# Hacking as Transgressive Infrastructuring: Mobile Phone Networks and the German Chaos Computer Club

**Susann Wagenknecht**
Department of Social Sciences
University of Siegen, Germany
su.wagen@gmail.com

**Matthias Korn**
School of Informatics and Computing
Indiana University, IUPUI, USA
korn@iupui.edu

## ABSTRACT

This paper applies the theoretical lens of infrastructure to study hacking practices that take issue with large-scale communication networks. The paper analyzes a series of hacks targeting the Global System for Mobile Communications (i.e., networks for mobile telephony) carried out by a cluster of people affiliated or sympathetic to the German Chaos Computer Club between 2001 and 2014. These hacks aim at acquiring proprietary knowledge and facilitating the autonomous operation of local mobile phone networks for communities, independent of corporate network providers. The contribution of this paper is to show how hacking of this kind can be understood as *transgressive infrastructuring*, a way of engaging critically with infrastructure that, in the case of GSM hacking, relied on three strategies—reverse engineering, re-implementation, and parallel operation, all of which aim at appropriating the targeted network intellectually, legally, functionally, and/or operationally.

### Author Keywords

Hacking; infrastructure; transgression; appropriation; politics; Germany; Chaos Computer Club; mobile phone networks; GSM.

### ACM Classification Keywords

K.4.2 [**Computers and Society**]: Social Issues; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—Unauthorized access (e.g., hacking); C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—Wireless communication.

## INTRODUCTION

This paper investigates how and why hacking targets infrastructures of mobile telephony, a practice that is as much about communication networks, their hard- and software components, as it is about the (sub-)cultural sociotechnical 'imaginary' [36] of 'open,' community-operated networks. Hacking is a way to learn about communication infrastructure, sometimes exploiting it, and sometimes, as we observe, trying to remake it. Hacking embodies the conviction that technology is made by humans and can be remade, and that the (re-)making of technology is shaped by social, cultural, economic, and political configurations as well as it helps shaping them [72], a conviction challenging technological determinism and making hacking a practice with techno-political impetus ([37]: 130ff.; see also [15, 54, 56]).

As a research field, Computer-Supported Collaborative Work (CSCW) has a growing interest in infrastructures—the sociotechnical substrate of computer-mediated human interaction (e.g., [1, 34, 46, 52]). As some researchers have pointed out, infrastructures are slow to change ([66]: 132) and tend to reinforce complacency and stasis [51], powerfully 'torquing' human-technology relations ([10]: 27). Given this condition, how can one engage with infrastructures critically and effectively? In this paper, we contribute one possible answer to this question when we investigate hacking with the analytic lens of infrastructure. With this lens, we characterize hacking practices in the case that we study as transgressive infrastructuring, i.e., as an in-the-wild attempt to subversively remake existing sociotechnical infrastructures.

Our case consists in a series of hacks targeting the Global System for Mobile Communications (GSM) carried out by a cluster of people affiliated or sympathetic to the Chaos Computer Club (CCC), Europe's largest hacker association. We study this case through a qualitative analysis of 42 presentations about GSM hacking that have been delivered at the CCC's annual convention between 2001 and 2014. For many involved hackers, the goal of GSM hacking is to 'open' proprietary knowledge and facilitate the autonomous operation of local GSM networks for communities, independent of corporate network providers—ultimately seeking to rebuild those networks and affect infrastructural change. We observe that to achieve this goal, hackers make use of three strategies: reverse engineering, re-implementation, and parallel operation. All of these infrastructuring strategies serve the aim to collectively appropriate GSM technology intellectually, legally, functionally, and/or operationally.

## RELATED WORK: HACKING

In CSCW and Human-Computer Interaction (HCI), the term 'hacking' is associated (and often treated as overlapping) with phenomena such as Do-It-Yourself and 'maker culture' [44, 63, 69, 70], but also end-user innovation [24, 76]. Drawing on a multitude of case studies, research in these fields addresses hacking as a technique and/or as a socio-cultural practice.

