit of H applied to m. That is, for the i-th bit of the message's hash $H(m)$ the signer publishes either of $p_{0,i}$ or $p_{1,i}$ depending on the hash bit's value, keeping the other entry of P secret.

Verification. The verifier can compute $H(m)$ themselves and check the corresponding entries $\sigma_i = k_{H(m)_i}$ of S correctly evaluate to $p_{b,i} = H\left(s_{b,i}\right)$ from P under H.

The above scheme is a one-time signature scheme only. After one signature has been published for a given key, the corresponding key must not be reüsed for other signatures. This is intuitively clear as we are effectively publishing part of the private key as the signature, and if we were to publish a signature for another message an attacker could derive additional signatures by "mixing" the two published signatures.

Winternitz signatures

An improvement to basic Lamport signatures as described above are Winternitz signatures as detailed in [91, 37]. Winternitz signatures reduce public key length as well as signature length for hash length n from $2n$ to $\mathcal{O}\left(n/t\right)$ for some choice of parameter t (usually a small number such as 4).

Setup. The signer generates a private key $s = (s_i)$ consisting of $\lceil \frac{n}{t} \rceil$ random bit strings. The signer publishes a public key $p = \Big(H^{2^t}\left(s_i\right)\Big)$ where each element $H^{2^t}\left(s_i\right)$ is the 2^t-fold recursive application of H to s_i.

Signing. The signer splits m padded to a multiple of t bits into $\lceil \frac{n}{t} \rceil$ chunks m_i of t bit each. The signer publishes the signature $\sigma = \left(\sigma_i = H^{m_i}\left(s_i\right)\right)$.

Verification. The verifier can calculate for each $\sigma_i = H^{m_i}\left(s_i\right)$ that $H^{2^t - m_i}\left(\sigma_i\right) = H^{2^t - m_i}\left(H^{m_i}\left(s_i\right)\right) = H^{2^t - m_i + m_i}\left(s_i\right) = p_i$.

To prevent an attacker from forging additional signatures from one signature by calculating $\sigma_i' = H\left(\sigma_i\right)$ matching $m_i' = m_i + 1$, this scheme is usually paired with a simple checksum as described in [91].

Using hash-based signatures for trigger authentication

Applying these concepts the most basic trigger authentication scheme possible would be to simply generate a random secret key bit string s and publish $p = H(s)$ for some hash function H. To activate the trigger, $\sigma = s$ is published and receivers verify that $H(\sigma) = p = H(s)$. This simplistic scheme has one main disadvantage: It is a fundamentally one-time construction. To prevent an attacker from re-triggering a receiver a second time by replaying a valid trigger σ all receivers have to blacklist any "used" σ. Alas, this means we can only ever trigger a receiver *once*. The good part is that any receiver that missed this trigger can still be triggered later, but the bad part is that once s is burned we are out of options. The trivial solution to this would be to simply provision each receiver with a whole list of public keys in advance. This however takes n times the amount of space for n-fold retriggerability and for each one we have to memorize separately whether it has been used up. Luckily we can easily derive a scheme that yields n-fold retriggerability and naturally memorizes replay state while using no more space than the original scheme by taking some inspiration from Winternitz signatures.

In this improved scheme the secret key s is still a random bit string. The public key is $p = H^n(s)$ for n-times retriggerability. The i-th time the trigger is activated, $\sigma_i = H^{n-i}(s)$ is published, and every receiver can verify that $\sigma_{i-1} = H(\sigma_i)$ with $\sigma_0 = p$. In case a receiver missed one or more previous triggers it continues computing $H\left(H\left(\sigma_i\right)\right)$ and $H\left(H\left(H\left(\sigma_i\right)\right)\right)$ and so on until either reaching the n-th recursion level–indicating an invalid signature–or finding $H^n\left(\sigma_i\right) = \sigma_j$ with σ_j being the last signature this receiver recorded or p in case there is none.

This scheme provides replay protection since the receiver memorizes the last signature they acted on. Public key length is equal to the length of the hash function H used. Even for our embedded systems use case n can realistically be up to $\mathcal{O}\left(10^3\right)$, which is enough for our purposes. This use of a hash chain for event authentication is identical to the one in the S/KEY one-time password system[6, 61, 62].

The "disarm" message we discussed above for replay protection can be integrated into this scheme by encoding the "enable" bit into the least significant bit of n in our H^n construction. In the chain of valid signatures every second one would be a disarm signature: Reset and disarm signatures would alternate in this scheme. By skipping a disarm signature two resets can still be triggered directly after one another.

In practice it may be useful to have some control over which meters reset. An attack exploiting a particular network protocol implementation flaw might only affect one series of meters made by one manufacturer. Resetting *all* meters may be too much in this case. A simple solution for this is to define addressable subsets of meters. "All meters" along with "meters made by manufacturer x" and "meters of model y" are good choices for such scopes. On the cryptographic level the protocol state is simply duplicated for each scope. This incurs memory and computation overhead linear in the number of scopes but device memory requirements are small at a few bytes only and computation is of no concern due

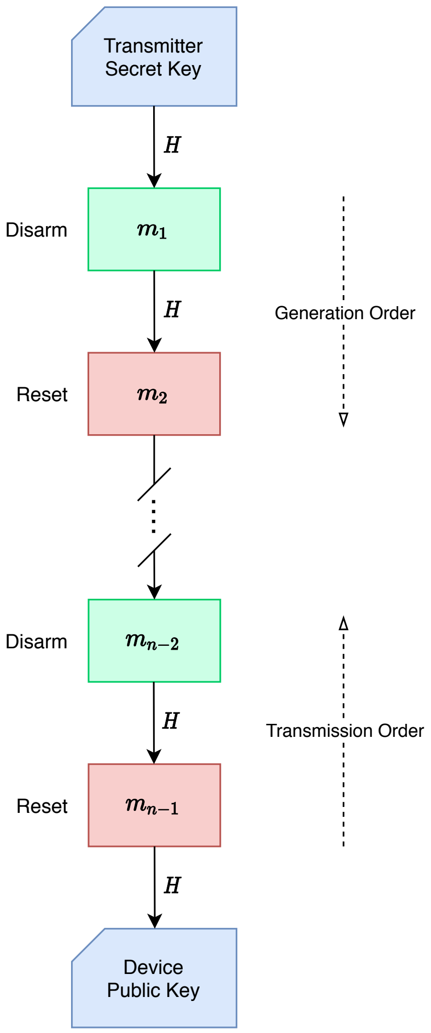


Figure 3.1: The hash chain between secret transmitter key and public device key. Each step represents one invocation of the hash function. To generate a new chain a random transmitter key is generated, then hashed n times to generate the corresponding device key. A new trigger message can be generated by generating the key at depth $m - 1$ where m is the height of the last used trigger, or n initially. Every second trigger message is a disarm message and every second one a reset message. Depending on which is needed either one may be skipped.

to the very slow channel so this simple solution is adequate. The transmitter has to either store copies of all scope's keys or derive these keys from a root key using the scope's identifier. Keys are small and the transmitter would be using a regular server or hardware security module for key management so either easily feasible.

A diagram of the key structure in this key management scheme is shown in Figure 3.1. The transmitter key management is shown in Figure 3.2. This scheme is simplistic but suffices for our prototype in Section 4.3 and may even be useful in a practical implementation. During standardization of a safety reset system the key management system would most likely have to be customized to the particular application's requirements. Developing an universal solution is outside the scope of this work.

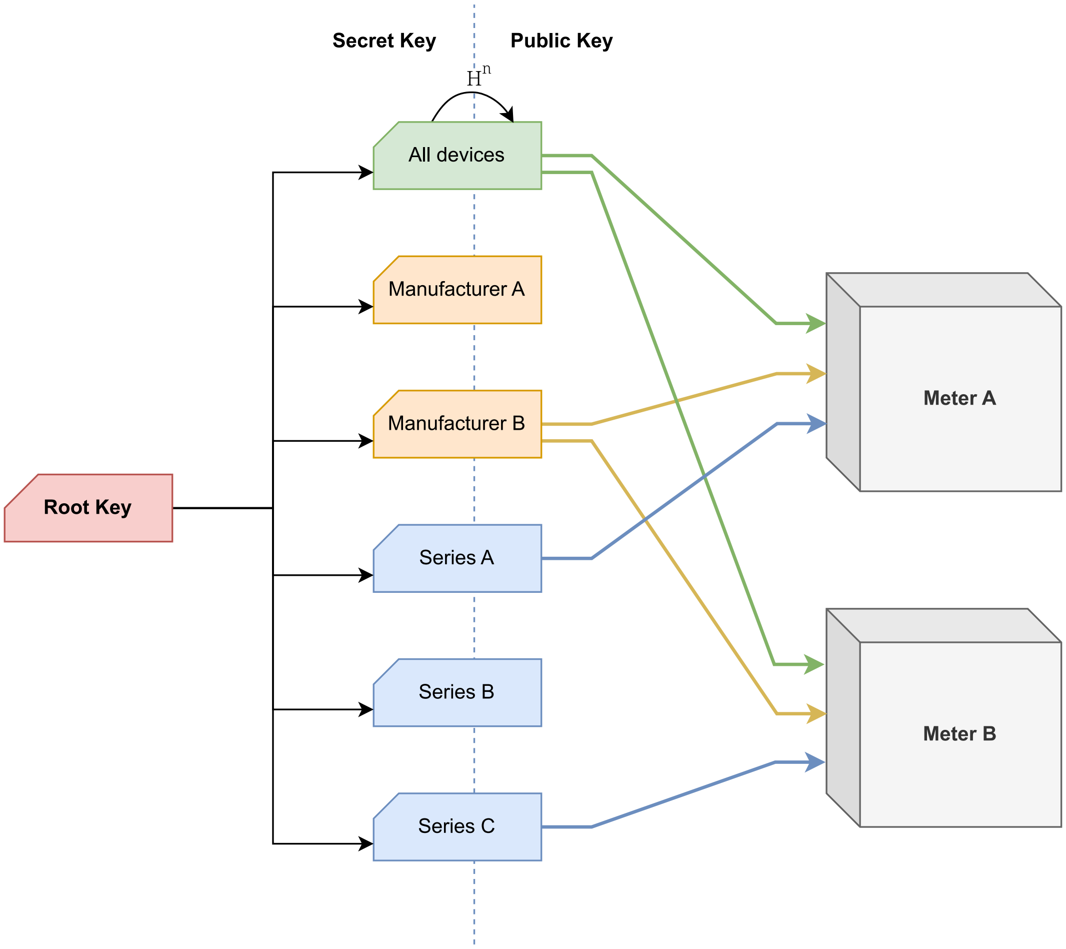


Figure 3.2: An illustration of a key management system using a common master key. First, the transmitter derives one secret key for each addressable group from the master key. Then public device keys are generated like in Figure 3.1. Finally for each device the manufacturer picks the group public keys matching the device. In this example one device is a series A meter made by manufacturer B so it gets provisioned with the keys for the "all devices", "manufacturer B" and "series A" groups. The other device is also made by manufacturer B but is a series C device so it gets provisioned with the "all devices", "manufacturer B" and "series C" device keys. In this example the transmitter stores (or is able to derive) all six shown group keys, but each device only needs to store the three applying to it–one for each of the three scopes "all devices", "manufacturer" and "series".

Chapter 4

Practical implementation

To validate the practical feasibility of the theoretical concepts we laid out in the previous chapter we decided to build a prototype of a safety reset controller. In this section we describe the reasoning behind the components of this prototype and the engineering that went into its firmware. The prototype consists of a smart meter whose application microcontroller is reset by a microcontroller on an external circuit board. We lay out how we extensively tested all parts of our firmware implementation. We conclude with results of a practical end-to-end experiment exercising every part of our prototype.

4.1 Data collection for channel validation

To design a solid system we needed to parametrize mains frequency variations under normal conditions. To set modulation amplitude as well as parameters of our modulation scheme we need a frequency spectrum of mains frequency variations (that is $\mathcal{F}\left(f(V(t))\right)$: Taking mains frequency $f(x)$ as a variable, the frequency spectrum of that variable, as opposed to the frequency spectrum of mains voltage $V(t)$ itself).

4.1.1 Grid frequency estimation

In commercial power systems Phasor Measurement Units (PMUs, also called *synchrophasors*) are used to precisely measure parameters of the mains voltage waveform, one of which is grid frequency. PMUs are used as part of SCADA systems controlling transmission networks to characterize the operational state of the network.

From a superficial viewpoint measuring grid frequency might seem like a simple problem. Take the mains voltage waveform, measure time between two rising-edge (or falling-edge) zero-crossings and take the inverse $f = t^{-1}$. In practice, phasor measurement units are significantly more complex than this. This discrepancy is due to the combination of both high precision and quick response that is demanded from these units. High precision is necessary since variations of mains frequency under normal operating conditions are quite small–in the range of 5 mHz to 10 mHz over short intervals of time. Relative to the nominal 50 Hz this is a derivation of less than 100 ppm. Relative to the corresponding period of 20 ms this means a time derivation of about 2μs from cycle to cycle. From this it is already obvious why a simplistic measurement cannot yield the required precision for manageable averaging times: We would need either an ADC sampling rate in the order of megabits per second or for a reconstruction through interpolated readings an impractically high ADC resolution.

Detail on the inner workings of commercial phasor measurement units is scarce but given their essential role to SCADA systems there is a large amount of academic research on such algorithms[96, 33, 9]. A popular approach to these systems is to perform a Short-Time Fourier Transform (STFT) on ADC data sampled at high sampling rate (e.g. 10 kHz) and then perform analysis on the frequency-domain data to precisely locate the peak at 50 Hz. A key observation here is that FFT bin size is going to be much larger than required frequency resolution. This fundamental limitation follows from the Nyquist criterion[**shannon01**] and if we had to process an *arbitrary* signal this would severely limit our practical measurement accuracy [1]. For this reason all approaches to grid frequency estimation are based on a model of the voltage waveform. Nominally this waveform is a perfect sine at $f = 50$ Hz. In practice it is a sine at $f \approx 50$ Hz superimposed with some aperiodic noise (e.g. irregular spikes from inductive loads being energized) as well as harmonic distortion that is caused by topologically nearby devices with power factor $\cos\theta \neq 1.0$. Under a continuous fourier transform over a long period the frequency spectrum of a signal distorted like this will be a low noise floor depending mainly on aperiodic noise on which a comb of harmonics as well as some sub-harmonics of $f \approx f_{\text{nom}} = 50$ Hz is riding. The main peak at $f \approx f_{\text{nom}}$ will be very strong with the harmonics being approximately an order of magnitude weaker in energy and the noise floor being at least another order of magnitude weaker. See Figure 4.8 for a measured spectrum. This domain knowledge about the expected frequency spectrum of the signal can be employed in a number of interpolation techniques to reconstruct the precise frequency of the spectrum's main component despite distortions and the comparatively coarse STFT resolution.

Published grid frequency estimation algorithms such as [96, 33] are rather sophisticated and use a combination of techniques to reduce numerical errors in FFT calculation and peak fitting. Given that we do not need reference standard-grade accuracy for our application we chose to start with a very basic algorithm instead. We chose to use a general approach to estimate the precise fundamental frequency of an arbitrary signal that was published by experimental physicists Gasior and Gonzalez at CERN[50]. This approach assumes a general sinusoidal signal superimposed with harmonics and broadband noise. Applicable to a wide spectrum of practical signal analysis tasks it is a reasonable first-degree

---

[1] Some software packages providing FFT or STFT primitives such as scipy[127] allow the user to super-sample FFT output by specifying an FFT width larger than input data length, padding the input data with zeros on both sides. Note that in line with the Nyquist theorem this *does not* actually provide finer output resolution but instead just amounts to an interpolation between output bins. Depending on the downstream analysis algorithm it may still be sensible to use this property of the DFT for interpolation, but in general it will be computationally expensive compared to other interpolation methods and in any case it will not yield any better frequency resolution aside from a potential numerical advantage[51].

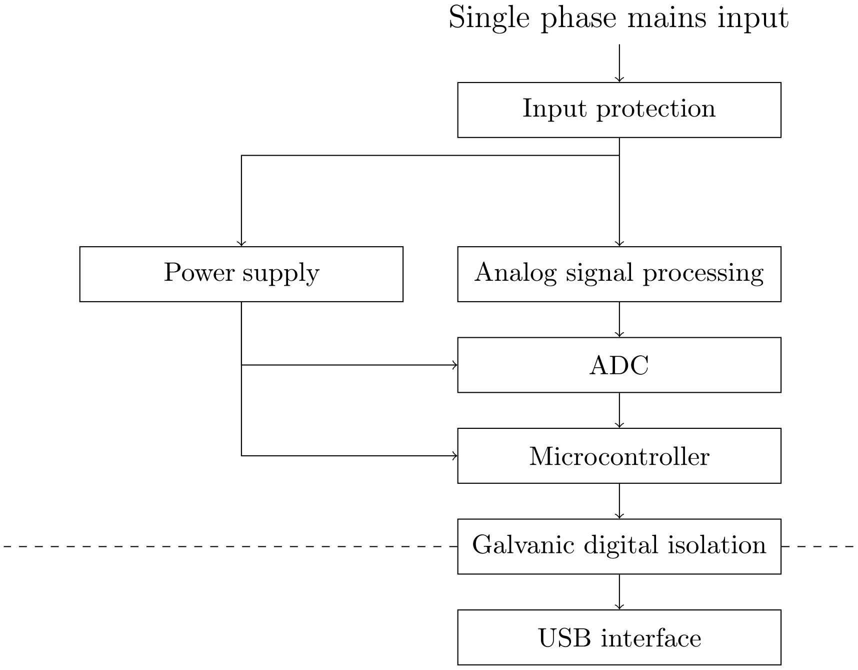


Figure 4.1: Frequency sensor hardware block diagram.

approximation of the much more sophisticated estimation algorithms developed specifically for power systems. Some algorithms use components such as kalman filters[96] that require a physical model. As a general algorithm [50] does not require this kind of application-specific tuning, eliminating one source of error.

The Gasior and Gonzalez algorithm[50] passes the windowed input signal through a DFT, then interpolates the signal's fundamental frequency by fitting a wavelet such as a Gaussian to the largest peak in the DFT results. The bias parameter of this curve fit is an accurate estimation of the signal's fundamental frequency. This algorithm is similar to the simpler interpolated DFT algorithm used as a reference in much of the synchrophasor estimation literature[13]. The three-term variant of the maximum side lobe decay window often used there is a Blackman window with parameter $\alpha = \frac{1}{4}$. Analysis has shown[9] that the interpolated DFT algorithm is worse than algorithms involving more complex models under some conditions but that there is *no free lunch* meaning that more complex perform worse when the input signal deviates from their models.

4.1.2 Frequency sensor hardware design

Our safety reset controller will have to measure mains frequency to later demodulate a reset signal transmitted through it. Since we have decided to do our own frequency measurement system here we can reüse this frequency measurement setup as a prototype for the frequency measurement component of the demodulation system we will develop later. Since we do not plan to do a large-scale field deployment of our measurement setup we can keep the hardware implementation simple by moving most of the signal processing to a regular computer and concentrating our hardware efforts on raw signal capture.

An overall block diagram of our system is shown in Figure 4.1. The microcontroller we chose is an `STM32F030F4P6` ARM Cortex M0 microcontroller made by ST Microelectronics. The ADC in Figure 4.1 in our implementation is the integrated 12-bit ADC of this microcontroller, which is sufficient for our purposes. The USB interface is a simple USB to serial converter IC (`CH340G`) and the galvanic digital isolation is accomplished with a pair of high speed optocouplers on its `RX` and `TX` lines. The analog signal processing is a simple voltage divider using high power resistors to get the required creepage along with some high frequency filter capacitors and an op-amp buffer. The power supply is an off-the-shelf mains-input power module. The system is implemented on a single two-layer PCB that is housed in an off-the-shelf industrial plastic case fitted with a printed label and a few status lights on its front. The schematics of our system can be found in Appendix A.

4.1.3 Clock accuracy considerations

Our measurement hardware will sample line voltage at some sampling rate f_S, e.g. 1 kHz. All downstream processing is limited in accuracy by the accuracy of f_S[2]. We generate our sampling clock in hardware by clocking the ADC from one of the microcontroller's timer blocks clocked from the microcontroller's system clock. This means our ADC's sampling window will be synchronized cycle-accurate to the microcontroller's system clock.

Our downstream estimation of mains frequency by nature is relative to our sampling frequency f_S. In the setup described above this means we have to make sure our system clock is stable. A frequency deviation of 1 ppm in our system clock causes a proportional grid frequency measurement error of $\Delta f = f_{\text{nom}} \cdot 10^{-6} = 50\,\mu\text{Hz}$. In a worst-case scenario where our system is clocked from a particularly bad crystal that exhibits 100 ppm of instabilities over our measurement period we end up with an error of 5 mHz. This is well within our target measurement range, so we need a more stable clock source. Ideally we want to avoid writing our own clock conditioning code where we try to change an oscillators operating frequency to match some reference. Clock conditioning algorithms are complex[97] and in our case post processing of measurement data and simply adding an offset is simpler and less error-prone.