In terms of technique, research in CSCW and HCI has described hacking as "appropriating, modifying, or 'kludging'

existing resources (devices, hardware, software, or anything within reach) to suit other purposes, often in an ingenious fashion" ([53]: 13). Hacking finds these resources commonly in off-the-shelf mass-produced devices [5, 35, 60, 81] or established infrastructures [17]. Hartman and collaborators relate hacking to "opportunistic practices" that create "mashup designs" of "ad-hoc nature" ([29]: 46). Similarly, Bardzell and collaborators [5] characterize tool-use and tool-making in hackerspaces as "ad hoc," i.e., pragmatic and situation-specific. In account of the ingenuity that hacking practices display, some authors emphasize the economic value of hacking in the context of product innovation, industrial R&D, and entrepreneurship [49, 77, 81].

In focusing on GSM hacking, our paper complements this research by providing a case of hacking that targets communication networks—a practice distinct from making and DIY practices, which often target end-user products, but not large-scale networks. We show that faced with the scope and complexity of GSM technology, hacking leverages specific strategies to target mobile phone networks.

When addressing hacking as socio-political practice, CSCW and HCI research has emphasized the 'politics' of hacking, examining underlying community structures [5, 49, 60, 79] and drawing attention to accompanying struggles for power and mechanism of social in- and exclusion [22, 61, 69]. Tanenbaum and collaborators, e.g., recognize hacking, making, and DIY as forms of resistance "against the hegemonic structures of mass production in the industrialized world," yet critically reflect upon the promise of democratization and empowering that accompanies parts of hacking, making, and DIY discourses ([70]: 2609; see also [3, 60, 79]). Here, and also in, e.g., Fox et al.'s study of a feminist hackerspace [26], some of the socio-political sensibilities of hacking are unpacked, and our paper is continuous with this research when we elaborate upon the vision of 'open' technology that underlies GSM hacking.

Beyond CSCW and HCI, social-scientific research and social theory has discussed the transgressive character of hacking practice, as well as its cultural identity and political sentiment.[1]

As hacking 'floods,' 'blocks,' 'invades,' 'destroys,' and 'rewrites' existing systems, social scientists have described computer hacking as "illicit" ([71]: 15) and "transgressive craft" ([68]: 125). Through deliberate misuse or re-purposing, hacking creatively undermines the conventions that are, often tacitly, inherent to existing systems and networks; hacking transgresses terms and conditions, established patterns of use, cultural expectations, economic standards, legal norms as well as programming rules. In making tacit conventions visible, hacking explicates the contingency and alterability of computer systems, problematizing the customary and its limitations [56]. In this vein, hacking can be understood as the perpetually evolving processing of program(matic) limitations, as indefatigable 'playing with the rules of the game.'

[1] All research referred to in the following, except for [56], is based on case studies of self-identified hacking communities.

Social-scientific research elucidates how the transgressive character of hacking is imbued with sub-cultural identity and political sentiment. Social scientists situate some of hacking's cultural roots in the cyberpunk movement ([71]: 168; [28]) and characterize hacking as practice that defies what it perceives as cultural mainstream, adopting independence, both technologically and culturally, as a core value ([72]: 80). Social scientists describe how hackers portray themselves as outsiders and vanguards ([75]: 198ff.) who, in contrast to non-hackers, do not subject themselves to the technological pressures that corporations and government agencies leverage against the public ([27]: 194; also [74]).

Social-scientific research has discerned a "strong antistatist and anticorporatist ideology" in parts of the hacker community ([7]: 53), finding that hackers oppose "the commodification of information" on the basis of an "anti-authoritarian" perspective ([71]: 61ff.). Some authors conceive of parts of hacking as a pedagogical mission, a push for social emancipation through the public diffusion of sociotechnical knowledge ([37]: 134)—handing computer systems' technical potential to the public that relies upon it [54].