Our solution to these problems is to use a crystal oven[3]as our main system clock source. Crystal ovens are expensive compared to ordinary crystal oscillators. Since any crystal oven will be much more accurate than a standard room-temperature crystal we chose to reduce cost by using one recycled from old telecommunications equipment.

---

[2] We are not considering the effect of clock jitter. We are highly oversampling the signal and the FFT done in our downstream processing will average out small jitter effects leaving only frequency stability to worry about.

[3] A crystal oven is a crystal oscillator closely thermally coupled to a heater and temperature sensor and enclosed in a thermally isolated case. The heater is controlled to hold the crystal oscillator at a near constant temperature some tens of degrees Celsius above ambient temperature. Ambient temperature variations will be absorbed by the temperature control. This yields a crystal frequency that is almost completely unaffected by ambient temperature variations below the oven temperature and whose main remaining instability is aging.

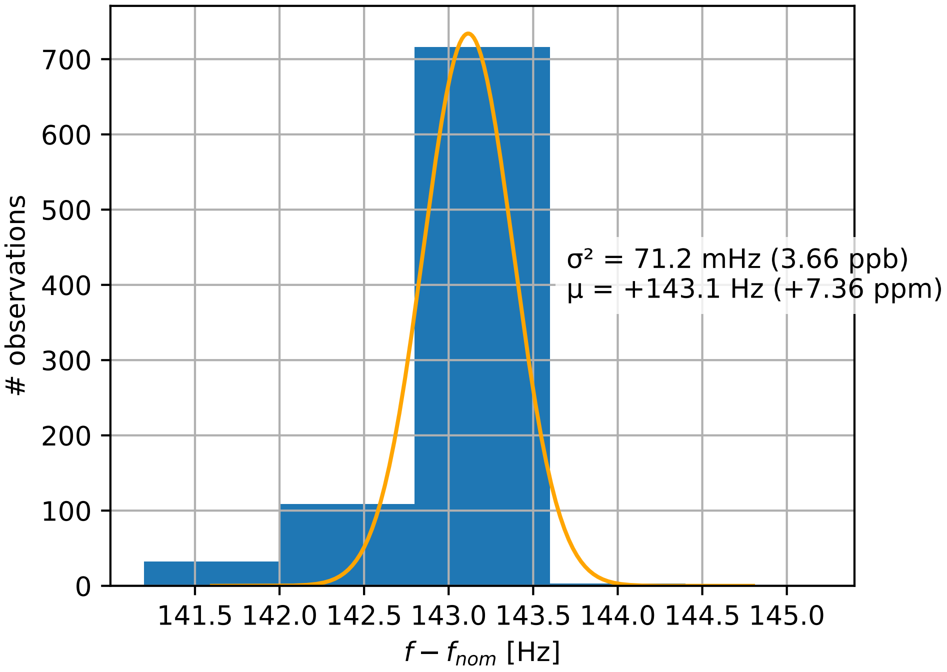


Figure 4.2: OCXO Frequency derivation from its nominal 19.440 MHz frequency measured against a GPS receiver's 1pps reference output.

To verify clock accuracy we routed an externally accessible SMA connector to a microcontroller pin that is routed to one of the microcontroller's timer inputs. By connecting a GPS 1pps signal to this pin and measuring its period we can calculate our system's Allan variance[4], thereby measuring both clock stability and clock accuracy. We ran a 4 hour test of our frequency sensor that generated the histogram shown in Figure 4.2. These results show that while we get a systematic error of about 10 ppm due to manufacturing tolerances the random error at less than 10 ppb is smaller than that of a room-temperature crystal oscillator by 3-4 orders of magnitude. Since we are interested in grid frequency variations over time but not in the absolute value of grid frequency the systematic error is of no consequence to us. The random error at 3.66 ppb corresponds to a frequency measurement error of about 0.2 µHz, well below what we can achieve at reasonable sampling rates and ADC resolution.

4.1.4 Firmware implementation

The firmware uses one of the microcontroller's timers clocked from an external crystal oscillator to produce an 1 ms tick that the internal ADC is triggered from for a sample rate of 1 ksps. Higher sample rates would be possible but reliable data transmission over the opto-isolated serial interface might prove challenging and 1 ksps already corresponds to 20 samples per cycle at f_{nominal}. This figure exceeds the Nyquist criterion by a factor of ten and is plenty for

---

[4] Allan variance is a measure of frequency stability between two clocks.

accurate measurements.

The ADC measurements are read using DMA and written into a circular buffer. Using DMA controller features this circular buffer is split in back and front halves with one being written to and the other being read at the same time. Buffer contents are moved from the ADC DMA buffer into a packet-based reliable UART interface as they come in. The UART packet interface keeps two ring buffers: One byte-based ring buffer for transmission data and one ring buffer pointer structure that keeps track of ADC data packet boundaries in the byte-based ring buffer. Every time a chunk of data is available from the ADC the data is framed into the byte-based ring buffer and the packet boundaries are logged in the packet pointer ring buffer. If the UART transmitter is idle at this time a DMA-backed transmission of the oldest packet in the packet ring buffer is triggered at this point. Data is framed using Consistent Overhead Byte Stuffing (COBS)[5][25] along with a CRC-32 checksum for error checking. When the host receives a new packet with a valid checksum it returns an acknowledgement packet to the sensor. When the sensor receives the acknowledgement, the acknowledged packet is dropped from the transmission packet ring buffer. When the host detects an incorrect checksum it simply stays quiet and waits for the sensor to resume with retransmission when the next ADC buffer has been received.

The serial interface logic presents most of the complexity of the sensor firmware. This complexity is necessary since we need reliable, error-checked transmission to the host. Though rare, bit errors on a serial interface do happen and data corruption is unacceptable. The packet layer queueing on the sensor is necessary since the host is not a realtime system and unpredictable latency spikes of several hundred milliseconds are possible.

The host in our recording setup is a Raspberry Pi 3 model B running a Python script. The Python script handles serial communication and logs data and errors into an SQLite database file. SQLite has been chosen for its simple yet flexible interface and its good tolerance of system resets due to unexpected power loss. Overall our setup performed adequately with IO contention on the Raspberry PI/Linux side causing only 16 skipped sample packets over a 68 hour recording span.

4.1.5 Frequency sensor measurement results

Our completed frequency sensor can be seen in Figure 4.3. The raw voltage waveform data we captured with it has been processed in the Jupyter Lab environment[76] and grid frequency estimates are extracted as described in Section 4.1.1 using the Gasior and Gonzalez[50] technique. The Jupyter note-

---

[5] COBS is a framing technique that allows encoding n bytes of arbitrary data into exactly $n + 1$ bytes with no embedded 0 bytes that can then be delimited using 0 bytes. COBS is simple to implement and allows both one pass decoding and encoding. The encoder either needs to be able to read up to 256 B ahead or needs a buffer of 256 B. COBS is very robust in that it allows self-synchronization. At any point a receiver can reliably synchronize itself against a COBS data stream by waiting for the next 0 byte. The constant overhead allows precise bandwidth and buffer planning and provides constant, good efficiency close to the theoretical maximum.

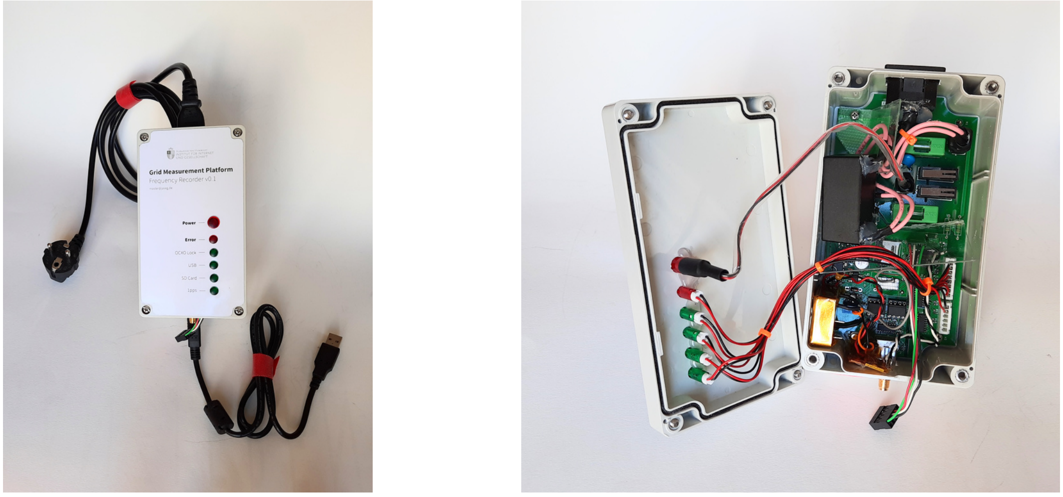


Figure 4.3: The finished grid frequency sensor device. The large yellow part on the bottom left is the crystal oven. The large black part is the power supply module. The microcontroller is on the bottom right of the device and the measurement circuit is in its middle. The device connects to the data recording computer via galvanically isolated USB on the bottom and to a regular wall socket through the IEC connector on the top of the device.

book we used for frequency measurement is included with the supplementary materials to this thesis. In Figure 4.4 we fed back to the frequency estimator its own output giving us an indication of its numerical performance. The result was 1.3 mHz of RMS noise over a 3600 s simulation time. This indicates performance is good enough for our purposes. In addition to this we validated our algorithm's performance by applying it to the test waveforms from [129]. In this test we got errors of 4.4 mHz for the *noise* test waveform, 0.027 mHz for the *interharmonics* test waveform and 46 mHz for the *amplitude and phase step* test waveform. Full results can be found in Figure 4.5.

Figures 4.6 and 4.7 show our measurement results over a 24-hour and a 2-hour window respectively.

4.2 Channel simulation and parameter validation

To validate all layers of our communication stack from modulation scheme to cryptography we built a prototype implementation in Python. Implementing all components in a high level language builds up familiarity with the concepts while taking away much of the implementation complexity. For our demonstrator we will not be able to use Python since our target platform is an inexpensive low-end microcontroller. Our demonstrator firmware will have to be written in a low-level language such as C or Rust. For prototyping these languages lack flexibility compared to Python.

To validate our modulation scheme we first performed a series of simulations on our Python demodulator prototype implementation. To simulate a

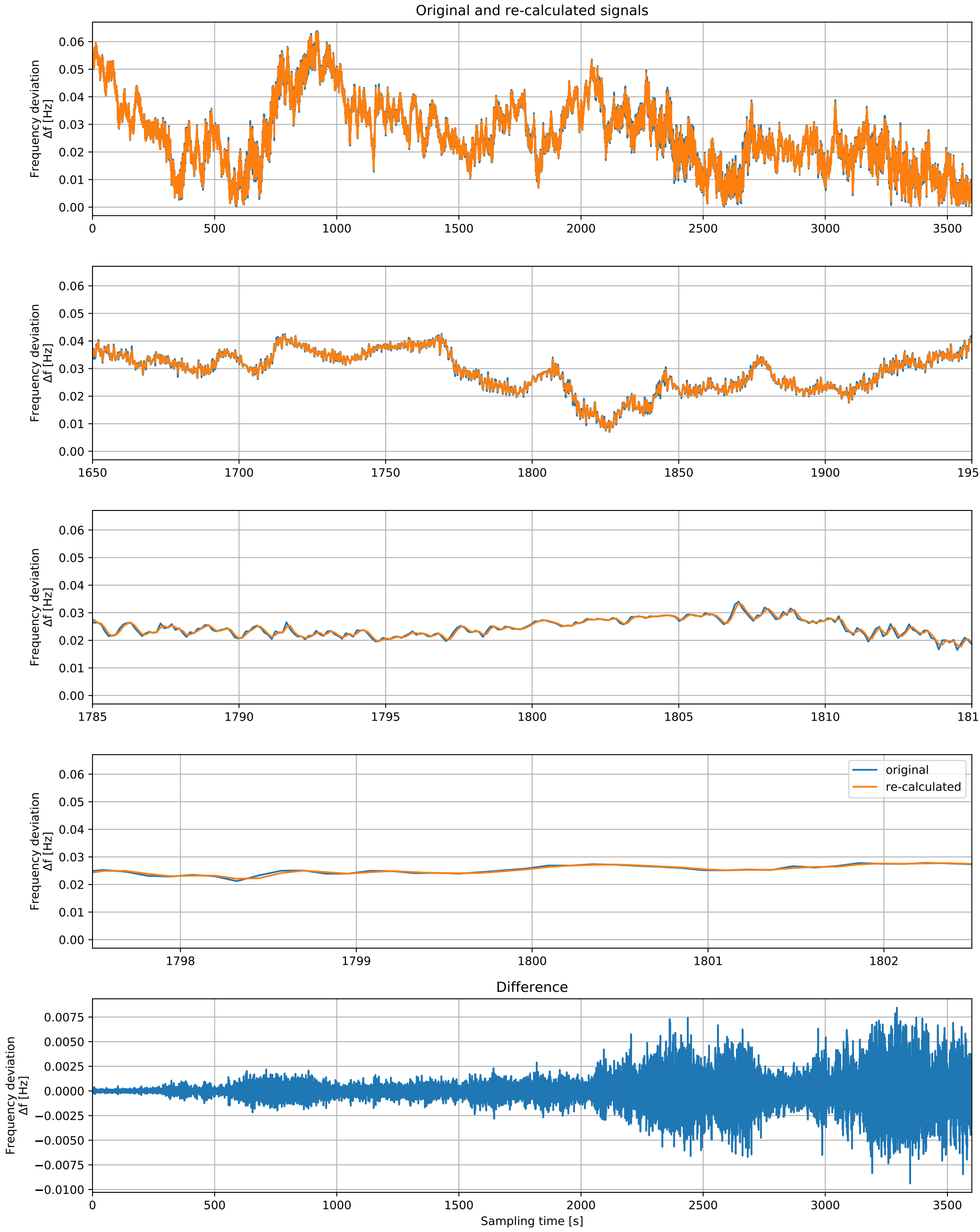


Figure 4.4: The frequency estimation algorithm applied to a synthetic noise-less mains waveform generated from its own output. This feedback simulation gives an indication of numerical errors in our estimation algorithm. The top four graphs show a comparison of the original trace (blue) and the re-calculated trace (orange). The bottom trace shows the difference between the two. As we can tell both traces agree very well with an overall RMS deviation of about 1.3 mHz. The bottom trace shows deviation growing over time. This is an effect of numerical errors in our ad hoc waveform generator.

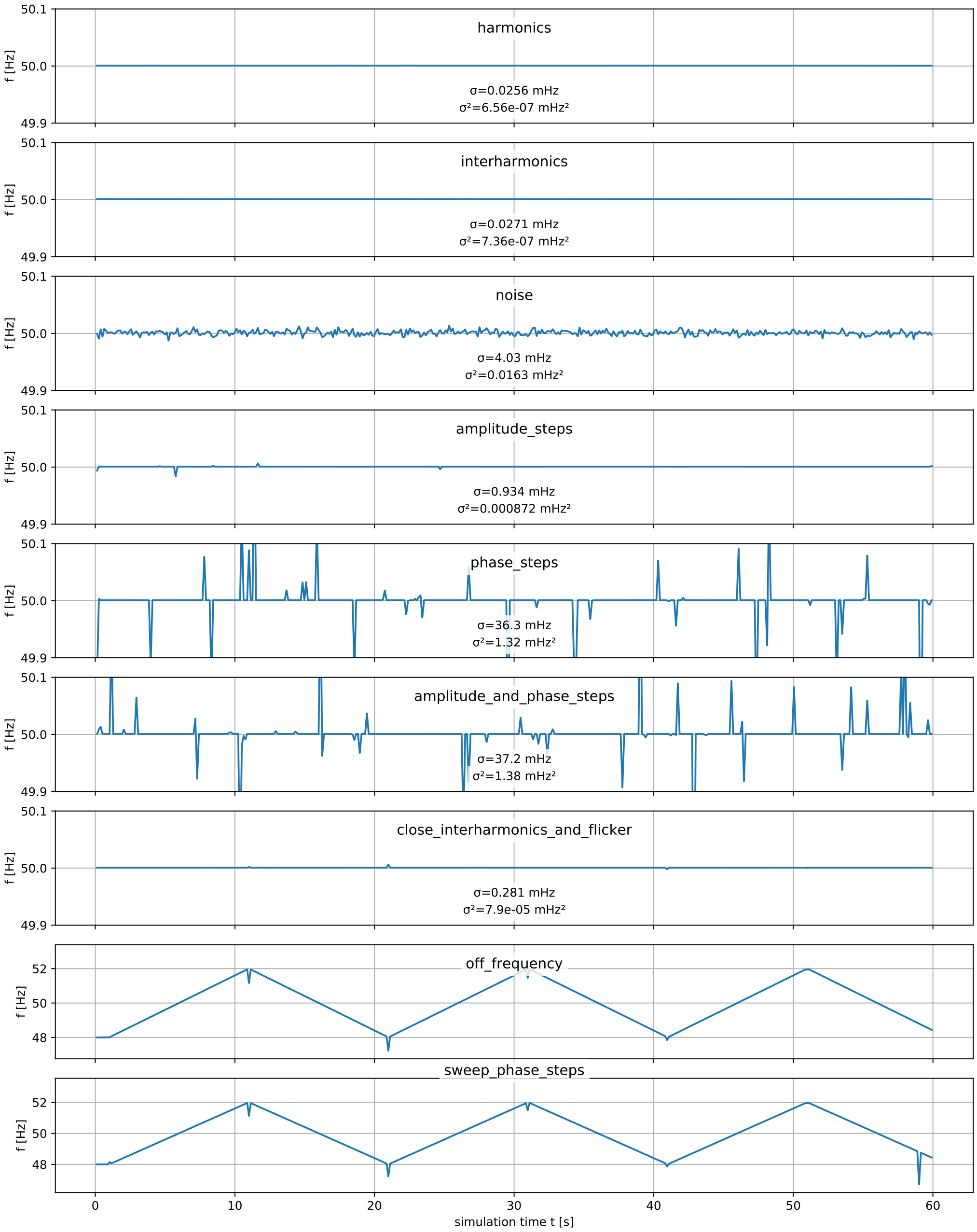


Figure 4.5: Performance of our frequency estimation algorithm under the test suite specified in [129]. Shown are standard deviation and variance measurements as well as time-domain traces of absolute differences.

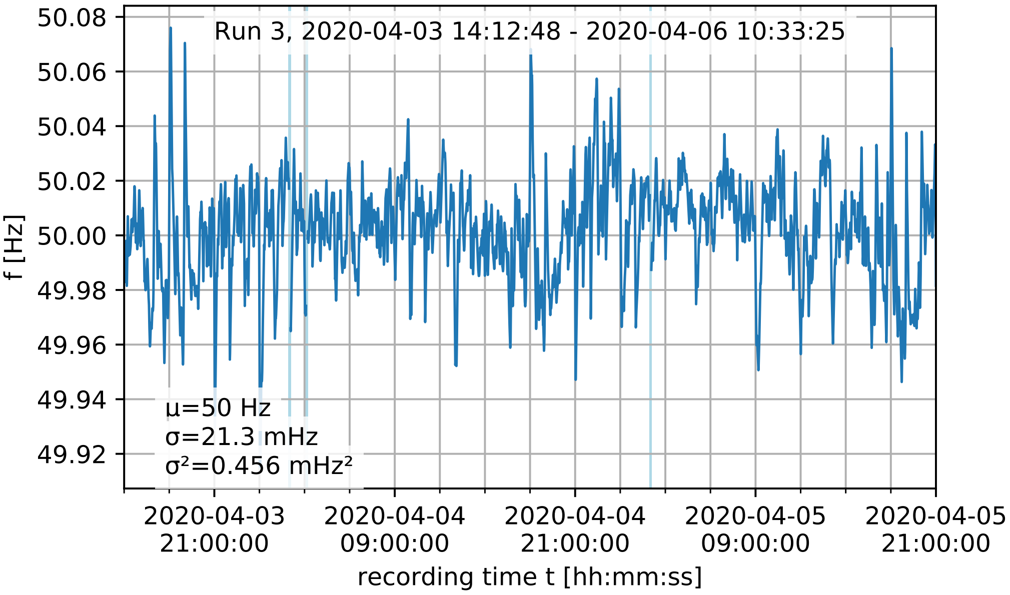
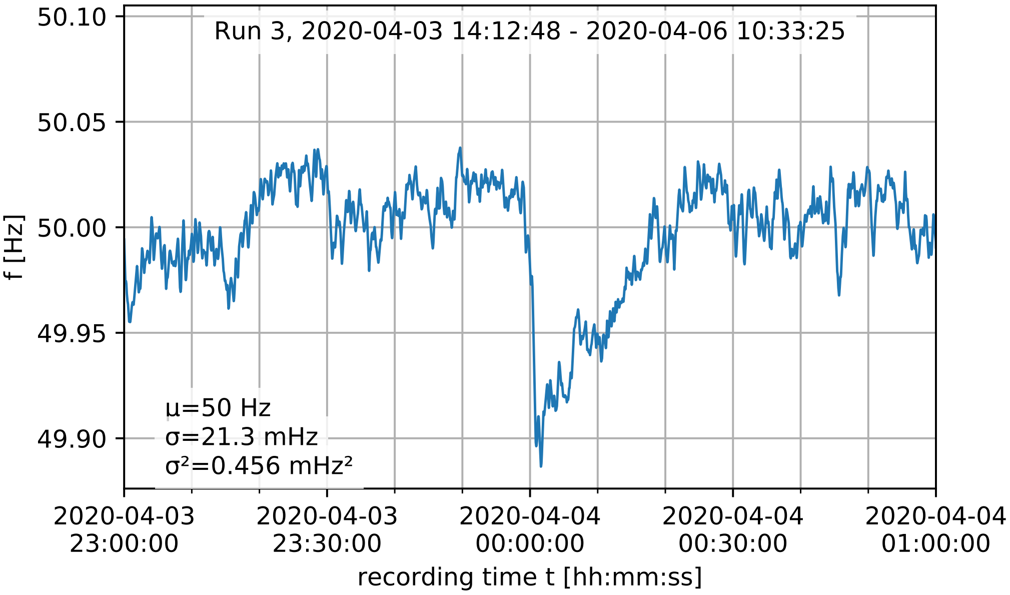


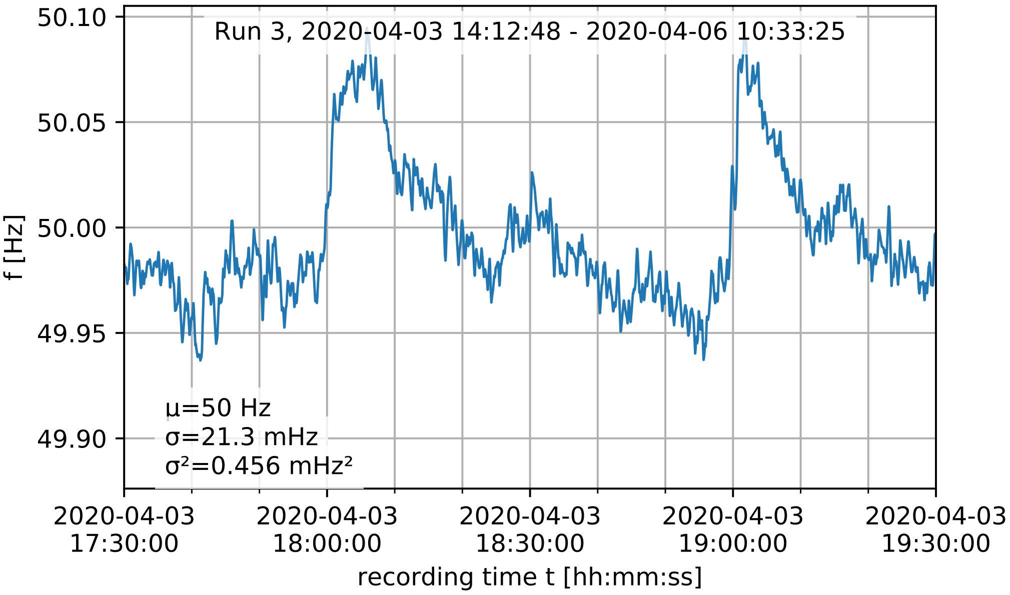
Figure 4.6: Trace of grid frequency over a 24 hour time span. One clearly visible feature are large positive and negative transients at full hours. Times shown are UTC. Note that the European continental synchronous area that this sensor is placed in covers several time zones which may result in images of daily load peaks appearing in 1 hour intervals. Figure 4.7 contains two magnified intervals from this plot.

modulated grid frequency signal we added noise to a synthetic modulation signal. For most simulations we used measured frequency data gathered with our frequency sensor. We only have a limited amount of capture data. Re-using segments of this data as background noise in multiple simulation runs could lead to our simulation results depending on individual features of this particular capture that would be common between all runs. To estimate the impact of this problem we re-ran some of our simulations with artificial random noise synthesized with a power spectral density matching that of our capture. To do this, we first measured our capture's PSD, then fitted a low-resolution spline to the PSD curve in log-log coördinates. We then generated white noise, multiplied the resampled spline with the DFT of the synthetic noise and performed an iDFT on the result. The resulting time-domain signal is our synthetic grid frequency data. Figure 4.9 shows the PSD of our measured grid frequency signal. The red line indicates the low-resolution log-log spline interpolation used for shaping our artificial noise. Figure 4.10 shows the PSD of our simulated signal overlaid with the same spline as a red line and shows time-domain traces of both simulated (blue) and reference signals (orange) at various time scales. Visually both signals look very similar, suggesting that we have found a good synthetic approximation of our measurements.

In our simulations, we manipulated four main variables of our modulation scheme and demodulation algorithm and observed their impact on symbol error



(a) A 2 hour window centered on 00:00 UTC.



(b) A 2 hour window centered on 18:30 UTC.

Figure 4.7: Two magnified 2 hour windows of the trace from Figure 4.6.

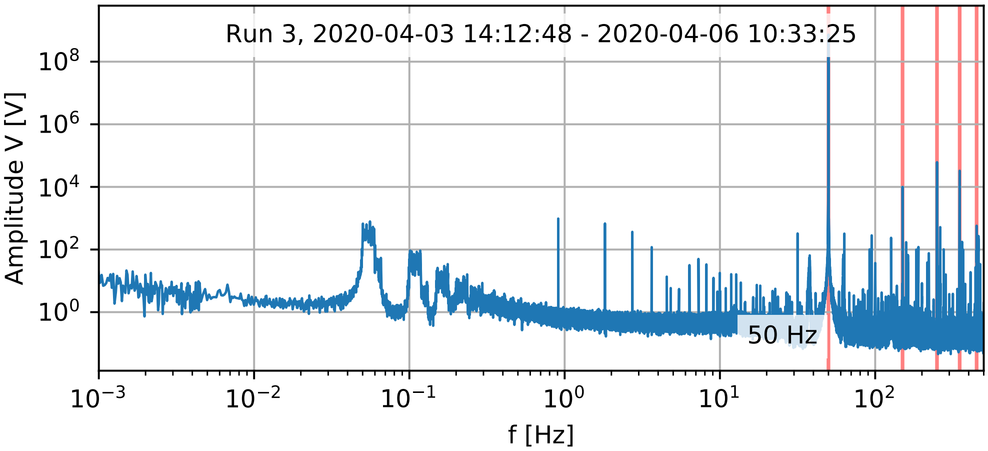


Figure 4.8: Power spectral density of the mains voltage trace in Figure 4.6. Data was captured using our frequency measurement sensor (4.1.2) and FFT-processed after applying a Blackman window. The vertical lines indicate 50 Hz and odd harmonics. We can see the expected peak at 50 Hz along with smaller peaks at odd harmonics. We can also see a number of spurious tones both between harmonics and at low frequencies. We can also see bands containing high noise energy around 0.1 Hz. This graph shows a high signal-to-noise ratio that is not very demanding on our frequency estimation algorithm.

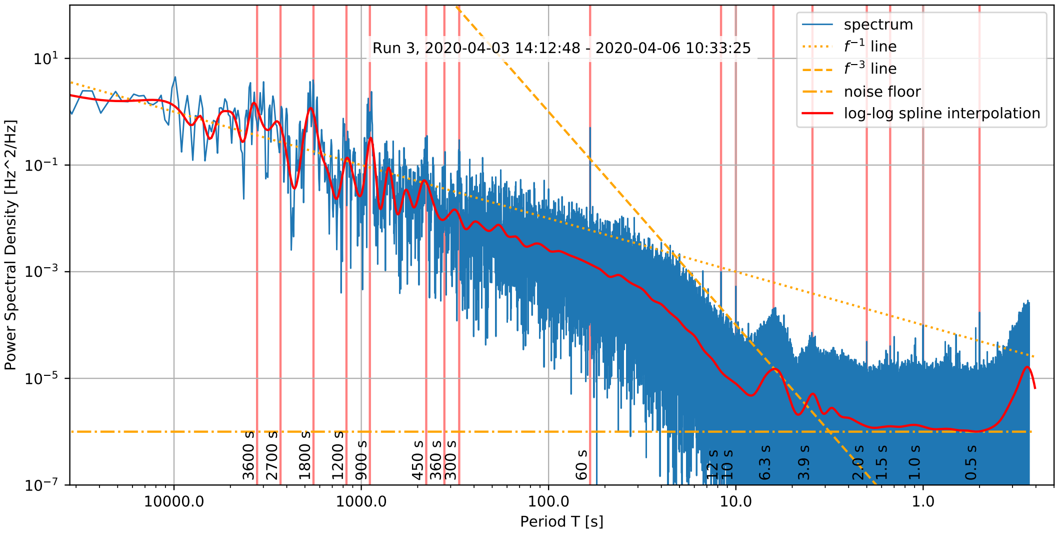


Figure 4.9: Power spectral density of the 24 hour grid frequency trace in Figure 4.6 with some notable peaks annotated with the corresponding period in seconds. The $\frac{1}{f}$ line indicates a pink noise spectrum. Around a period of 20 s the PSD starts to fall off at about $\frac{1}{f^3}$ until we can make out some bumps at periods around 2 and 3 s. Starting at at around 1 Hz we can see a white noise floor in the order of $\mu Hz^2/Hz$.

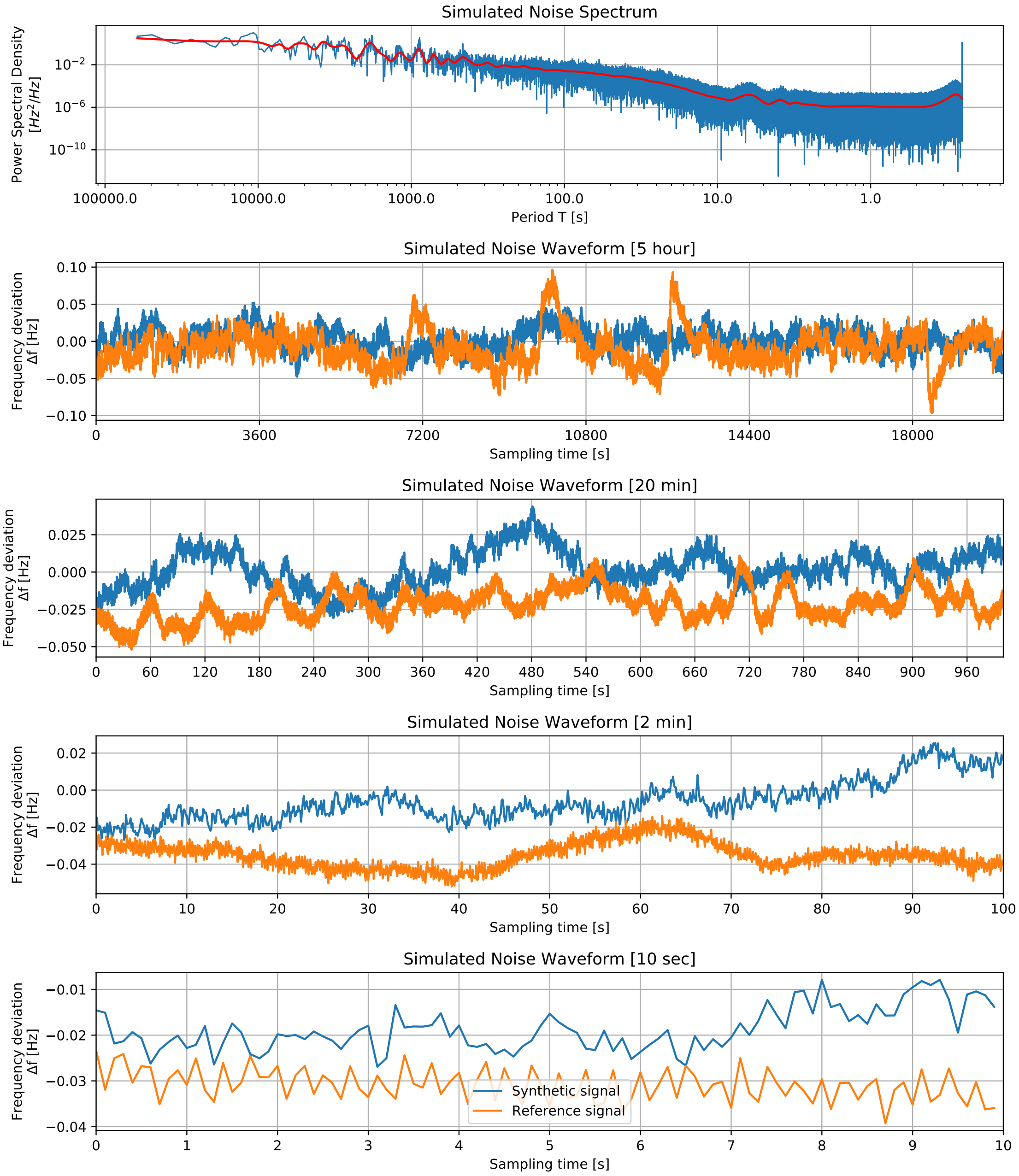


Figure 4.10: Synthetic grid frequency in comparison with measured data. The topmost graph shows the synthetic spectrum compared to the spline approximation of the measured spectrum (red line). The other graphs show time-domain synthetic data (blue) in comparison with measured data (orange).

rate (SER):

Modulation amplitude. Higher amplitude corresponds to a lower SER.

Modulation bit count. Higher bit count n means longer transmissions but yields higher theoretical decoding gain, and should increase demodulator sensitivity. Ultimately, we want to find a sweet spot of manageable transmission length at good demodulator sensitivity.

Decimation or DSSS chip duration. The chip time determines where in the grid frequency spectrum (Figure 4.9) our modulated signal is located. Given our noise spectrum (Figure 4.9) lower chip durations (shifting our signal upwards in the spectrum) should yield lower in-band background noise which should correspond to lower symbol error rates.

Demodulation correlator peak threshold factor. The first step of our prototype demodulation algorithm is to calculate the correlation between all $2^n + 1$ Gold sequences and our signal and to identify peaks corresponding to the input data containing a correctly aligned Gold sequence. The threshold factor determines peaks of which magnitude compared to baseline noise levels are considered in the following maximum likelihood estimation (MLE) decoding (cf. Figure 4.19).

Our results indicate that symbol error rate is a good proxy of demodulation performance. With decreasing signal-to-noise ratio, margins in various parts of the demodulator decrease which statistically leads to an increased symbol error rate. Our simulations yield smooth, reproducible SER curves with adequately low error bounds. This shows SER is related monotonically to the signal-to-noise margins inside our demodulator prototype.

4.2.1 Sensitivity as a function of sequence length

A basic parameter of our DSSS modulation is the length of the Gold codes used. The length of a Gold code is exponential in the code's bit count. Figure 4.11 shows a plot of the symbol error rate of our demodulator prototype depending on amplitude for each of five, six, seven and eight bit Gold sequences. In regions where symbol error rate is neither clipping at 0 nor at 1 we can see the expected dependency that a $n + 1$ bit Gold sequence at roughly twice the length yields roughly one half the SER. We can also observe a saturation effect: At low amplitudes, increasing the correlation length does not yield much benefit in SER anymore. In particular at a signal amplitude of 2.5 mHz even with asymptotically infinite sequence length our demodulator would still not be able to produce a good demodulation. This is likely due to numerical errors in our demodulator. Since Gold codes of more than 7 bit would yield unacceptably long transmission times this does not pose a problem in practice.

Figure 4.12 for each bit count shows the minimum signal amplitude at which our demodulator crossed below SER $= 0.5$. If we have sufficient transmitter power to allocate selecting either a 5 bit or a 6 bit Gold code yields sufficient performance at manageable data rates.

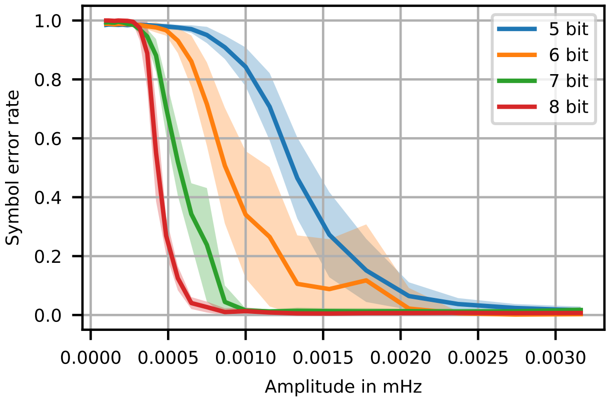


Figure 4.11: Symbol Error Rate (SER) as a function of transmission amplitude. The line represents the mean of several measurements for each parameter set. The shaded areas indicate one standard deviation from the mean. Background noise for each trial is a random segment of measured grid frequency. Background noise amplitude is the same for all trials. Shown are four traces for four different DSSS sequence lengths. Using a 5-bit gold code, one DSSS symbol measures 31 chips. 6 bit per symbol are 63 chips, 7 bit are 127 chips and 8 bit 255 chips. This simulation uses a decimation of 10, which corresponds to an 1s chip length at our 10Hz grid frequency sampling rate. At 5 bit per symbol, one symbol takes 31s and one bit takes 6.2s amortized. At 8 bit one symbol takes 255s = 4min15s and one bit takes 31.9s amortized. Here, slower transmission speed buys coding gain. All else being equal this allows for a decrease in transmission power.

4.2.2 Sensitivity versus peak detection threshold factor

One of the high level parameters of our demodulation algorithm is the *threshold factor*. This parameter is an implementation detail specific to our algorithm and not general to all possible DSSS demodulation algorithms. After correlating the input signal against the template Gold sequences our algorithm runs a single channel discrete wavelet transform (DWT) on the correlator output to better discriminate peaks from background noise. The output of this DWT is then normalized against a running average and then fed into a simple threshold detector. The threshold of this detector is our threshold factor. This threshold is the ratio that a correlation peak after DWT has to stand out from long-term average background noise to be considered a peak.

The threshold factor is an empirically determined unitless parameter. Low threshold factors yield many false positives that in the extreme ultimately overload our MLE estimator's capacity to discard them. Moderate numbers of false positives do not pose much of a challenge to our MLE since these spurious peaks have a random time distribution and are easily discarded by our MLE's detection of sequences of equally-spaced symbols. High threshold factors lead the algorithm to completely ignore some valid peaks. To some degree this can be compensated by our later interpolation step for missing peaks but in the extreme will also break demodulation. In our simulations good values lie in the range from 4.0 to 5.5.

Figure 4.13 contains plots of demodulator sensitivity like the one in Figure

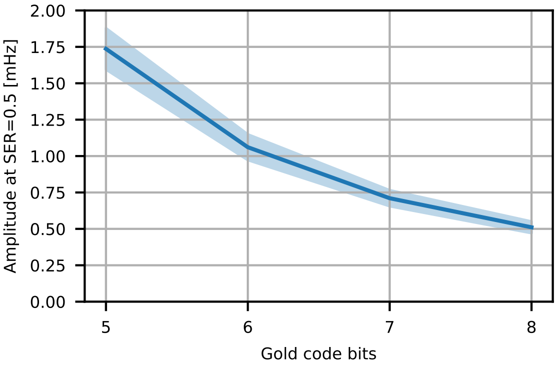


Figure 4.12: Amplitude at an SER of 0.5 in mHz depending on symbol length. Here we can observe an increase of sensitivity with increasing symbol length, but we can clearly see diminishing returns above 6 bit (63 chips). Considering that each bit roughly doubles overall transmission time for a given data length it seems lower bit counts are preferrable if the required transmitter power can be realized.

4.11. This time there is one color-coded trace for each threshold factor between 1.5 and 10.0 in steps of 0.5. We can see a clear dependency of demodulation performance from threshold factor with both very low and very high values breaking the demodulator. The runaway traces that we can see at low threshold factors are artifacts of an implementation issue with our prototype code. We later fixed this issue in the demonstrator firmware in Section 4.3.2. For comparison purposes this issue do not matter.

If we again look at the intercept points where the amplitude traces cross SER = 0.5 in these graphs we get the plots in Figure 4.14. From this we can conclude that the range between 4.0 and 5.0 will yield adequate threshold factors for our use case.

4.2.3 Chip duration and bandwidth

A parameter of any DSSS system is the frequency band used for transmission. Instead of specifying absolute frequencies in our simulations we expressed DSSS bandwidth through chip duration and Gold sequence length. In our prototype, chip duration is specified in grid frequency sampling periods to ease implementation without loss of generalization.

Figure 4.15 shows the dependence of symbol error rate at a fixed good threshold factor from chip duration. The color bars indicate both chip duration translated to seconds real-time and the resulting symbol duration at the given Gold code length. In the lower graphs we show the trace of amplitude at SER = 0.5 over chip duration like we did in Figure 4.14 for threshold factor. In both graphs we can see a faint optimum for very short chips with a decrease of sensitivity for long chips. This effect is due to longer chips moving the signal band into noisier spectral regions (cf. Figure 4.9).