Notably sociologists Coleman and Golub [15] have studied the political sentiment of hacking, showing how the ethos dominating the American hacker community reflects deep-rooted liberalism (or, libertarianism), its core values being the right to privacy, freedom of speech, individual power, and meritocracy. The authors distinguish different facets of political sentiment: While the movement for crypto-freedom tends to communicate a notion of negative freedom (i.e., (encrypted) privacy as the freedom *from* others), hacking practices that 'open,' reclaim, and re-create proprietary knowledge tend to articulate a notion of positive freedom (i.e., freedom as *enabling* to do things). Positive freedom in this context is tied to ideas of "reciprocity, pedagogy and scientific openness" ([15]: 261). Furthermore, Coleman and Golub characterize (legal) transgression as a strategy in the "constant arms race between those with the knowledge and power to erect barriers and those with the equal power, knowledge and especially desire, to disarm them" ([15]: 263). In relating to the creative and exploratory expression of individuality and individual power, Coleman and Golub argue, the motif of transgression, too, is palpably reflecting American liberalism.

The social-scientific research presented here characterizes hacking as an effort to challenge 'the rules of the game'—a large-scale game that intertwines the technical with the social and the political. As background to our case study, this literature has reinforced our decision to study our case of hacking through the lens of infrastructure.

## THEORETICAL FRAMING: INFRASTRUCTURE

We apply the theoretical lens of infrastructure in order to study hacking practices that take issue with large-scale communication networks and their sociotechnical configuration. During the last ten years, research in CSCW has increasingly focused on infrastructures, particularly on "cyberinfrastructures" and "e-infrastructures," i.e., IT-based work infrastructures for collaboration within and across organizations [34, 46, 55, 57, 59]. Recently, Monteiro and collaborators have argued that to shift

attention from artifacts to infrastructures helps facilitating a shift of interest from groups to communities, from a localist focus on the 'here and now' to long-term perspectives ([52]; see also [39, 40, 51]).

Lately, infrastructures for mobile phone use have emerged as a new research interest in CSCW [1, 32, 80]. Mobile phone infrastructures confront CSCW research with a new scale. They require a far-spanning mesh of hard- and software, and they are interwoven with conventional landlines and other networks, involving global corporations and national markets, international standards, laws and regulation, multi-level lobbying and policy making (see, e.g., [80]). Most importantly, mobile phone infrastructures involve, potentially, millions of users, diverse practices of use and repair, forms of appropriation, cultural interpretations, and socio-economic contexts [1].

Conceptually, we root our notion of infrastructure in the work of Star and collaborators [10, 65, 66] who conceive of infrastructure as the sociotechnical substrate for human action, structures that can powerfully 'torque' humans and limit their scope of action ([10]: 27). Star and Ruhleder [66] show how infrastructures are shaped by and shape conventions of practice; their design incorporates values and reifies them tacitly, breeding complacency and stasis (cf. [51]). As they support other activities, infrastructures are taken for granted and remain largely invisible, turning visible only in cases of breakdown (see also [55]). Therefore, Star and Ruhleder suggest to focus on the relational and processual character of infrastructures.

Whether a sociotechnical arrangement is infrastructure or not depends on the way in which humans relate to it. Sociotechnical arrangements *become* infrastructure when they are acted 'through' rather than acted 'upon,' when they function as substrate rather than object of technologically-mediated action. With the questions '*when* is infrastructure' and '*for whom*,' Star and Ruhleder [66] point out that what functions as infrastructure in one context may loose its taken-for-granted—i.e., infrastructural—character in another (e.g., when Wifi looses its signal). Similarly, what is infrastructure for some (e.g., users) may not be so for others (e.g., for network administrators).

Furthermore, infrastructures continuously change, if slowly and often barely discernible. Infrastructures are built, maintained, updated, adjusted, modified, worn down, damaged, repaired, sometimes contested (typically not) and rarely challenged effectively. Star and Bowker have therefore suggested to study infrastructures through the practices upholding them over time, raising the question 'how to infrastructure?' [65]. To infrastructure requires what Bowker and Star have called "infrastructural inversion"—"a struggle against the tendency of infrastructures to disappear," "look[ing] closely at technologies and arrangements that, by design and by habit, tend to fade into the woodwork" ([10]: 34).