In the previous graphs we have used random clips of measured grid frequency noise as noise in our simulations. Comparing between a simulation using measured noise and synthetic noise generated as we outlined in the beginning of Section 4.2 we get the plots in Figure 4.16. We can see that while not perfect our simulated noise is an adequate approximation of reality: Our prototype demodulator shows no significant difference in behavior between measured and

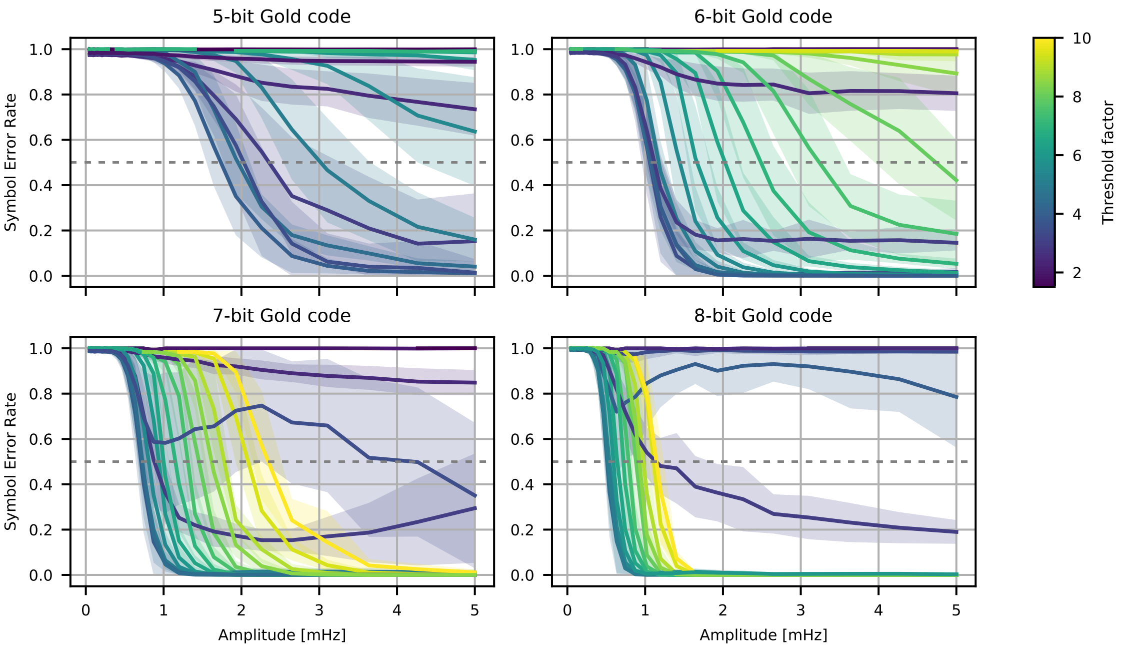


Figure 4.13: SER vs. amplitude graph similar to Figure 4.11 with one color-coded traces for threshold factors between 1.5 and 10.0. Each graph shows traces for a single DSSS symbol length.

simulated noise. Simulated noise causes slightly worse performance for long chips. Overall the results for both are very close in absolute value.

4.3 Implementation of a demonstrator unit

To demonstrate the viability of our reset architecture we decided to implement a demonstrator system. In this demonstrator we use JTAG to reset part of a commodity smart meter from an externally-connected reset controller. The reset controller receives its commands over the grid frequency modulation system we outlined in this thesis. To keep implementation cost low the reset controller is fed a simulation of a modulated grid frequency signal through a standard 3.5 mm audio jack[6]. Measurement of actual grid frequency instead would simply require a voltage divider and depending on the setup an analog optoisolator.

4.3.1 Selecting a smart meter for demonstration purposes

For our demonstrator to make sense we wanted to select a realistic reset target. In Germany where this thesis was written a standards-compliant setup would consist of a comparatively feature-limited smart meter and a smart meter

---

[6] By generously cutting two PCB traces the meter we chose to use can be easily modified to provide galvanic separation between grid and main application microcontroller. With this modification we have to supply power to its main application MCU externally along with the JTAG interface but now the modified meter is electrically safe.

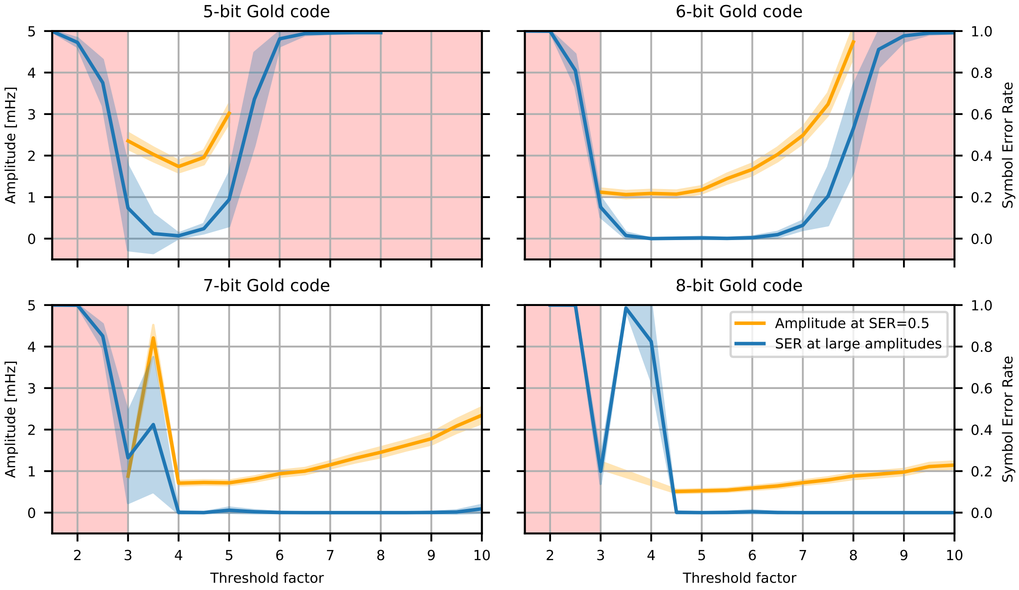
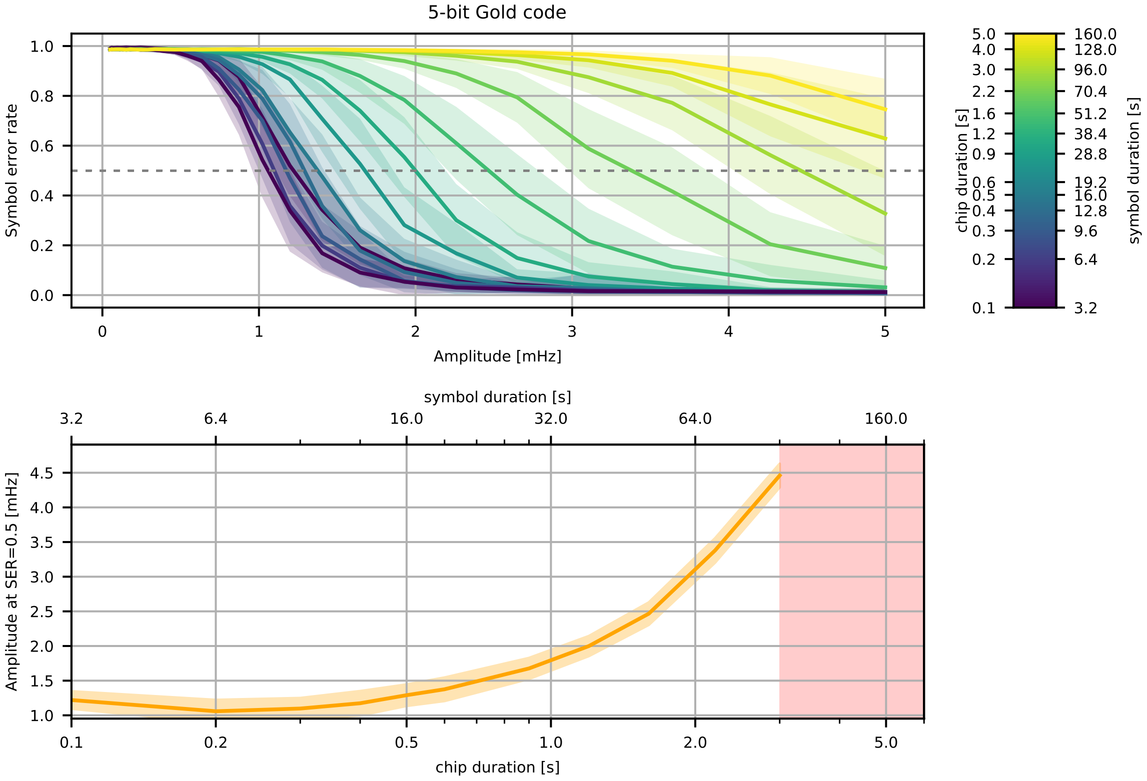


Figure 4.14: Graphs of amplitude at $SER = 0.5$ for each symbol length as well as asymptotic SER for large amplitudes. Areas shaded red indicate that $SER = 0.5$ was not reached for any amplitude in the simulated range. The bumps in the 7 bit and 8 bit graphs are due to the convergence problem we identified above and do not exist in our demonstrator implementation. We see that smaller symbol lengths favor lower threshold factors, and that optimal threshold factors for all symbol lengths are between 4.0 and 5.0.

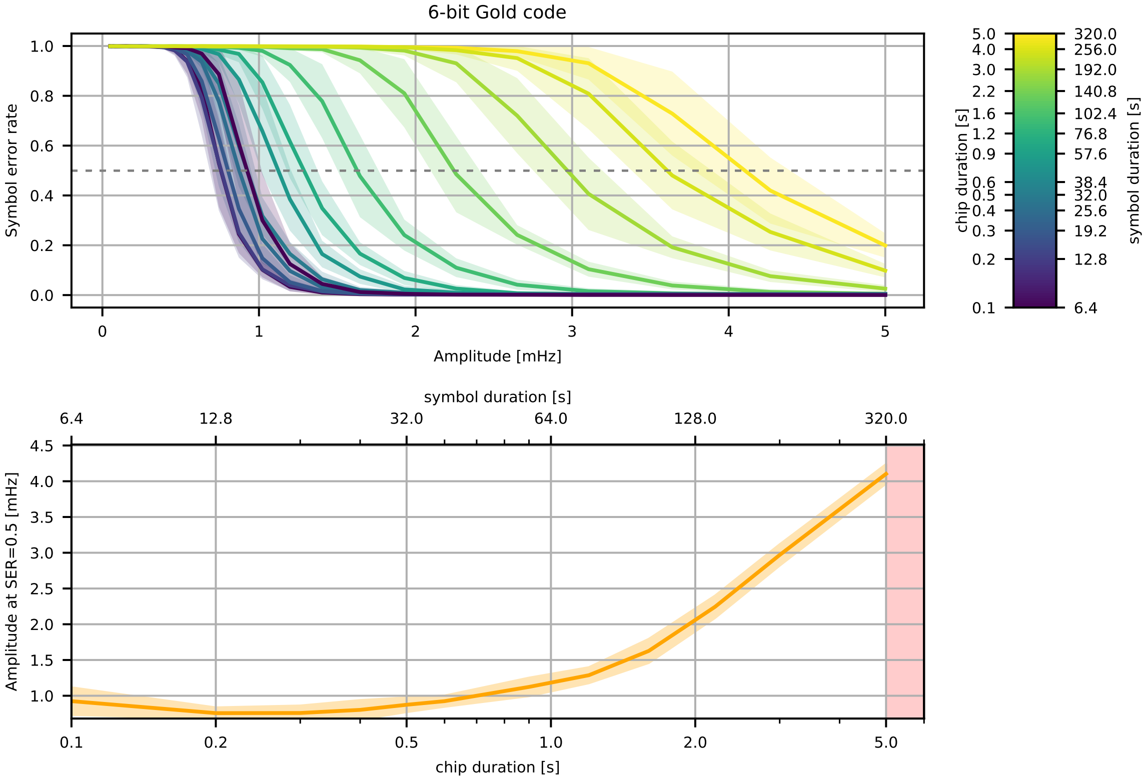
gateway (SMGW) containing all of the complex bidirectional protocol logic such as wireless or landline IP connectivity. The realistic target for a setup in this architecture would be the components of an SMGW such as its communication modem or main application processor. In the German architecture the smart meter does not even have to have a bi-directional data link to the SMGW effectively mitigating any attack vector for remote compromise.

Despite these considerations we still chose to reset the application MCU

Figure 4.15 *(following page)*: Dependence of demodulator sensitivity on DSSS chip duration. Due to computational constraints this simulation is limited to 5 bit and 6 bit DSSS sequences. There is a clearly visible sensitivity maximum at short chip lengths around 0.2s. Short chip durations shift the entire transmission band up in frequency. In Figure 4.9 we can see that noise energy is mostly concentrated at lower frequencies, so shifting our signal up in frequency will reduce the amount of noise the decoder sees behind the correlator by shifting the band of interest into a lower-noise spectral region. For a practical implementation chip duration is limited by physical factors such as the maximum modulation slew rate ($\frac{\mathrm{d}P}{\mathrm{d}t}$) that can be technically realized and the maximum Rate-Of-Change-Of-Frequency (ROCOF, $\frac{\mathrm{d}f}{\mathrm{d}t}$) that the grid can tolerate.



(a) 5 bit Gold code.



(b) 6 bit Gold code.

inside smart meter for two reasons. One is that SMGWs are much rarer on the second-hand market. The other is that SMGWs are a particular feature of the German standardization landscape and in many other countries functions of an SMGW such as wireless protocol handling are integrated into the meter itself (see e.g. [65]).

In the end we settled on a Q3DA1002 three phase 60A meter made by German manufacturer EasyMeter. This meter is typical of what would be found in an average German household and can be acquired very inexpensively as new old stock on online marketplaces.

The meter consists of a plastic enclosure with a transparent polycarbonate top part and a gray ABS bottom part that are ultrasonically welded together. In the bottom part of the case a PCB we call the *measurement* board is potted in epoxide resin (see Figure 4.17). This PCB contains three separate energy measurement ASICs for the three phases (see Figure 4.18). It also contains a capacitive dropper power supply for the meter circuitry and external modules such as a SMGW. The measurement board through three infrared links (one per phase) communicates with a smaller unpotted PCB we call the *display* board in the top of the case. This PCB handles measurement logging and aggregation, controls a small segment LCD displaying totals and handles the externally accessible kW h impulse LED and serial IR links.

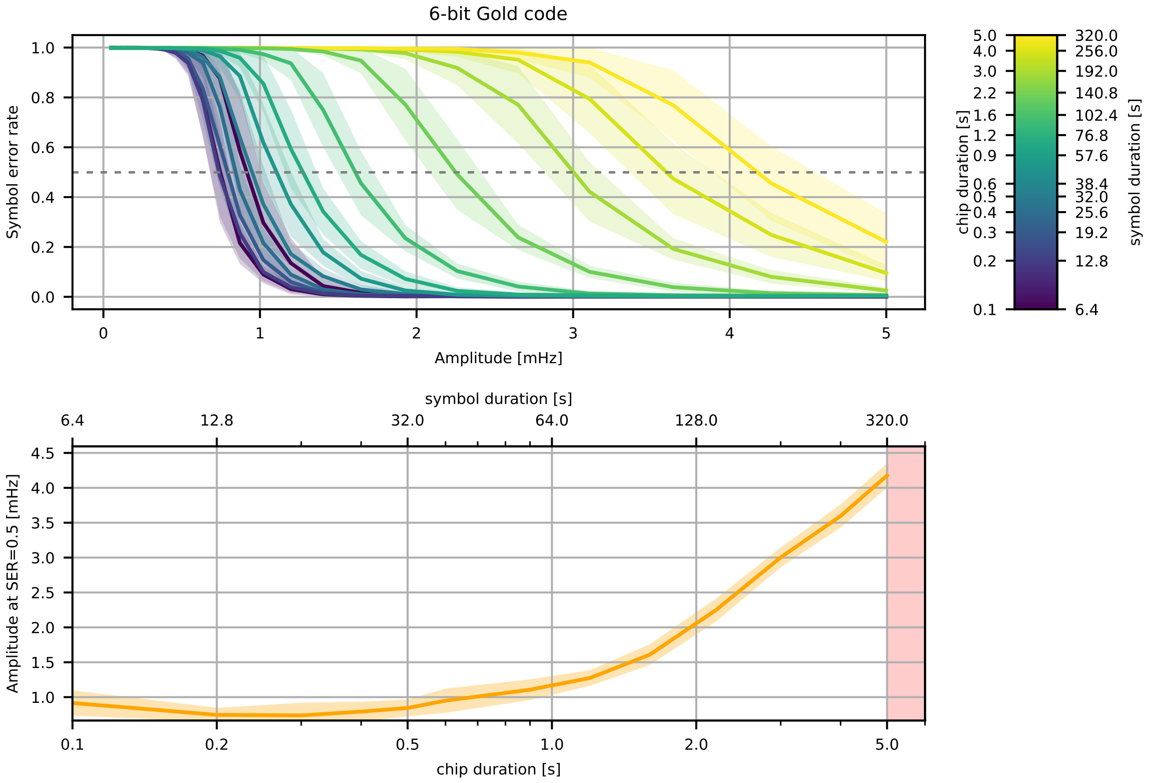
The measurement board does not contain any logging or outside communication interfaces. All of that is handled on the display board by a Texas Instruments `MSP430F2350` application MCU. This is a 16-bit RISC MCU with 16 kB flash and 2 kB SRAM[7]. There is an I2C EEPROM that is used in conjunction with the microcontroller's internal 256 B data flash to keep redundant copies of energy consumption aggregates. On the side of the display board there is a 14-pin header containing both a standard TI MSP430 JTAG pinout and a UART serial interface for debugging. Conveniently, the JTAG port was left enabled by fuse in our particular production unit.

We chose to use this `MSP430` series application MCU as our reset target. Though in this particular unit remote compromise is impossible due to a lack of bidirectional communication links some of its sister models do contain bidirectional communication links[41] making compromise through communication

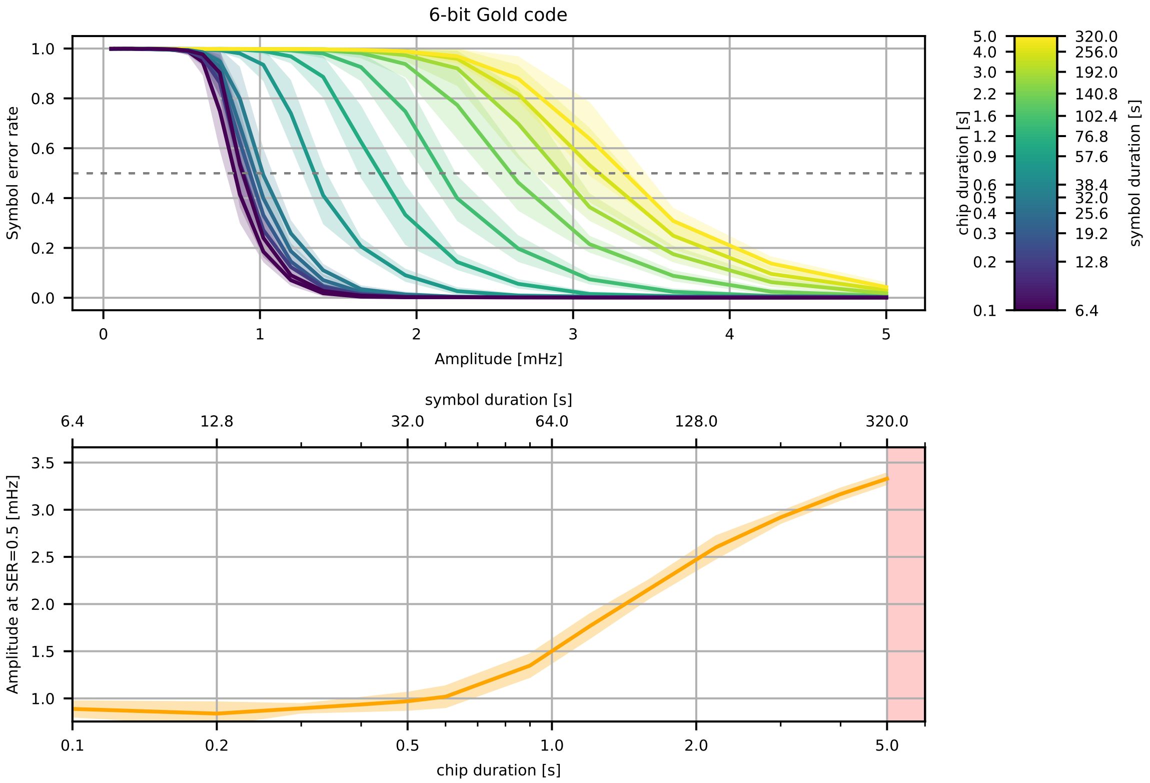
---

[7] At first glance the microcontroller might seem overkill for such a simple application, but most of its 16 kB program flash is in fact used. A casual glance with Ghidra shows that a large part of program flash is expended on keeping multiple redundant copies of energy consumption aggregates including error recovery in case of data corruption and some effort has even been made to guard against data corruption using simple non-cryptographic checksums. Another large part of the MCU's firmware handles data transmission over the meter's externally accessible IR link through Smart Message Language[18].

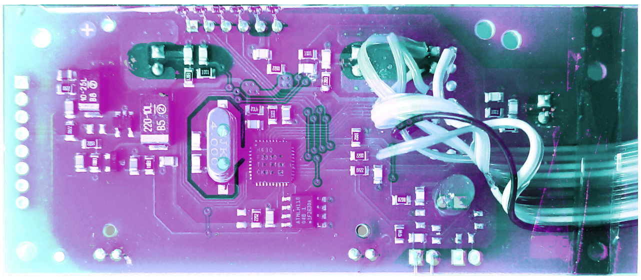
Figure 4.16 *(following page)*: Chip duration/sensitivity simulation results like in Figure 4.15 compared between a simulation using measured frequency data like in the previous graphs and one using artificially generated noise. There is little visible difference indicating that we have found a good model of reality in our noise synthesizer, but also that real grid frequency behaves like a frequency-shaped Gaussian noise process.



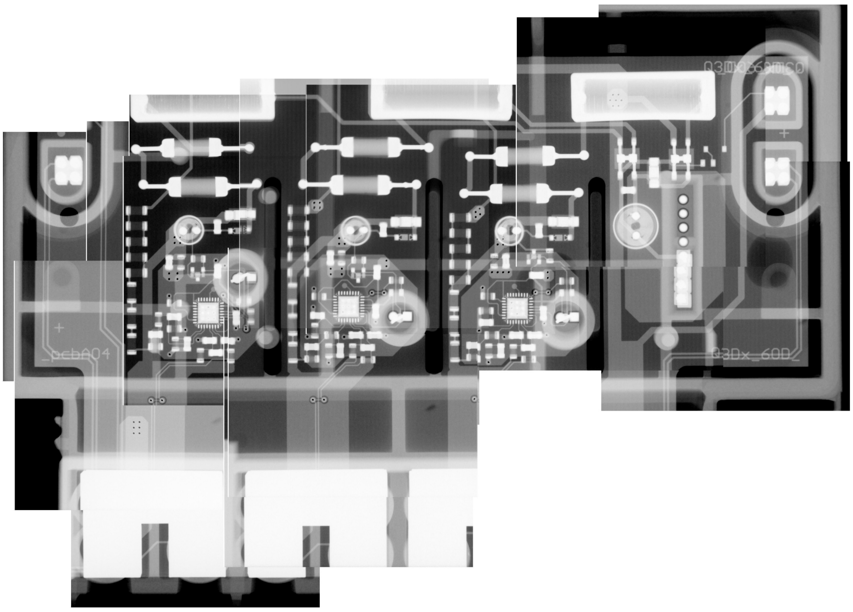
(a) Simulation using baseline frequency data from actual measurements.



(b) Simulation using synthetic frequency data.



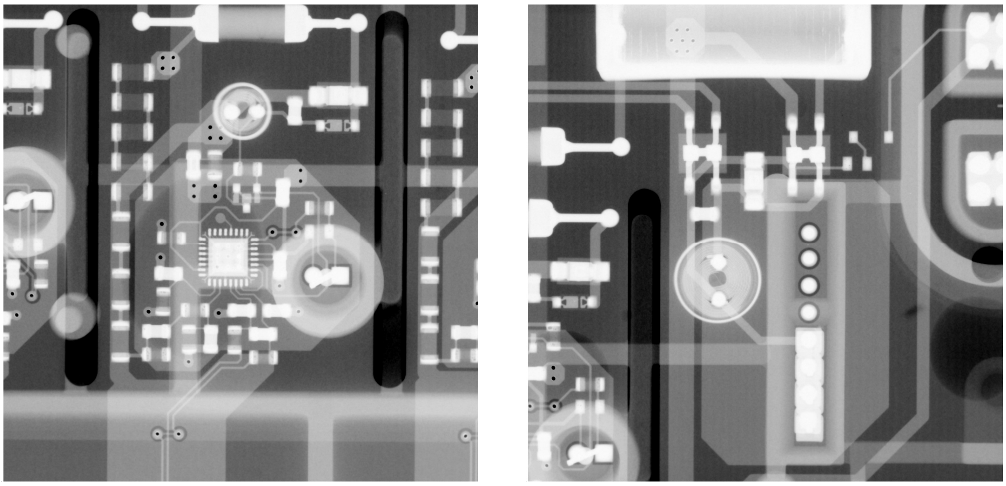
(a) Optical composite image of the display and data logging board in the top of the case. The six pins at the top are the SPI chip-on-glass segment LCD. Of the eight pads on the left six are unused and two carry the auxiliary power supply from the measurement board below. The bottom right section contains the kW h impulse LED and the angled IR communication LED. The flying wires connect to the 14-pin JTAG and serial debug header.



(b) Composite microfocus x-ray image of the potted measurement module in the bottom of the case. The ovals on the top left and right are power supply and data jumper connections for external modules such as SMGW interfaces. The bright parts at the bottom are the massive screw terminals with integrated current shunts. The circuitry right of the three independent measurement channels is the power supply circuit for the display board.

Figure 4.17: Composite images of the circuit boards inside the EasyMeter Q3DA1002 smart electricity meter used in our demonstration.

interfaces an at least theoretical possibility. In other countries, meters with a similar architecture to the Q3DA1002 include complex protocol logic as part of the meter itself or have bidirectional links to it[65, 36, 10, 68]. As an example, the Honeywell REX2 uses a Maxim Integrated `71M6541` main application microcontroller along with a Texas Instruments `CC1000` series radio transceiver and is advertised to support both over-the-air firmware upgrade and a remotely accessible disconnect switch.



(a) Microfocus x-ray of one channel's data acquisition circuit.

(b) Microfocus x-ray of the auxiliary power supply.

Figure 4.18: Microfocus x-rays of major sections of the EasyMeter Q3DA1002 measurement board.

4.3.2 Firmware implementation

We based our safety reset demonstrator firmware on the grid frequency sensor firmware we developed in Section 4.1.2. We implemented DSSS demodulation by translating the Python prototype code we developed in Section 4.2 to embedded C code. After validating the C translation in extensive simulations we integrated our code with a Reed-Solomon implementation and a libsodium-based implementation of the cryptographic protocol we designed in Section 3.3.4. To reprogram the target `MSP430` microcontroller we ported the low-level bitbang JTAG driver of `mspdebug`[8]. See Figure 4.19 for a schematic overview of signal processing in our demonstrator.

For all computation-heavy high level modules of our firmware such as the DSSS demodulator or the grid frequency estimator we wrote test fixtures that allow the same code that runs on the microcontroller to be executed on the host for testing. These test fixtures are very simple C programs that load input data from a file or the command line, run the algorithm and print results on standard output. To enable automatic testing of a large parameter set we run these test fixtures repeatedly from a set of Python scripts sweeping parameters.

4.4 Grid frequency modulation emulation

To emulate a modulated grid frequency signal we superimposed a DSSS-modulated signal at the proper amplitude with synthetic grid frequency noise generated according to the measurements we took in Section 4.1.2. In this primitive simulation we do not simulate the precise impulse response of the grid to a DSSS-modulated stimulus signal. Our results still serve to illustrate

---

[8] `https://github.com/dlbeer/mspdebug`

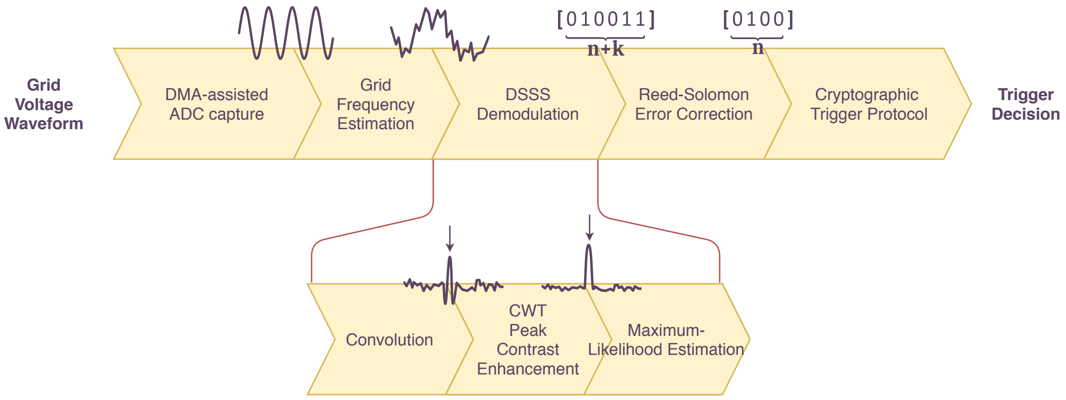

Grid Voltage Waveform
DMA-assisted ADC capture
Grid Frequency Estimation
DSSS Demodulation
[010011]
n+k
Reed-Solomon Error Correction
[0100]
n
Cryptographic Trigger Protocol
Trigger Decision
Convolution
CWT Peak Contrast Enhancement
Maximum-Likelihood Estimation

Figure 4.19: The signal processing chain of our demonstrator.

the possibility of data transmission in this manner this impulse response can be compensated for at the transmitter by selecting appropriate modulation parameters (e.g. chip rate and amplitude) and at the receiver by equalization with a matched filter.

4.5 Experimental results

After extensive simulations and testing of the individual modules of our solution we proceeded to conduct a real-world experiment. We tried the demonstrator setup in Figure 4.20 using an emulated noisy DSSS signal in real-time. Our experiment went without any issues and the firmware implementation correctly reset the demonstrator's meter. We were happy to see that our extensive testing paid off: The demonstrator setup worked on its first try.

Our experiment consisted of the demonstrator prototype with the meter flashed with its factory firmware connected to a microcontroller development board acting as the safety reset controller. The safety reset controller is connected to a laptop's audio output through an adapter board. The laptop plays back an emulated grid voltage waveform that the safety reset microcontroller measures and analyzes as it would when directly connected to the mains. When the microcontroller receives a reset sequence that is a valid signature using a development key incorporated into its firmware through JTAG it re-programs the smart meter with a modified firmware image that displays a success message on the meter's LCD.

We used a signature truncated at 120 bit in our experiment. We chose a 5 bit DSSS sequence. Taking the sign bit into account the length of the encoded signature is 20 DSSS symbols. On top of this we used Reed-Solomon error correction at a 2:1 ratio inflating total message length to 30 DSSS symbols. At the 1 s chip rate we used in other simulations as well this equates to an overall transmission duration of approximately 15 min. To give the demodulator some time to settle and to produce more realistic conditions of signal reception we padded the modulated signal unmodulated noise on both ends.

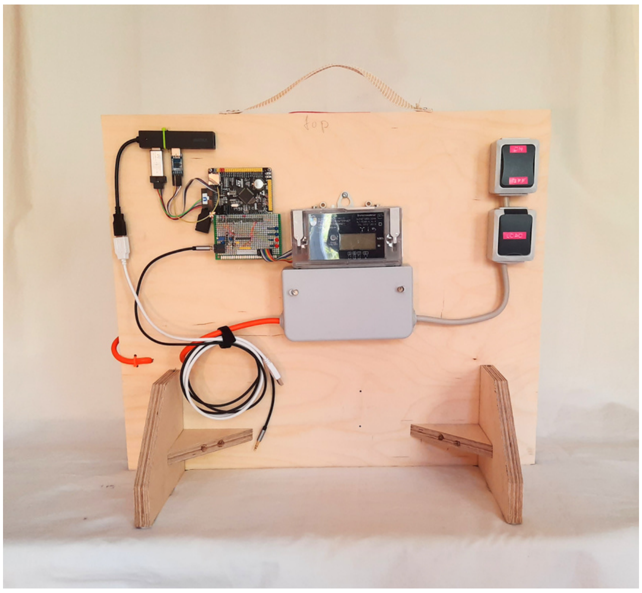
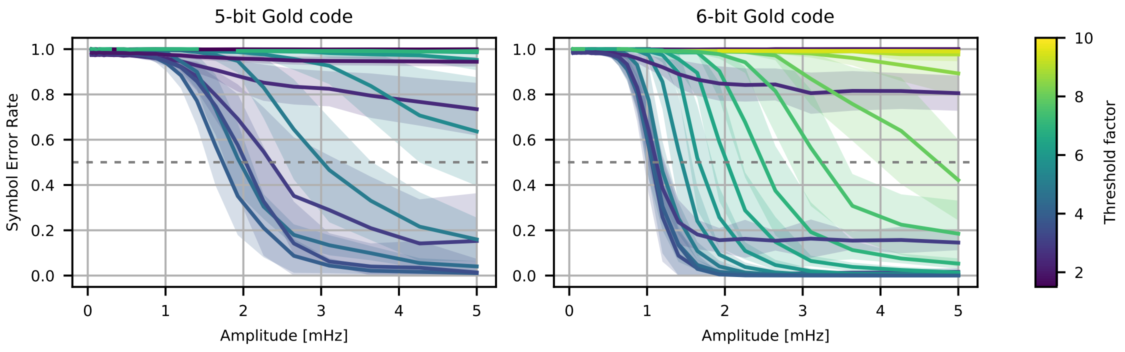


Figure 4.20: The completed prototype setup. The board on the left is the safety reset microcontroller. It is connected to the smart meter in the middle through an adapter board. The top left contains a USB hub with debug interfaces to the reset microcontroller. The cables on the bottom left are the debug USB cable and the 3.5 mm audio cable for the simulated mains voltage input.

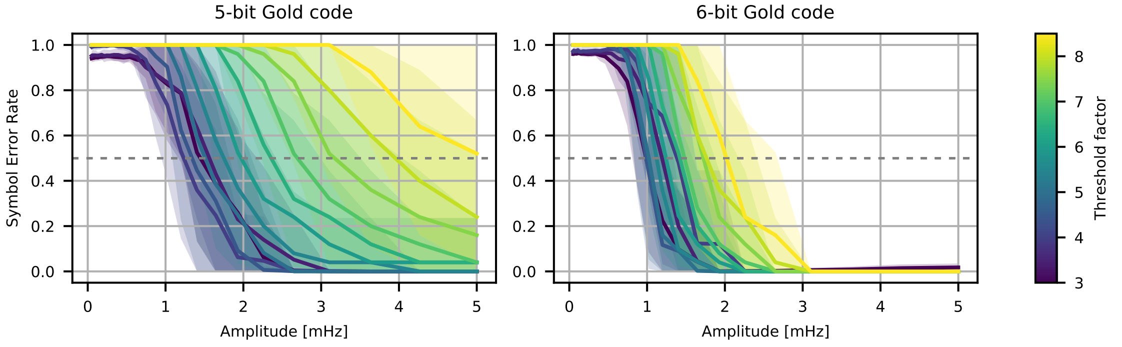
4.6 Lessons learned

Before settling on the commercial smart meter we first tried to use an `EVM430-F6779` smart meter evaluation kit made by Texas Instruments. This evaluation kit did not turn out well for two main reasons. One, it shipped with half the case missing and no cover for the terminal blocks. Because of this some work was required to get it electrically safe. Even after mounting it in an electrically safe manner the safety reset controller prototype would also have to be galvanically isolated to not pose an electrical safety risk since the main MCU is not isolated from the grid and the JTAG port is also galvanically coupled. The second issue we ran into was that the `EVM430-F6779` is based around an `MSP430F6779` microcontroller. This microcontroller is a rather large part within the `MSP430` series and uses a new revision of the CPU core and associated JTAG peripheral that are incompatible with all `MSP430` programmers we tried to use on it. `mspdebug` does not have support for it and porting TI's own JTAG programmer reference sources did not yield any results either. Finally we tried an USB-based programmer made by TI themselves that turned out to either have broken firmware or a hardware defect, leading to it frequently reënumerating on the USB.

Overall our initial assumption that a development kit would certainly be easier to program than a commercial meter did not prove to be true. Contrary to our expectations the commercial meter had JTAG enabled allowing us to easily read out its stock firmware without needing to reverse-engineer vendor firmware update files or circumventing code protection measures. The fact that its firmware was only available in its compiled binary form was not much of a



(a) Python prototype.



(b) Embedded C implementation.

Figure 4.21: Symbol error rate plots versus threshold factor for both our Python prototype (above) and our firmware implementation of our demodulation algorithm. Note the slightly different threshold factor color scales. Cf. Figure 4.13.

hindrance as it proved not to be too complex and all we wanted to know could be found out with just a few hours of digging in Ghidra.

In the firmware development phase our approach of testing every module individually (e.g. DSSS demodulator, Reed-Solomon decoder, grid frequency estimation) proved to be very useful. In particular debugging benefited greatly from being able to run several thousand tests within seconds. In case of our DSSS demodulator this modular testing and simulation architecture allowed us to simulate thousands of runs of our implementation on test data and directly compare it to our Jupyter/Python prototype (see Figure 4.21). Since we spent more time polishing our embedded C implementation it turned out to perform better than our Python prototype. At the same time it shows fundamentally similar response to its parameters. One significant bug we fixed in the embedded C version was the Python version's tendency towards incorrect decodings at even very large amplitudes.

In accordance with our initial estimations we did not run into any code space nor computation bottlenecks for chosing floating point emulation instead of porting over our algorithms to fixed point calculations. The extremely slow sampling rate of our systems makes even heavyweight processing such as FFT or our brute-force dynamic programming approach to DSSS demodulation possible well within our performance constraints.

Since we are only building a prototype we did not optimize firmware code size at all. The compiled code size of our firmware implementation is slightly larger than we would like at around 64 kB for our firmware image including everything except the target microcontroller firmware image. See appendix B for a graph illustrating the contribution of various parts of the signal processing toolchain to this total. Overall the most heavy-weight operations by far are the SHA512 implementation from libsodium and the FFT from ARM's CMSIS signal processing library. Especially the SHA512 implementation has large potential for size optimization because it is highly optimized for speed using extensive manual loop unrolling.

Chapter 5

Future work

5.1 Precise grid characterization

We based our simulations on a linear relationship between the generation/consumption power imbalance and grid frequency. Our literature study suggests that this is an appropriate first order approximation[30]. We kept the modulation bandwidth in our simulations inside a 1000 mHz to 100 mHz frequency band that we reason is most likely to exhibit this linear behavior in practice. At lower frequencies primary control kicks in. With the frequency delta thresholds specified for primary control systems[119] this would lead to significant non-linear effects. At higher frequencies grid frequency estimation at the receiver becomes more complex since the margins of the FFT transform shrink. Higher frequencies also come close to modes of mechanical oscillation in generators that usually lie at 5 Hz and above[30].

An analysis of the above concerns can be performed using dynamic grid simulation models[112, 44]. Presumably out of security concerns these models are only available under non-disclosure agreements. Integrating NDA-encumbered results stemming from such a model in an open-source publication such as this one poses a logistical challenge which is why we decided to leave this topic for a separate future work.

After detailed model simulation we ultimately aim to validate our results experimentally. Assuming linear grid behavior even under very small disturbances a small-scale experiment is an option. Such a small-scale experiment would require very long integration times: Given a frequency characteristic of $30\,\mathrm{GW\,Hz^{-1}}$ a stimulus of 10 kW yields $\Delta f = 0.33\,\mu\mathrm{Hz}$. At an estimated 20 mHz of RMS noise over a bandwidth of interest this results in an SNR slightly better than $-50\,\mathrm{dB}$. The correlation time necessary to offset this with DSSS processing gain at a chip rate of 1 Bd would be in the order of days. With such long correlation times clock stability starts to become a problem as during correlation transmitter and receiver must maintain close phase alignment with respect to one chip period. A phase difference requirement of less than 10°over this period of time would translate into clock stability better than 10 ppm. Though certainly not impossible to achieve this does pose an engineering challenge.

A way to reduce clock alignment might be to use grid frequency itself as a reference. Instead of keying the DSSS modulator/demodulator on a local crystal oscillator, chip timings would be described in fractions of a mains voltage cycle. This would track grid frequency variations synchronously at both ends and would maintain phase alignment even over long periods of time at cost of a slight increase in system complexity. The receiver would then measure differences between consecutive chips instead of their absolute values.

5.2 Technical standardization

The description of a safety reset system provided in this work could be translated into a formalized technical standard. Our system is simple compared to e.g. a full smart meter communication standard and thus can conceivably be described in a single, concise document. The complicated side of standardization would be the standardization of the backend operation including key management, coördination and command authorization.