A processual notion of infrastructure has also emerged in the field of Participatory Design (PD), where Le Dantec and DiSalvo characterize 'infrastructuring' as "the work of creating sociotechnical resources that intentionally enable adoption and appropriation beyond the initial scope of the design" ([45]: 247). In PD, infrastructuring has been proposed as a design approach that empowers users through guided capacity building [8, 23, 38]. Some authors, however, hold that infrastructuring also occurs 'in-the-wild' and is observable in the practices of non-professionals or professionals-off-duty [19, 41], a line of research we reflect when we characterize hacking as infrastructuring for the in-the-wild case we observe.

Building upon the work of Star and collaborators, and adopting impulses from infrastructuring approaches in PD, we speak of infrastructuring—by way of a definition—as practices that render infrastructures visible, problematize them, engage with them, and work 'upon' them, not through them, cognizant of the fact that these structures are the substrate of human activity that usually goes unquestioned and barely noticed (cf. [39, 55]).

In our study, we place particular emphasis on the forms of *appropriation* that infrastructuring involves (cf. [8]: 43): Infrastructuring is performed by people who familiarize themselves enough with existing infrastructures to reconfigure them; it is performed intentionally to address people's needs and desires and accommodate existing or envisioned practices of use. In fact, we will argue that the vision of 'open,' community-operated networks that underlies GSM hacking is, essentially, a vision of appropriation.

In CSCW, appropriation is a multi-faceted concept, constitutive for the field. Appropriation commonly refers to "the ways in which people adopt and adapt interactive technologies, fitting them into working practices and evolving those practices around them" ([20]: 487). Typically, studies in appropriation are interested in how technology is 'made work' in local setups. In this vein, appropriation has been described as "configuration work" [4] that requires collective efforts [21]. While much research discusses appropriation in the context of group collaborations and organizations, scholars have begun to characterize appropriation as a cultural phenomenon that intertwines with "collective belonging, economic interest and political discourse" ([48]: 78; see also [2]). Another strand of recent research directs attention to unexpected forms of appropriation, showing how technologies are appropriated in ways unforeseen by design and even challenging design intentions [62, 73]. Against this background, we explore different forms of appropriation that occur in the case of hacking we study, pursuing a conceptual angle that prior research on hacking, making, DIY, and repair has opened [29, 60, 70, 79].

## EMPIRICAL CONTEXT: THE CCC

The Chaos Computer Club (CCC), founded in 1986, is today's largest hacker association in Europe. In its online mission statement, the CCC describes itself as a community that seeks to promote "Informationsfreiheit" (*freedom of information*), discusses the impact of technology on individual and collective level, and educates the public about it.[2] Prominently placed on the CCC homepage are the principles of the 'hacker ethic' that Steven Levy reconstructed from participant observation and presented in his 1984 book:[3]

[2] See http://www.ccc.de/satzung

[3] See http://www.ccc.de/de/hackerethik

*Access to computers – and anything which might teach you something about the way the world works – should be unlimited and total.*
*Always yield to the Hands-On Imperative!*
*All information should be free.*
*Mistrust Authority – Promote Decentralization.*
*Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.*
*You can create art and beauty on a computer.*
*Computers can change your life for the better.*
([47]: 40-45)

The CCC homepage, in fact, supplements Levy's hacker ethic with two, more privacy-focused rules: "Mülle nicht in den Daten anderer Leute" (*Do not rummage in other people's data*) and "Öffentliche Daten nützen, private Daten schützen" (*Use public data, protect personal data*).

The foundation of the CCC as a legally registered association in 1986 was preceded by years of informal, loose networking of politically sensitized computer enthusiasts across Germany since 1981. The zeitgeist that permeates the early history of the CCC is that of left-wing community activism, which was inspired by the Frankfurt School and their critical media theory [18, 43]. From the mid-1980s on, members and affiliates of the CCC have been involved in a number of hacks that raised public attention, forced government agencies to react, and entailed law enforcement. Since the mid-1990s, the CCC is widely known to the German public and plays an increasingly important role in German policy debates on information and communication technology [50]. During recent years, the CCC has become an important voice in the European privacy debate ([7]: 53). At present, the association has successfully established itself as a pool of technological expertise for the public, but also government agencies and legal actors.

Its relation to political and legal actors is a characteristic that may distinguish the CCC from large parts of the American hacker milieu. This relation is sometimes confrontational, sometimes collaborative. For example, in 2009 the German Federal Constitutional Court asked the CCC for an official advisory opinion on data retention laws.[4] From 2010 to 2013, Constanze Kurz, a longtime spokeswoman of the association, was an expert member of the parliamentary committee of inquiry on "Internet and digital society" [67].[5] In 2011, members of the CCC reverse-engineered a surveillance program used by German police and demonstrated that the software incorporated unlawful features.[6] In July 2014, the CCC filed a complaint against the German government and others with the German Federal Prosecutor General because one of its servers was explicitly targeted by foreign surveillance agencies who collected user data of German citizens.[7] The CCC's role in national politics, however, relies upon the expertise it gains through hacking practice, a practice that often confronts established political and legal conventions.

[4] See http://ccc.de/de/updates/2009/vds-gutachten
[5] See http://ccc.de/de/updates/2011/adhocracy-enquete
[6] See http://ccc.de/en/updates/2011/staatstrojaner
[7] See http://ccc.de/en/updates/2014/tor-gba

## METHOD

Our case consists in a series of GSM hacks carried out by a cluster of people, some of which are members of the CCC, some of which are only loosely affiliated with the organization. We confine our analysis to those steps of GSM hacking that have been reported at the CCC's annual convention, the *Chaos Communication Congress*,[8] and the CCC's summer camp, which takes place every four years.[9] While the summer camp is a relatively intimate gathering, the convention is a large event. It takes place in Hamburg or Berlin at the end of each year; and it had more than 10,000 reported attendees in 2014.

When we analyze hacking in the context of the CCC, we find it important to note that we are situated observers with a particular 'standpoint' [6]. While we pursue no immediate activist concerns (cf. [34]), we have developed sympathies for the CCC over many years. One author has visited the association's annual convention twice; the other author has visited the convention more than ten times since 2000 and is an (inactive) member of the CCC since 2009. We are not, however, in any way acquainted with the people whose work we describe.

For our case study, we make use of document analysis, a qualitative method frequently used in the analysis of digital sources for history writing [9, 16, 25]. As sources, we identified and analyzed 42 publicly accessible video talks and slide decks that were originally presented and recorded at the annual convention or the summer camp between 2001 and 2014.[10] In our analysis, we included all talks with any relation to mobile telephony. Intended to facilitate knowledge exchange, the talks describe how GSM networks and related technologies can be hacked, detailing challenges, necessary tools, and technical insights gained. To complement talks and slide decks, we have also drawn on convention-specific public wikis, podcast discussions, and news articles. All video talks and slide decks have been authored in English (except for [91]) by people with first-hand experience in the subject matter, who have 'hacked' the technologies in question and self-identify as hackers. Many of these people have been affiliated with the Osmocom project, THC GSM, and/or OpenBTS projects.[11]

With infrastructure as a theoretical lens, we performed a thematic analysis [11] of our sources, probing different themes until stable thematic motifs emerged. One group of motifs clusters around efforts to 'open' proprietary technologies; another group of motifs clusters around efforts to identify security weaknesses, a complex facet of GSM hacking that lies outside the scope of this paper.[12] The thematic analyses of digital sources whose creation stretches over more than a decade

[8] See http://events.ccc.de/congress/
[9] See, e.g., http://events.ccc.de/camp/2011/
[10] In this paper, we only reference a subset of those sources.
[11] The OSMoCom project (http://osmocom.org/) is a network of people interested in building devices and infrastructure for Open Source Mobile Communication. THC (https://wiki.thc.org/gsm/simtoolkit) is a loose group of hackers some of which focused on GSM temporarily. OpenBTS (http://openbts.org/) is an open source project that has evolved into a US-based company.
[12] We refer to [12] for a recent review on security issues and attacks on the GSM standard.

allows us to move beyond the 'here and now,' observing practices that are temporally, locally, and socially dispersed—a necessity when studying how large-scale infrastructures evolve (cf. [52]: 582). In the result of our analysis, we provide a trajectory of GSM hacking as a collective effort, elaborating different ways in which GSM hacking engages with and works upon the sociotechnical infrastructures of mobile telephony. Although we have striven to provide a synthesized account of GSM hacking, we are not suggesting that there exists a consensus or a joint commitment binding all involved individuals together.