5.3 Regulatory adoption

Since the proposed system adds significant cost and development overhead at no immediate benefit to either consumer or utility company it is unlikely that it would be adopted voluntarily. Market forces limit what long-term planning utility companies can do. An advanced mitigation such as this one might be out of their reach on their own and might require regulatory intervention to be implemented. To regulatory authorities a system such as this one provides a primitive to guard against attacks. Due to the low-level approach our system might allow a regulatory authority to restore meters to a safe state without the need of fine-grained control of implementation details such as application network protocols.

A regulatory authority might specify that all smart meters must use a standardized reset controller that on command resets to a minimal firmware image that disables external communication, continues basic billing functions and enables any disconnect switches. This system would enable the regulatory authority to directly preempt a large-scale attack irrespective of implementation details of the various smart meter implementations.

Cryptographic key management for the smart reset system is not much different to the management of highly privileged signing keys as they are used in many other systems such as TLS already. If the safety reset system is implemented by a regulatory authority they would likely be able to find a public entity that is already managing root keys for other government systems to also manage safety reset keys. Availability and security requirements of safety reset keys do not differ significantly from those for other types of root keys.

5.4 Zones of trust

In our design, we opted for a safety reset controller in form of a separate micocontroller entirely separate from whatever application microcontroller the smart meter design is already using. This design nicely separates the meter into an untrusted application on the core microcontroller and the trusted reset controller. Since the interface between the two is simple and one-way, it can be validated to a high standard of security.

Despite these security benefits, the cost of such a separate hardware device might prove high in a mass-market rollout. In this case, one might attempt to integrate the reset controller into the core microcontroller in some way. Primarily, there would be two ways to accomplish this. One is a solution that physically integrates an additional microcontroller core into the main application microcontroller package either as a module on the same die or as a separate die in a multi-chip module (MCM) with the main application microcontroller. A custom solution integrating both on a single die might be a viable path for very large-scale deployments but will most likely be too expensive in tooling costs alone to justify its use. More likely for a medium- to large-scale deployment of millions of meters would be a MCM integrating an off-the-shelf smart metering microcontroller die with the reset controller running on another, much smaller off-the-shelf microcontroller die. This solution might potentially save some cost compared to a solution using a discrete microcontroller for the reset controller.

The more likely approach to reducing cost overhead of the reset controller would be to employ virtualization technologies such as ARM's TrustZone in order to incorporate the reset controller firmware into the application firmware on the same processor core without compromising the reset controller's security or disturbing the application firmware's operation.

TrustZone is a virtualization technology that provides a hardware-assisted privileged execution domain. In traditional virtualization setups a privileged hypervisor is managing several unprivileged applications that share resources between them. Separation between applications in this setup is longitudinal between adjacent virtual machines. Two applications would both be running in unprivileged mode sharing the same CPU and the hypervisor would merely schedule them, configure hardware resource access and coördinate communication. This longitudinal virtualization simplifies application development since from the application's perspective the virtual machine looks very similar to a physical one. In addition, in general this setup can be used to reciprocally isolate two applications with neither one being able to gain control over the other.

In contrast to this, a TrustZone-like system in general does not provide several application virtual machines and longitudinal separation. Instead, it provides lateral separation between two domains: The unprivileged application firmware and a privileged hypervisor. Application firmware may communicate with the hypervisor through defined interfaces but due to TrustZone's design it need not even be aware of the hypervisor's existence. This makes a perfect fit for our reset controller. The reset controller firmware would be running in privileged mode and without exposing any communication interfaces to application firmware. The application firmware would be running in unprivileged mode without any modification. The main hurdles to the implementation to a system like this are the requirement for a microcontroller providing this type of virtualization on the one hand and the complexity of correctly employing this virtualization on the other hand. Virtualization systems such as TrustZone are still orders of magnitude more complex to correctly configure than it is to simply use separate hardware and secure the interfaces in between.

Chapter 6

Conclusion

In this thesis we have developed an end-to-end design of a reset system to restore smart meters to a safe operating state during an ongoing large-scale cyberattack. We have laid out the fundamentals of smart metering infrastructure and elaborated the need for an out of band method to reset a meter's firmware due to the large attack surface of this complex firmware. To allow our system to be triggered even in the middle of a cyberattack we have developed a broadcast data transmission system based on intentional modulation of the global grid frequency. We have developed the theoretical foundations of the process based on an established model of inertial grid frequency response to load variations and shown the viability of our end-to-end design through extensive simulations. To put these simulations on a solid foundation we have developed a grid frequency measurement methodology comprising of a custom-designed hardware device for electrically safe data capture and a set of software tools to archive and process captured data. Our simulations show good behavior of our broadcast communication system and give an indication that coöperating with a large consumer such as an aluminum smelter would be a feasible way to set up a transmitter with very low hardware overhead. Based on our broadcast primitive we have developed a cryptographic protocol ready for embedded implementation in resource-constrained systems that allows triggering all or a selected subset of devices within a quick response time of less than 30 minutes. Finally, we have experimentally validated our system using simulated grid frequency data in a demonstrator setup based on a commercial microcontroller as our safety reset controller and an off-the-shelf smart meter. We have laid out a path for further research and standardization related to our system. Our code and electronics designs are available at the public repository listed on the second page of this document.

Bibliography

[1] Asmaa Abdallah. *Security and Privacy in Smart Grid.* Ed. by Xuemin Shen. SpringerBriefs in Electrical and Computer Engineering. Cham: Springer International Publishing, 2018. 1 Online-Ressource (XIV, 126 p. 30 illus., 23 illus. in color). ISBN: 9783319936772.

[2] Lisa Alejandro et al. *Global Market for Smart Electricity Meters. Government Policies Driving Strong Growth.* Research rep. 2014.

[3] R. J. Anderson and S. J. Bezuidenhoudt. "On the reliability of electronic payment systems". In: *IEEE Transactions on Software Engineering* 22.5 (1996), pp. 294–301. DOI: `https://doi.org/10.1109/32.502222`.

[4] Ross J. Anderson. *Security engineering. A guide to building dependable distributed systems.* 3rd. Preview of upcoming edition. Wiley, 2020.

[5] Ross Anderson and Shailendra Fuloria. "Who controls the off switch?" In: *2010 First IEEE International Conference on Smart Grid Communications.* Gaithersburg, MD, 2010, pp. 96–101. DOI: `10.1109/SMARTGRID.2010.5622026`.

[6] Ross Anderson et al. "A New Family of Authentication Protocols". In: *ACM SIGOPS Operating Systems Review* (1998). DOI: `https://doi.org/10.1145/302350.302353`.

[7] Pol Van Aubel and Erik Poll. "Smart metering in the Netherlands: what, how and why". In: *International Journal of Electrical Power and Energy Systems* 109 (2019), pp. 719–725. ISSN: 0142-0615. DOI: `https://doi.org/10.1016/j.ijepes.2019.01.001`.

[8] Mohammed W. Ayoub and Francis V. P. Robinson. *A comparative study between diode and thyristor based AC to DC converters for aluminium smelting process.* 2013. DOI: `https://doi.org/10.1109/IEEEGCC.2013.6705851`.

[9] Daniel Belega and Dario Petri. "Accuracy Analysis of the Multicycle Synchrophasor Estimator Provided by the Interpolated DFT Algorithm". In: *IEEE Transactions on Instrumentation and Measurement* 62 (5 2013), pp. 942–953. ISSN: 0018-9456. DOI: `10.1109/tim.2012.2236777`.

[10] bigclivedotcom. *Inside a smart meter, and the REAL problem with them.* Oct. 26, 2018. URL: `https://www.youtube.com/watch?v=G32NYQpvy8Q` (visited on 06/03/2020).

[11] Matt Blaze et al. “The role of trust management in distributed systems security”. In: *Secure Internet Programming.* Springer, 1999, pp. 185–210.

[12] Shekhar Borkar. “Designing reliable systems from unreliable components: the challenges of transistor variability and degradation”. In: *IEEE Micro* 25.6 (2005), pp. 10–16.

[13] Jozef Borkowski, Dariusz Kania, and Janusz Mroczka. “Interpolated-DFT-Based Fast and Accurate Frequency Estimation for the Control of Power”. In: *IEEE Transactions on Industrial Electronics* 61 (12 2014), pp. 7026–7034. ISSN: 0278-0046. DOI: `10.1109/tie.2014.2316225`.

[14] Stuart Borlase, ed. *Smart Grids: Advanced Technologies and Solutions.* Electric Power and Energy Engineering. CRC Press, 2017. ISBN: 978-1-4987-9955-3.

[15] *Branchenempfehlung Strommarkt Schweiz Handbuch Smart Metering CH.* 2010.

[16] Marilyn A. Brown and Shan Zhou. “Smart-Grid Policies: An International Review. The Large-scale Renewable Energy Integration Challenge”. In: *Advances in Energy Systems: The Large-scale Renewable Energy Integration Challenge.* First Ed. Wiley, 2019. DOI: `10.1002/9781119508311`.

[17] Bundesamt für Sicherheit in der Informationstechnik. *Technische Richtlinie BSI TR-03109.* Bundesamt für Sicherheit in der Informationstechnik, Nov. 2015.

[18] Bundesamt für Sicherheit in der Informationstechnik. *TR-03109-1 Anlage IV: Feinspezifikation "Drahtgebundene LMN-Schnittstelle" Teil b: "SML Smart Message Language".* Bundesamt für Sicherheit in der Informationstechnik, Mar. 2013.

[19] Bundesamt für Sicherheit in der Informationstechnik. *TR-03109-1: Anforderungen an die Interoperabilität der Kommunikationseinheit eines intelligenten Messsystems.* Bundesamt für Sicherheit in der Informationstechnik, Jan. 2019.

[20] Bundesamt für Sicherheit in der Informationstechnik. *TR-03109-2 Anhang A: Smart Meter Gateway Sicherheitsmodul Use Cases.* Bundesamt für Sicherheit in der Informationstechnik, Dec. 2014.

[21] Bundesamt für Sicherheit in der Informationstechnik. *TR-03109-2: Smart Meter Gateway - Anforderungen an die Funktionalität und Interoperabilität des Sicherheitsmoduls.* Bundesamt für Sicherheit in der Informationstechnik, Dec. 2014.

[22] Bundesamt für Sicherheit in der Informationstechnik and Bundesministerium für Wirtschaft und Energie. *Standardisierungsstrategie zur sektorübergreifenden Digitalisierung nach dem Gesetz zur Digitalisierung der Energiewende.* Jan. 2019. URL: `https://web.archive.org/web/20190919100713/https://www.bmwi.de/Redaktion/DE/Downloads/S-T/standardisierungsstrategie.pdf` (visited on 09/19/2019).

[23] Bundesministerium für Wirtschaft und Energie. *Häufig gestellte Fragen rund um Smart Meter.* Apr. 14, 2020. URL: `https://www.bmwi.de/Redaktion/DE/FAQ/Smart-Meter/faq-smart-meter.html` (visited on 05/20/2020).

[24] Bundesministerium für Wirtschaft und Energie and Ernst and Young. *Kosten-Nutzen-Analyse für einen flächendeckenden Einsatz intelligenter Zähler.* 2013. URL: `https://www.bmwi.de/Redaktion/DE/Publikationen/Studien/kosten-nutzen-analyse-fuer-flaechendeckenden-einsatz-intelligenterzaehler.pdf` (visited on 05/12/2020).

[25] Stuart Cheshire and Mary Baker. "Consistent overhead Byte stuffing". In: *IEEE/ACM Trans. Netw.* 7.2 (1999), pp. 159–172. DOI: `10.1109/90.769765`.

[26] Frances M. Cleveland. "Cyber security issues for advanced metering infrasttructure (AMI)". In: *2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century.* IEEE. 2008, pp. 1–5.

[27] Liviu Constantinescu-Simon, ed. *Handbuch Elektrische Energietechnik.* 1997. DOI: `10.1007/978-3-322-85061-4`.

[28] Enrico Costanza et al. "Doing the Laundry with Agents: a Field Trial of a Future SmartEnergy System in the Home. a field trial of a future smart energy system in the home". In: *CHI 2014, One of a CHInd.* 2014. DOI: `10.1145/2556288.2557167`.

[29] Valentin Crastan. *Elektrische Energieversorgung 1.* 2015. DOI: `10.1007/978-3-662-45985-0`.

[30] Valentin Crastan. *Elektrische Energieversorgung 3.* 2012. ISBN: 978-3-642-20099-1. DOI: `10.1007/978-3-642-20100-4`.

[31] Colette Cuijpers and Bert-Jaap Koops. "Smart metering and privacy in Europe: lessons from the Dutch case". In: *European data protection. Coming of age* (2012), pp. 269–293. DOI: `https://doi.org/10.1007/978-94-007-5170-5_12`.

[32] R. Czechowski and A. M. Kosek. "The most frequent energy theft techniques and hazards in present power energy consumption". In: *2016 Joint Workshop on Cyber- Physical Security and Resilience in Smart Grids (CPSR-SG).* IEEE. Apr. 2016, pp. 1–7. DOI: `10.1109/CPSRSG.2016.7684098`.

[33] Asja Derviškadić, Paolo Romano, and Mario Paolone. "Iterative-interpolated DFT for synchrophasor estimation: A single algorithm for P-and M-class compliant PMUs". In: *IEEE Transactions on Instrumentation and Measurement* 67.3 (2017), pp. 547–558.

[34] Statistisches Bundesamt DeStatis, ed. *Erzeugung - Bilanz - Monatsbericht über die Elektrizitätsversorgung.* Mar. 6, 2020. URL: `https://www.destatis.de/DE/Themen/Branchen-Unternehmen/Energie/Erzeugung/Tabellen/bilanz-elektrizitaetsversorgung.html` (visited on 05/07/2020).

[35] Whitfield Diffie and Martin Hellman. “New directions in cryptography”. In: *IEEE transactions on Information Theory* 22.6 (2976), pp. 644–654.

[36] Miro Djuric. *Elster REX2 Smart Meter Teardown.* iFixit. 2011. URL: `https://www.ifixit.com/News/14306/elster-rex2-smart-meter-teardown` (visited on 05/06/2020).

[37] Chris Dods, Nigel P Smart, and Martijn Stam. “Hash based digital signature schemes”. In: *Post-Quantum Cryptography.* Ed. by D. J. Bernstein, J. Buchmann, and E. Dahmen. Springer, 2009, pp. 96–115. ISBN: 978-3-540-88701-0. DOI: `https://doi.org/10.1007/978-3-540-88702-7_3`.

[38] Roman Düssel. “Paradigm Shift in the Indication of a Stable Cell during Power Modulation”. In: *Proceedings of the 36th International ICSOBA Conference.* 2018.

[39] *Dutch Smart Meter Requirements P3 Companion Standard.* Version 4.0.7. 2014.

[40] Dacfey Dzung, Inigo Berganza, and Alberto Sendin. “Evolution of powerline communications for smart distribution: From Ripple Control to OFDM”. In: *2011 IEEE International Symposium on Power Line Communications and Its Applications* (2011). DOI: `10.1109/ISPLC.2011.5764444`.

[41] EasyMeter GmbH. *Datenblatt Moderne Messeinrichtung Q3A Drehstromzähler.* 2020.

[42] Christian Egenhofer et al. *Composition and Drivers of Energy Prices and Costs: Case Studies in SelectedEnergy Intensive Industries – 2018.* 2018. DOI: `10.2873/937326`.

[43] David Eisma and Pretesh Patel. “Challenges in Power Modulation”. In: *Essential Readings in Light Metals, Volume 2, Aluminum Reduction Technology.* Ed. by Geoff Bearne, Marc Dupuis, and Gary Tarcy. 2016, pp. 683–688.

[44] ENTSO-E. *ENTSO-E Initial Dynamic Model of Continental Europe.* 2019. URL: `https://www.entsoe.eu/publications/system-operations-reports/#entso-e-initial-dynamic-model-of-continental-europe` (visited on 05/14/2020).

[45] ENTSO-E System Protection Dynamics and WG. *Oscillation Event 03.12.2017.* Mar. 2018.

[46] ENTSO-E Working Group Incident Classification Scale Under System Operations Committee. *Incidents Classification Methodology.* 2014.

[47] Michael J. Fell et al. “Public acceptability of domestic demand-side response in Great Britain: The role of automation and direct load control”. In: *Energy Research and Social Science* 9 (2015), pp. 72–84. ISSN: 2214-6296. DOI: `10.1016/j.erss.2015.08.023`.

[48] Daniel Fraunholz, Simon Duque Anton, and Hans Dieter Schotten. “Introducing GAMfIS: A Generic Attacker Model for Information Security”. In: IEEE, Nov. 2017.

[49] Jochen Fritz and Alexander Hovi. *Transkommando-System.* 2020. URL: `http://www.rundsteuerung.de/entwicklung/transkommando.html`.

[50] M Gasior and JL Gonzalez. *Improving FFT frequency measurement resolution by parabolic and gaussian interpolation.* 2004.

[51] Marek Gasior. "Improving frequency resolution of discrete spectra: algorithms of three-node interpolation". 2006.

[52] Daphne Geelen et al. "The use of apps to promote energy saving: a study of smartmeter–related feedback in the Netherlands". In: *Energy Efficiency* (12 2019). DOI: `https://doi.org/10.1007/s12053-019-09777-z`.

[53] Martin Georgiev et al. "The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software". In: *ACM Conference on Computer and Communications Security.* 2012, pp. 38–49.

[54] German Government Bundesnetzagentur. *Monitoring Report 2018.* 2018.

[55] Alois M. J. Goiser. *Handbuch der Spread-Spectrum Technik.* Springer, 1998. ISBN: 3-211-83080-4.

[56] E. Grebe et al. "Low Frequency Oscillations in the Interconnected System of Continental Europe". In: IEEE, Aug. 2010. DOI: `10.1109/PES.2010.5589932{\textperiodcentered}`.

[57] Ulrich Greveler et al. "Multimedia Content Identification Through SmartMeter Power Usage Profiles". In: *Computers, Privacy and Data Protection* (2012).

[58] Vehbi C. Güngör et al. "Smart Grid Technologies: Communication Technologies and Standards". In: *IEEE Transactions on Industrial Informatics* 7.4 (Nov. 2011), pp. 529–539.

[59] *Gutachten Digitalisierung der Energiewende.* 2019. URL: `https://www.bmwi.de/Redaktion/DE/Publikationen/Studien/digitalisierung-der-energiewende-thema-1.pdf?__blob=publicationFile&v=4`.

[60] Hager Group. *Hager Smart Meter EHZ363 Betriebsanleitung.* 2017. URL: `https://bnnetze.de/downloads/kunden/netzkunden/messstellenbetrieb-und-messung/funktionalitaet/hager-ehz363-betriebsanleitung.pdf` (visited on 05/11/2020).

[61] Neil M. Haller. "The S/KEY One-Time Password System". In: *Symposium on Network and Distributed System Security.* 1994, pp. 151–157.

[62] Neil M. Haller. *The S/KEY One-Time Password System.* RFC 1760. RFC Editor, 1995. DOI: `https://dx.doi.org/10.17487/RFC1760`.

[63] Ignacio Fernández Hernández. *Increasing Digital Tachograph Resilience: Galileo Open Service Navigation Message Authentication.* Ed. by European Commission. 2019. URL: `https://ec.europa.eu/transparency/regexpert/?do=groupDetail.groupMeetingDoc&docid=36951` (visited on 06/08/2020).

[64] Martin Holland. *Cambridge Analytica: Mehrere Untersuchungen angekündigt, mögliche Billionenstrafe für Facebook.* Ed. by Heise Online. Mar. 19, 2018. URL: `https://www.heise.de/newsticker/meldung/Cambridge-Analytica-Mehrere-Untersuchungen-angekuendigt-moegliche-Billionenstrafe-fuer-Facebook-3998151.html`.

[65] Honeywell Smart Energy. *Datasheet Honeywell REX2 smart meter.* 2017.

[66] Mitsuhide Ishima, Kiyoyuki Terai, and Yoshihiro Ogita. *Construction and Operation of Communication System for Smart Meter System of TEPCO Power Grid, Inc.* 2018.

[67] Itron Inc. *Benutzerhandbuch Smart Meter EM 214.* 2012. URL: `https://www.ewh.de/fileadmin/user_upload/Stromnetz/Zaehlerstaende/Produktbeschreibung_ITRON_EM214.pdf` (visited on 05/11/2020).

[68] Dave Jones. *EEVblog 409 - EDMI - Smart Meter Teardown.* Jan. 8, 2013. URL: `https://www.youtube.com/watch?v=dm-yZ1N3xmc` (visited on 06/03/2020).

[69] Yasin Kabalci. "A survey on smart metering and smart grid communication". In: *Renewable and Sustainable Energy Reviews* 57 (2016), pp. 302–318. DOI: `10.1016/j.rser.2015.12.114`.

[70] Kamstrup A/S. *STS prepayment meter.* URL: `https://www.kamstrup.com/en-en/electricity-solutions/smart-electricity-meters/sts-prepayment-meter` (visited on 05/11/2020).