## CASE STUDY: EXPLORING AND REBUILDING GSM

Our case study analyzes the ways in which hackers have explored the Global System for Mobile Communications (GSM) and its sociotechnical configuration from 2001 to 2014—years of research and experimentation that eventually enabled the CCC community to operate its own officially registered local GSM network at its annual convention from 2009 on. To describe the trajectory of this development, we shortly elaborate on the challenges of GSM hacking and then characterize the three technical foci that it has pursued.

### Challenges of GSM hacking

GSM hacking targets technologies that are, to a large degree, physically and intellectually inaccessible to the public—in stark contrast to the internet, a decentralized network with publicly available protocol specifications, based on software that is largely open source and on hardware components that are readily commercially available:

> *On the Ethernet/IP based Internet, we are used to Free Software and general-purpose hardware. The worlds second largest communications network GSM couldn't be any more different. Even though the protocols are standardized and publicly available at the ETSI, all implementations are highly-guarded proprietary secrets of a few major players in the industry. The hardware is even more closed, as there is not a single GSM subscriber or base station chipset with even the least bit of publicly known information.* [90]

At the beginning of GSM hacking, hackers had virtually no access to hardware documentation and software source code. While protocol specifications for mobile phone networks are publicly available through, e.g., the European Telecommunications Standards Institute (ETSI), soft- and hardware implementations of GSM are guarded off by few major manufacturers. Only about ten companies build the baseband chips crucial for every mobile phone to communicate with the network, and there exist only about four closed-source embedded operating systems implementing the GSM protocol for baseband chips [95]. Furthermore, hackers face great difficulty in gaining physical access to many hardware components required for a GSM network. Only about four companies build the equipment for GSM networks and handle much of network planning, servicing, and maintenance. Since only network operators buy GSM equipment, quantities are low and prices extremely high [95]. In lack of hard- and software documentation, controllable hardware, and analytic software tools, hackers found themselves in a situation where, as one of them put it, "*a baseband is mostly seen as a blackbox running code for a terrifyingly complex network stack*" [84].

A further challenge of GSM hacking is the fact that the signal space that GSM and related technologies occupy is subject to the strict regulation of national authorities and that by broadcasting unlicensed mobile phone networks hackers potentially interfere with other networks.

### The trajectory of GSM hacking

The series of GSM hacks that we analyze has followed, to use a metaphor of hacking culture, a trajectory of step-by-step 'opening,' driving "*a wedge of Openness*" into GSM infrastructures [90]. GSM hacking literally opens mobile phones and base transceiver stations mounted on cell towers. But as a metaphor, the 'opening' of technology through hacking practices comes with much further-reaching demands for information and control [14]. 'Opening' technological systems means to re-construct and publicly disseminate corporately-owned, proprietary information and expertise. 'Open' systems are technologies whose functioning is publicly documented. Once 'opened,' i.e., once systems are intellectually accessible, hackers will try to control their technical functionalities. Control, on the one hand, enables hackers to exploit and repurpose a system, or to cause its breakdown. On the other hand, degrees of technical control are a precondition for experimentation and further knowledge acquisition. For many hackers in our case, the ultimate goal of these activities is to provide an open source, free code alternative to existing, inaccessible (i.e., 'closed') infrastructures—an alternative that is 'open' to previously excluded purposes and forms of use.

On the basis of our analysis, we are able to describe three technical foci of GSM hacking: (1) end-user facing services (such as call encryption or Voice-over-IP); (2) end-user devices such as mobile phones; (3) and network-side components such as cell towers and their controllers, crucial to operate a mobile phone network (see Figure 1).