[71] Uri Kanonov and Avishai Wool. "Secure containers in Android: the Samsung KNOX case study". In: *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices.* 2016, pp. 3–12.

[72] Abraham Kaplan. "The Law of the Instrument". In: *The Conduct of Inquiry: Methodology for Behavioral Science.* San Francisco: Chandler Publishing Co., 1964, p. 28. ISBN: 9781412836296.

[73] Auguste Kerckhoff. "La cryptographie militaire". In: *Journal des sciences militaires* IX (1883), pp. 5–38.

[74] Himanshu Khurana et al. "Smart-grid security issues". In: *IEEE Security and Privacy Magazine* 8 (2010), pp. 81–85. ISSN: 1540-7993. DOI: `10.1109/msp.2010.49`.

[75] Jinsub Kim and Lang Tong. "On Topology Attack of a Smart Grid: Undetectable Attacks and Countermeasures". In: *IEEE Journal on Selected Areas in Communications* 31.7 (July 2013). DOI: `10.1109/JSAC.2013.130712`.

[76] Thomas Kluyver et al. "Jupyter Notebooks - a publishing format for reproducible computational workflows". In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas, 20th International Conference on Electronic Publishing, Göttingen, Germany, June 7-9, 2016.* Ed. by Fernando Loizides and Birgit Schmidt. IOS Press, 2016, pp. 87–90. DOI: `10.3233/978-1-61499-649-1-87`.

[77] Oliver Kosut et al. “Malicious Data Attacks on the Smart Grid”. In: *IEEE Transactions on Smart Grid* 2.4 (Nov. 2011), pp. 645–658.

[78] Stefan Krempl. *36C3: Schwere Sicherheitslücken in Kraftwerken.* Ed. by Heise Online. Dec. 29, 2019. URL: `https://www.heise.de/newsticker/meldung/36C3-Schwere-Sicherheitsluecken-in-Kraftwerken-4624529.html` (visited on 05/22/2020).

[79] Prabha Kundur. *Power system stability and control.* eng. The EPRI power system engineering series. New York, NY u.a.: McGraw-Hill, 1994. ISBN: 007035958X.

[80] Leslie Lamport. *Constructing digital signatures from a one-way function.* 1979.

[81] Leslie Lamport, Robert Shostak, and Marshall Pease. “The Byzantine Generals Problem”. In: *ACM Transactions on Programming Languages and Systems* 4.3 (July 1982), pp. 382–401.

[82] Landis+Gyr Group AG. *Landis+Gyr Annual Report 2019.* May 28, 2020.

[83] Landis+Gyr Group AG. *Landis+Gyr Financial Report 2019.* May 6, 2020.

[84] Robert M. Lee, Michael J. Assante, and Tim Conway. “Analysis of the cyber attack on the Ukrainian power grid”. In: *Electricity Information Sharing and Analysis Center (E-ISAC)* (2016).

[85] Nancy G. Leveson and Clark S. Turner. “An Investigation of the Therac-25 Accidents”. In: *IEEE Computer* 26.7 (July 1993), pp. 18–41.

[86] Jaime Lloret et al. “An Integrated IoT Architecture for Smart Metering”. In: *IEEE Communications Magazine* 54 (2016), pp. 50–57.

[87] Deborah Lupton. “The diverse domains of quantified selves: self-tracking modes and dataveillance”. In: *Economy and Society* 45 (2016), pp. 101–122. ISSN: 0308-5147. DOI: `10.1080/03085147.2016.1143726`.

[88] David J. C. MacKay. *Information theory, inference, and learning algorithms.* Repr. with corr. Literaturverz. S. 613 - 619. Cambridge [u.a.]: Univ. Press, 2005. XII, 628. ISBN: 0521642981.

[89] Peter Mahlknecht. “Diplomarbeit Sicherheitsmodul für ein Smart Metering Gateway”. Technische Universität Wien, 2014.

[90] Anzar Mahmood, Nadeem Javaid, and Sohail Razzaq. “A review of wireless communications for smart grid”. In: *Renewable and Sustainable Energy Reviews* 41 (2015), pp. 248–260. DOI: `10.1016/j.rser.2014.08.036`.

[91] Ralph C Merkle. “A certified digital signature”. In: *Conference on the Theory and Application of Cryptology.* Springer. 1989, pp. 218–238.

[92] Hermann Merz, Thomas Hansemann, and Christof Hübner. *Building automation. Communication systems with EIB/KNX, LON, and BACnet.* Springer series on signals and communication technology. Berlin [u.a.]: Springer, 2009. X, 282. ISBN: 9783540888284.

[93] Yilin Mo et al. “Cyber-Physical Security of a Smart Grid Infrastructure”. In: *Proceedings of the IEEE* 100.1 (Jan. 2012), pp. 195–209.

[94] Vivek Mohan and Silicon Labs. *An Introduction to Wireless M-Bus*. 2015. URL: `http://pages.silabs.com/rs/634-SLU-379/images/introduction-to-wireless-mbus.pdf`.

[95] Andrés Molina-Markham et al. “Private Memoirs of a Smart Meter”. In: *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building* (2010). Title from The ACM Digital Library.

[96] Claudio Narduzzi et al. “Fast-TFM—Multifrequency phasor measurement for distribution networks”. In: *IEEE Transactions on Instrumentation and Measurement* 67.8 (2018), pp. 1825–1835.

[97] National Semiconductor. *Clock Conditioner Owner’s Manual.* 2006. URL: `http://www.ti.com/lit/ug/snaa103/snaa103.pdf?ts=1591194443306`.

[98] Dieter Nelles and Christian Tuttas. *Elektrische Energietechnik.* 1998. ISBN: 978-3-663-09902-4. DOI: `10.1007/978-3-663-09902-4`.

[99] Harald A. Øye. *Power Failure, Temporary Pot Shut-Down, Restart and Repair.* 2012.

[100] Dillon Pariente and Emmanuel Ledinot. *Formal verification of industrial C code using Frama-C: a case study.* 2010.

[101] McDaniel Patrick and McLaughlin Stephen. “Security and Privacy Challenges in the Smart Grid”. In: *Secure Systems* (May 2009).

[102] Adrian Perrig et al. “Efficient Authentication and Signing of Multicast Streams over Lossy Channels”. In: *Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000.* IEEE, 2000. ISBN: 0-7695-0665-8. DOI: `https://doi.org/10.1109/SECPRI.2000.848446`.

[103] Trevor Perrin. “The Noise protocol framework, 2015”. In: *URL http://noiseprotocol.org/noise. pdf* (2016).

[104] Fabien Petitcolas. *Kerckhoffs’ principles from “La cryptographie militaire”.* English. 2020. URL: `https://www.petitcolas.net/kerckhoffs/index.html` (visited on 05/25/2020).

[105] James Pierce and Eric Paulos. “Beyond Energy Monitors: Interaction, Energy, and Emerging Energy Systems. interaction, energy, and emerging energy systems”. In: *CHI 2012.* 2012. DOI: `10.1145/2207676.2207771`.

[106] Sandro Pinto and Nuno Santos. “Demystifying Arm TrustZone: A Comprehensive Survey”. In: *ACM Comput. Surv.* 51.6 (Jan. 2019). ISSN: 0360-0300. DOI: `10.1145/3291047`.

[107] Tom A. Rodden et al. “At Home with Agents: Exploring Attitudes Towards Future Smart Energy Infrastructures”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI ’13.* 2013. DOI: `10.1145/2470654`.

[108] Graham Rogers. “Power System Oscillations”. In: Kluwer, 2000.

[109] Dan Rosenberg. “Qsee trustzone kernel integer over flow vulnerability”. In: *Black Hat conference*. 2014.

[110] Takuro Sato et al. *Smart Grid Standards. Specifications, Requirements and Technologies*. Wiley, 2015.

[111] Bruce Schneier. *Secrecy, Security, and Obscurity*. May 2002. URL: `https://www.schneier.com/crypto-gram/archives/2002/0515.html`.

[112] Anatoli Semerow et al. *Dynamic Study Model for the interconnected power system of Continental Europe in different simulation tools*. 2015. DOI: `10.1109/ptc.2015.7232578`.

[113] *Single Market Progress Report: Country Profiles – Italy*. Research rep. 2014.

[114] *Stromgrundversorgungsverordnung StromGVV § 19 Unterbrechung der Versorgung*. Mar. 14, 2019. URL: `http://www.gesetze-im-internet.de/stromgvv/__19.html` (visited on 05/18/2020).

[115] Michael Stuber. “Standards, Security, and Smart Meters”. In: *Smart Grid Handbook*. 2016, pp. 1–14. DOI: `10.1002/9781118755471.sgd036`.

[116] William G. Temple, Binbin Chen, and Nils Ole Tippenhauer. “Delay Makes a Difference: Smart Grid Resilience Under Remote Meter Disconnect Attack”. In: *2013 IEEE International Conference on Smart Grid Communications*. 2013. DOI: `https://doi.org/10.1109/SmartGridComm.2013.6688001`.

[117] The CEN/CENELEC/ETSI Joint Working Group Standards Smart on for Grids. *Final report of the CEN/CENELEC/ETSI Joint Working Group on Standards for Smart Grids*. CEN/CENELEC/ETSI, May 2011.

[118] The European Commission, ed. *The Energy Efficiency Directive*. 2012. URL: `https://ec.europa.eu/energy/topics/energy-efficiency/targets-directive-and-rules/energy-efficiency-directive_en` (visited on 05/18/2020).

[119] UCTE/ENTSO-E. *Operation Handbook*. 2004.

[120] UCTE/ENTSO-E. *Operation Handbook*. 2009.

[121] UK Department for Business Energy and Industrial Strategy. *Smart Meter Statistics Quarterly Report to end March 2019*. 2019.

[122] UK Department for Business, Energy and Industrial Strategy. *Smart Meter Rollout Cost-Benefit Analysis Part I*. 2016.

[123] UK Department for Business, Energy and Industrial Strategy. *Smart Metering Implementation Programme Progress Report for 2018*. 2018.

[124] UK Department of Energy and Climate Change. *Smart Metering Implementation Programme: Smart Metering Equipment Technical Specifications*. Version 1.58. 2014.

[125] Noelia Uribe-Pérez et al. "State of the Art and Trends Review of Smart Metering in Electricity Grids". In: *Applied Sciences* 6.3 (Feb. 2016), p. 68. DOI: `10.3390/app6030068`.

[126] Andoni Urtasun et al. "Energy management strategy for a battery-diesel stand-alone system with distributed PV generation based on grid frequency modulation". In: *Renewable Energy* 66 (Jan. 2014), pp. 325–336.

[127] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: `https://doi.org/10.1038/s41592-019-0686-2`.

[128] Loren Weith. "DLMS / COSEM Protocol Security Evaluation". MA thesis. Department of Mathematics and Computer Science, Eindhoven University of Technology, 2014.

[129] Paul S. Wright. *Library of ROCOF Test Waveforms – Pseudo Code, V1.0, May 2019.* 2019. DOI: `10.5281/zenodo.3559798`.

[130] Yongdong Wu et al. "Resonance Attacks on Load Frequency Control of Smart Grids". In: *IEEE Transactions on Smart Grid* 9.5 (Sept. 2018), pp. 4490–4502. DOI: `10.1109/TSG.2017.2661307`.

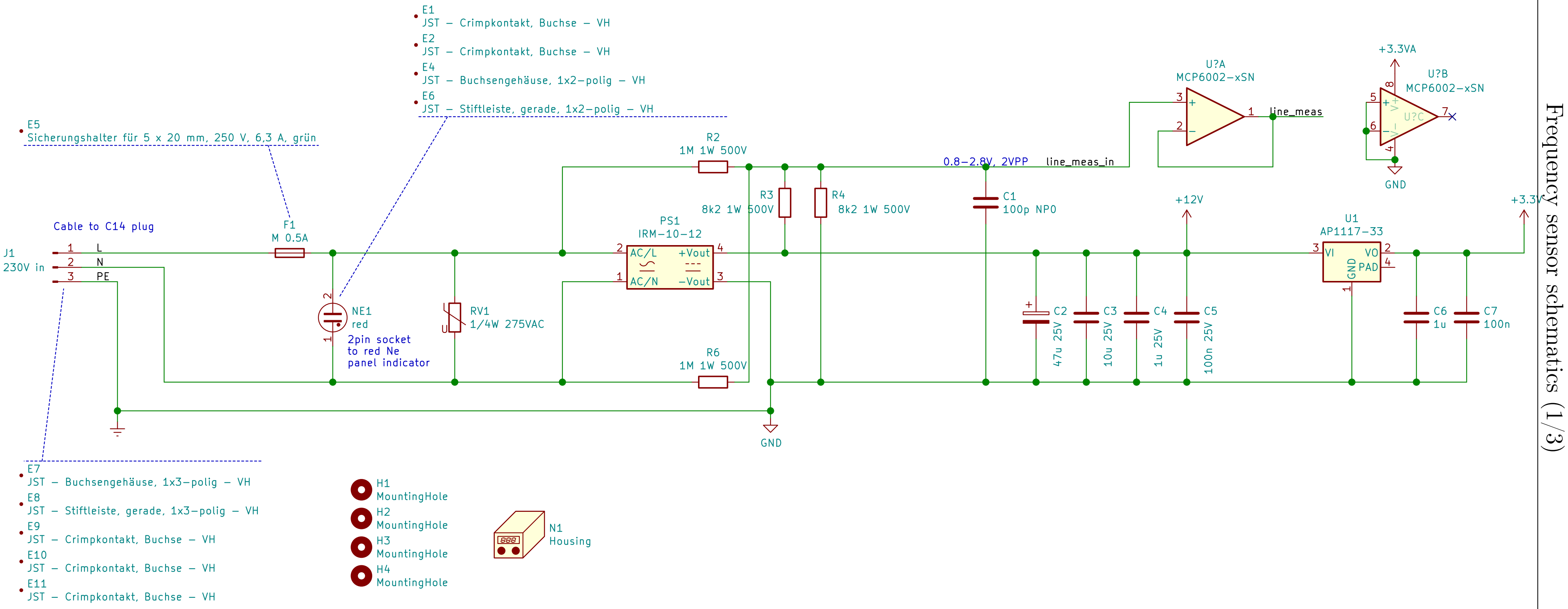
[131] Ye Yan et al. "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges". In: *IEEE Communications Surveys & Tutorials* (2012). DOI: `10.1109/SURV.2012.021312.00034`.

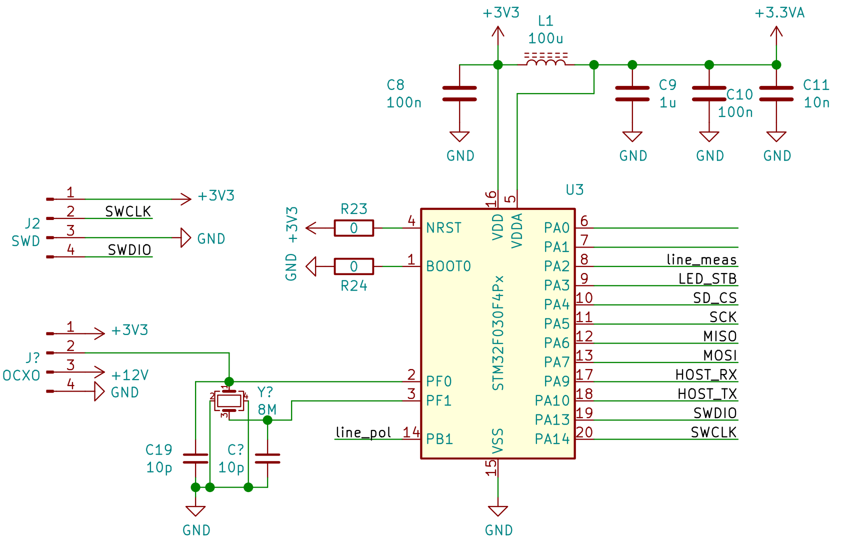
[132] Jixuan Zheng, David Wenzhong Gao, and Li Lin. "Smart meters in smart grid: An overview". In: *2013 IEEE Green Technologies Conference (GreenTech).* IEEE. 2013, pp. 57–64.

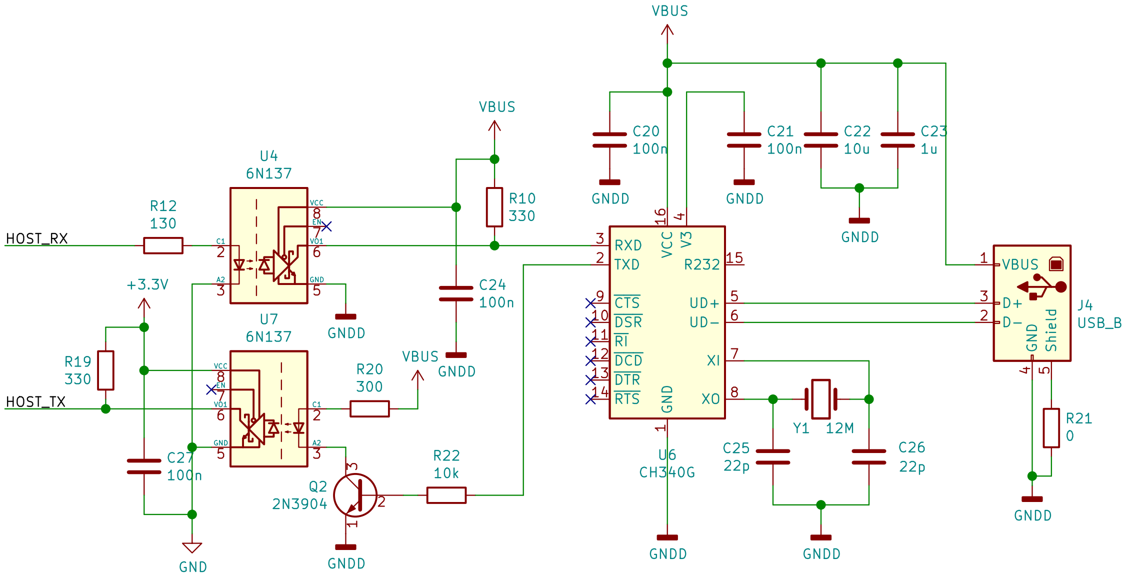
[133] Shoshana Zuboff. *The Age of Surveillance Capitalism.* 2019.

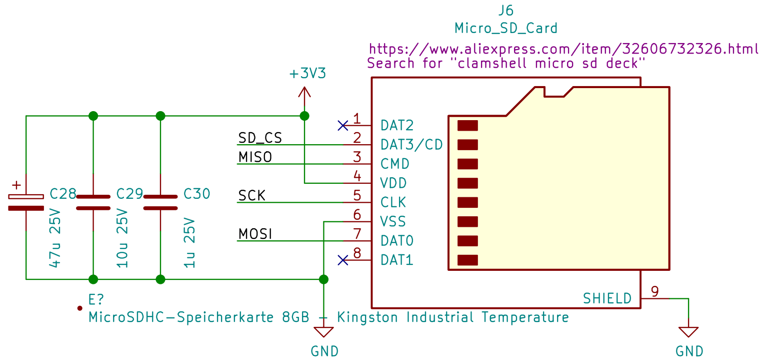
Appendix A

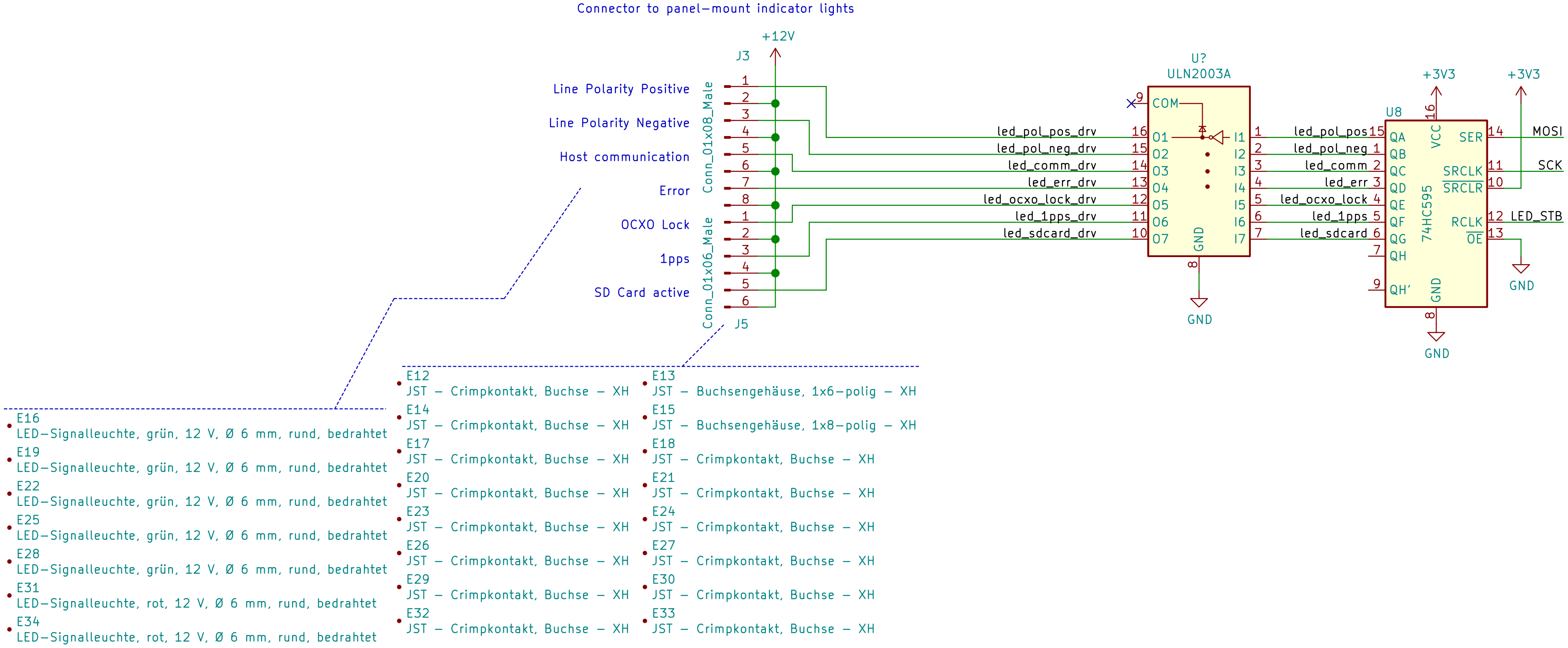
Frequency sensor schematics


E1
JST – Crimpkontakt, Buchse – VH
E2
JST – Crimpkontakt, Buchse – VH
E4
JST – Buchsengehäuse, 1x2–polig – VH
E6
JST – Stiftleiste, gerade, 1x2–polig – VH
E5
Sicherungshalter für 5 x 20 mm, 250 V, 6,3 A, grün
Cable to C14 plug
J1
230V in
1 L
2 N
3 PE
F1
M 0.5A
NE1
red
2pin socket
to red Ne
panel indicator
RV1
1/4W 275VAC
R2
1M 1W 500V
R3
8k2 1W 500V
R4
8k2 1W 500V
PS1
IRM–10–12
AC/L
AC/N
+Vout
–Vout
R6
1M 1W 500V
GND
0.8–2.8V, 2VPP
line_meas_in
C1
100p NP0
U?A
MCP6002–xSN
line_meas
U?B
MCP6002–xSN
U?C
+3.3VA
GND
+12V
U1
AP1117–33
VI
VO
PAD
GND
+3.3V
C2
47u 25V
C3
10u 25V
C4
1u 25V
C5
100n 25V
C6
1u
C7
100n
E7
JST – Buchsengehäuse, 1x3–polig – VH
E8
JST – Stiftleiste, gerade, 1x3–polig – VH
E9
JST – Crimpkontakt, Buchse – VH
E10
JST – Crimpkontakt, Buchse – VH
E11
JST – Crimpkontakt, Buchse – VH
H1
MountingHole
H2
MountingHole
H3
MountingHole
H4
MountingHole
N1
Housing


+3V3
L1
100u
+3.3VA
C8
100n
C9
1u
C10
100n
C11
10n
GND
U3
J2
SWD
1 +3V3
2 SWCLK
3 GND
4 SWDIO
R23
0
R24
0
+3V3
GND
NRST
BOOT0
VDD
VDDA
STM32F030F4Px
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA9
PA10
PA13
PA14
line_meas
LED_STB
SD_CS
SCK
MISO
MOSI
HOST_RX
HOST_TX
SWDIO
SWCLK
J?
OCXO
+12V
PF0
PF1
PB1
line_pol
VSS
Y?
8M
C19
10p
C?
10p


VBUS
C20
100n
C21
100n
C22
10u
C23
1u
GNDD
U4
6N137
R12
130
HOST_RX
HOST_TX
R10
330
C24
100n
U7
6N137
+3.3V
R19
330
C27
100n
R20
300
Q2
2N3904
R22
10k
GND
RXD
TXD
VCC
V3
R232
CTS
DSR
RI
DCD
DTR
RTS
UD+
UD−
XI
XO
GND
U6
CH340G
Y1
12M
C25
22p
C26
22p
VBUS
D+
D−
GND
Shield
J4
USB_B
R21
0

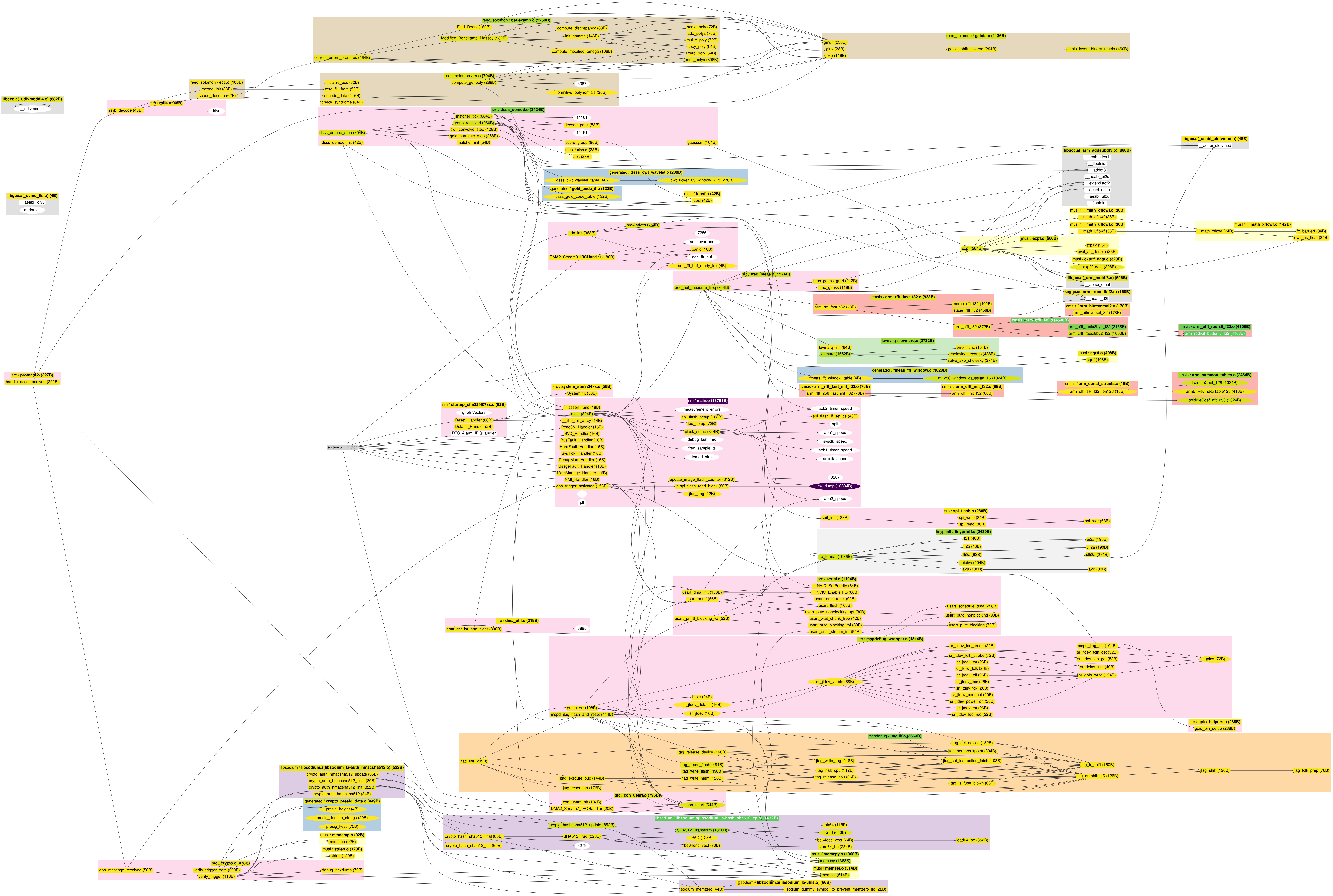

J6
Micro_SD_Card
https://www.aliexpress.com/item/32606732326.html
Search for "clamshell micro sd deck"
+3V3
SD_CS
MISO
SCK
MOSI
DAT2
DAT3/CD
CMD
VDD
CLK
VSS
DAT0
DAT1
SHIELD
C28
47u 25V
C29
10u 25V
C30
1u 25V
E?
MicroSDHC−Speicherkarte 8GB − Kingston Industrial Temperature
GND


Connector to panel−mount indicator lights
+12V
J3
Conn_01x08_Male
Conn_01x06_Male
J5
Line Polarity Positive
Line Polarity Negative
Host communication
Error
OCXO Lock
1pps
SD Card active
U?
ULN2003A
COM
9
16 O1
15 O2
14 O3
13 O4
12 O5
11 O6
10 O7
GND
8
I1 1
I2 2
I3 3
I4 4
I5 5
I6 6
I7 7
led_pol_pos_drv
led_pol_neg_drv
led_comm_drv
led_err_drv
led_ocxo_lock_drv
led_1pps_drv
led_sdcard_drv
led_pol_pos
led_pol_neg
led_comm
led_err
led_ocxo_lock
led_1pps
led_sdcard
U8
74HC595
+3V3
+3V3
VCC
QA
QB
QC
QD
QE
QF
QG
QH
QH'
SER 14
SRCLK 11
SRCLR 10
RCLK 12
OE 13
MOSI
SCK
LED_STB
GND
GND
GND
GND
E12
JST − Crimpkontakt, Buchse − XH
E13
JST − Buchsengehäuse, 1x6−polig − XH
E14
JST − Crimpkontakt, Buchse − XH
E15
JST − Buchsengehäuse, 1x8−polig − XH
E16
LED−Signalleuchte, grün, 12 V, Ø 6 mm, rund, bedrahtet
E17
JST − Crimpkontakt, Buchse − XH
E18
JST − Crimpkontakt, Buchse − XH
E19
LED−Signalleuchte, grün, 12 V, Ø 6 mm, rund, bedrahtet
E20
JST − Crimpkontakt, Buchse − XH
E21
JST − Crimpkontakt, Buchse − XH
E22
LED−Signalleuchte, grün, 12 V, Ø 6 mm, rund, bedrahtet
E23
JST − Crimpkontakt, Buchse − XH
E24
JST − Crimpkontakt, Buchse − XH
E25
LED−Signalleuchte, grün, 12 V, Ø 6 mm, rund, bedrahtet
E26
JST − Crimpkontakt, Buchse − XH
E27
JST − Crimpkontakt, Buchse − XH
E28
LED−Signalleuchte, grün, 12 V, Ø 6 mm, rund, bedrahtet
E29
JST − Crimpkontakt, Buchse − XH
E30
JST − Crimpkontakt, Buchse − XH
E31
LED−Signalleuchte, rot, 12 V, Ø 6 mm, rund, bedrahtet
E32
JST − Crimpkontakt, Buchse − XH
E33
JST − Crimpkontakt, Buchse − XH
E34
LED−Signalleuchte, rot, 12 V, Ø 6 mm, rund, bedrahtet

Appendix B

Demonstrator firmware symbol size map

Please find this appendix enclosed in the pouch on the inside of the back cover.


reed_solomon / berlekamp.o (2250B)
Find_Roots (180B)
Modified_Berlekamp_Massey (532B)
correct_errors_erasures (464B)
compute_discrepancy (88B)
init_gamma (146B)
compute_modified_omega (106B)
scale_poly (72B)
add_polys (76B)
mul_z_poly (72B)
copy_poly (64B)
zero_poly (54B)
mult_polys (396B)
reed_solomon / galois.o (1136B)
gmult (238B)
ginv (28B)
gexp (116B)
galois_shift_inverse (294B)
galois_invert_binary_matrix (460B)
reed_solomon / ecc.o (100B)
rscode_init (38B)
rscode_decode (62B)
reed_solomon / rs.o (794B)
initialize_ecc (32B)
zero_fill_from (56B)
decode_data (116B)
check_syndrome (64B)
compute_genpoly (288B)
6387
primitive_polynomials (36B)
libgcc.a(_udivmoddi4.o) (682B)
__udivmoddi4
src / rslib.o (48B)
rslib_decode (48B)
driver
src / dsss_demod.o (3424B)
matcher_tick (684B)
group_received (960B)
cwt_convolve_step (128B)
gold_correlate_step (268B)
matcher_init (54B)
dsss_demod_step (804B)
dsss_demod_init (42B)
11161
decode_peak (58B)
11191
score_group (96B)
gaussian (104B)
musl / abs.o (28B)
abs (28B)
generated / dsss_cwt_wavelet.o (280B)
dsss_cwt_wavelet_table (4B)
cwt_ricker_69_window_7F3 (276B)
generated / gold_code_5.o (132B)
dsss_gold_code_table (132B)
musl / fabsf.o (42B)
fabsf (42B)
libgcc.a(_dvmd_tls.o) (4B)
__aeabi_ldiv0
attributes
libgcc.a(_aeabi_uldivmod.o) (48B)
__aeabi_uldivmod
libgcc.a(_arm_addsubdf3.o) (888B)
__aeabi_drsub
__floatsidf
__adddf3
__aeabi_ui2d
__extendsfdf2
__aeabi_dsub
__aeabi_ul2d
__floatdidf
musl / __math_oflowf.o (36B)
__math_oflowf (36B)
musl / __math_uflowf.o (36B)
__math_uflowf (36B)
musl / __math_xflowf.o (142B)
__math_xflowf (74B)
fp_barrierf (34B)
eval_as_float (34B)
musl / expf.o (660B)
expf (564B)
top12 (26B)
eval_as_double (36B)
musl / exp2f_data.o (328B)
__exp2f_data (328B)
libgcc.a(_arm_muldf3.o) (596B)
__aeabi_dmul
libgcc.a(_arm_truncdfsf2.o) (160B)
__aeabi_d2f
src / adc.o (754B)
adc_init (368B)
7256
adc_overruns
panic (16B)
adc_fft_buf
DMA2_Stream0_IRQHandler (180B)
adc_fft_buf_ready_idx (4B)
src / freq_meas.o (1274B)
func_gauss_grad (212B)
func_gauss (118B)
adc_buf_measure_freq (944B)
cmsis / arm_rfft_fast_f32.o (938B)
merge_rfft_f32 (402B)
stage_rfft_f32 (458B)
arm_rfft_fast_f32 (78B)
cmsis / arm_bitreversal2.o (178B)
arm_bitreversal_32 (178B)
cmsis / arm_cfft_f32.o (4530B)
arm_cfft_f32 (372B)
arm_cfft_radix8by4_f32 (3158B)
arm_cfft_radix8by2_f32 (1000B)
cmsis / arm_cfft_radix8_f32.o (4108B)
arm_radix8_butterfly_f32 (4108B)
levmarq / levmarq.o (2732B)
levmarq_init (64B)
levmarq (1652B)
error_func (154B)
cholesky_decomp (488B)
solve_axb_cholesky (374B)
musl / sqrtf.o (40B)
sqrtf (40B)
generated / fmeas_fft_window.o (1028B)
fmeas_fft_window_table (4B)
fft_256_window_gaussian_16 (1024B)
cmsis / arm_rfft_fast_init_f32.o (76B)
arm_rfft_256_fast_init_f32 (76B)
cmsis / arm_cfft_init_f32.o (88B)
arm_cfft_init_f32 (88B)
cmsis / arm_const_structs.o (16B)
arm_cfft_sR_f32_len128 (16B)
cmsis / arm_common_tables.o (2464B)
twiddleCoef_128 (1024B)
armBitRevIndexTable128 (416B)
twiddleCoef_rfft_256 (1024B)
src / protocol.o (327B)
handle_dsss_received (292B)
src / system_stm32f4xx.o (56B)
SystemInit (56B)
src / startup_stm32f407xx.o (82B)
g_pfnVectors
Reset_Handler (80B)
Default_Handler (2B)
RTC_Alarm_IRQHandler
section .isr_vector
src / main.o (18761B)
__assert_func (18B)
main (824B)
__libc_init_array (14B)
PendSV_Handler (16B)
SVC_Handler (16B)
BusFault_Handler (16B)
HardFault_Handler (16B)
SysTick_Handler (16B)
DebugMon_Handler (16B)
UsageFault_Handler (16B)
MemManage_Handler (16B)
NMI_Handler (16B)
oob_trigger_activated (156B)
iplt
plt
measurement_errors
spi_flash_setup (188B)
led_setup (72B)
clock_setup (344B)
debug_last_freq
freq_sample_ts
demod_state
update_image_flash_counter (312B)
jt_spi_flash_read_block (80B)
jtag_img (12B)
apb2_timer_speed
spi_flash_if_set_cs (48B)
spif
apb1_speed
sysclk_speed
apb1_timer_speed
auxclk_speed
8287
fw_dump (16384B)
apb2_speed
src / spi_flash.o (260B)
spif_init (128B)
spi_write (34B)
spi_read (30B)
spi_xfer (68B)
tinyprintf / tinyprintf.o (2430B)
tfp_format (1036B)
i2a (46B)
li2a (46B)
lli2a (62B)
putchw (404B)
a2u (102B)
ui2a (190B)
uli2a (190B)
ulli2a (274B)
a2d (80B)
src / serial.o (1194B)
usart_dma_init (156B)
usart_printf (56B)
usart_printf_blocking_va (52B)
__NVIC_SetPriority (84B)
__NVIC_EnableIRQ (60B)
usart_dma_reset (92B)
usart_flush (108B)
usart_putc_nonblocking_tpf (30B)
usart_wait_chunk_free (42B)
usart_putc_blocking_tpf (30B)
usart_dma_stream_irq (94B)
usart_schedule_dma (228B)
usart_putc_nonblocking (90B)
usart_putc_blocking (72B)
src / dma_util.o (319B)
dma_get_isr_and_clear (300B)
6895
src / mspdebug_wrapper.o (1514B)
printf_err (108B)
mspd_jtag_flash_and_reset (444B)
htole (24B)
sr_jtdev_default (16B)
sr_jtdev (16B)
sr_jtdev_vtable (68B)
sr_jtdev_led_green (22B)
sr_jtdev_tclk_strobe (72B)
sr_jtdev_tst (26B)
sr_jtdev_tclk (26B)
sr_jtdev_tdi (26B)
sr_jtdev_tms (26B)
sr_jtdev_tck (26B)
sr_jtdev_connect (20B)
sr_jtdev_power_on (20B)
sr_jtdev_rst (26B)
sr_jtdev_led_red (22B)
mspd_jtag_init (104B)
sr_jtdev_tclk_get (52B)
sr_jtdev_tdo_get (52B)
sr_delay_inst (40B)
sr_gpio_write (124B)
gpios (72B)
src / gpio_helpers.o (288B)
gpio_pin_setup (288B)
mspdebug / jtaglib.o (3663B)
jtag_init (292B)
jtag_release_device (160B)
jtag_erase_flash (484B)
jtag_write_flash (490B)
jtag_write_mem (128B)
jtag_execute_puc (144B)
jtag_reset_tap (176B)
jtag_get_device (132B)
jtag_set_breakpoint (304B)
jtag_write_reg (218B)
jtag_set_instruction_fetch (108B)
jtag_halt_cpu (112B)
jtag_release_cpu (66B)
jtag_is_fuse_blown (68B)
jtag_ir_shift (150B)
jtag_dr_shift_16 (126B)
jtag_shift (190B)
jtag_tclk_prep (76B)
libsodium / libsodium.a(libsodium_la-auth_hmacsha512.o) (522B)
crypto_auth_hmacsha512_update (36B)
crypto_auth_hmacsha512_final (80B)
crypto_auth_hmacsha512_init (322B)
crypto_auth_hmacsha512 (84B)
generated / crypto_presig_data.o (449B)
presig_height (4B)
presig_domain_strings (20B)
presig_keys (75B)
src / con_usart.o (796B)
con_usart_init (132B)
DMA2_Stream7_IRQHandler (20B)
con_usart (644B)
libsodium / libsodium.a(libsodium_la-hash_sha512_cp.o) (4672B)
crypto_hash_sha512_update (852B)
crypto_hash_sha512_final (80B)
crypto_hash_sha512_init (60B)
SHA512_Transform (1816B)
SHA512_Pad (228B)
6279
PAD (128B)
be64enc_vect (70B)
rotr64 (118B)
Krnd (640B)
be64dec_vect (74B)
store64_be (254B)
load64_be (352B)
musl / memcmp.o (92B)
memcmp (92B)
musl / strlen.o (120B)
strlen (120B)
musl / memcpy.o (1368B)
memcpy (1368B)
musl / memset.o (514B)
memset (514B)
src / crypto.o (478B)
oob_message_received (58B)
verify_trigger_dom (220B)
verify_trigger (116B)
debug_hexdump (72B)
libsodium / libsodium.a(libsodium_la-utils.o) (66B)
sodium_memzero (44B)
_sodium_dummy_symbol_to_prevent_memzero_lto (22B)