### Focus 1: Circumventing service limitations

It is difficult to pinpoint the exact beginnings of GSM hacking. Around 2001, we observe a rising interest in understanding and exploiting technologies related to GSM such as text messages (and their capabilities for sending control commands [85]) and so-called IMSI catchers (which are used by law enforcement to intercept mobile phone traffic and track movements of mobile phone users [91]). These activities prepare for what we describe as a first focus of GSM hacking.

This first focus is concerned with end-user facing services that run on top of the GSM network. Particularly in 2005 and 2006, different groups of hackers sought to circumvent the limitations that German and other mobile phone network operators imposed in order to block services such as call encryption or cheap internet telephony via Voice-over-IP (e.g., Skype). A group of hackers, e.g., investigated the inner workings of the mobile data, i.e., the IP-based side of 3G GSM networks over which VoIP runs, claiming the "*right to talk via voice-over-ip wherever and whenever you want to*" [86]. Another group sought to encrypt telephone calls, arguing that "*we*

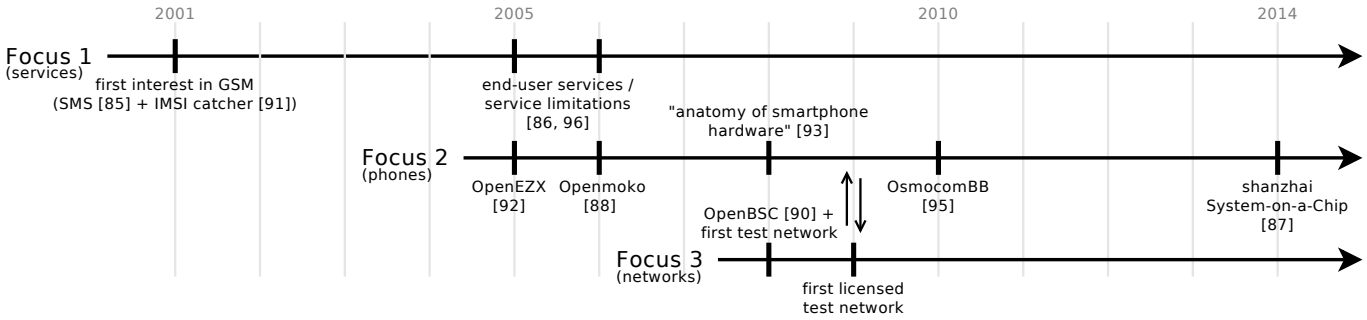


**Figure 1. The trajectory of GSM hacking, showing important presentations and developments at the CCC conventions.**

*can no longer trust the security of the telecom infrastructure*" [96]. The investigation of VoIP and call encryption applied knowledge about internet technologies to mobile telephony, triggering an interest in underlying GSM technologies.

A way to characterize the beginnings of GSM hacking is to describe them as a form of 'infrastructural breakdown'—not a breakdown of technology but a breakdown in user expectation [55]. Hackers lacked some of the services that they, as mobile phone users, expected to have and perceived this as an infringement upon their individual freedom. Their frustration led them to find ways to circumvent the limitations that network operators imposed upon their mobile phone use.

### Focus 2: Opening mobile phones

A second focus of GSM hacking are mobile phones, their hard- and software, and their baseband chip. Even though mobile phones are an interface to the network that hackers can easily get hold of, the expertise that is necessary to understand the inner workings of a phone is typically owned by corporations and not publicly accessible.

Mobile phones are split into two separate subsystems: the application processor-side, which runs the operating system and applications software; and the baseband processor-side, which is responsible for telephony. Hackers have incrementally taken on these elements by focusing on (i) the operating system and other software running on phones (application-side software), (ii) the general phone hardware (application-side hardware), and (iii) the baseband chip in the phone along with its embedded operating system (baseband-side hard- and software).

#### *(i) Application-side software*

Application-side software hacking is facilitated by the fact that application programming interfaces (API) are often provided by phone manufacturers to support the software development of mobile phone applications. Under the label of 'unlocking' or 'jailbreaking,' hacker communities (such as, e.g., xda-developers.com) experiment with APIs to examine the basic functioning of mobile phone application software, trying to rid phones of the software limitations that manufacturers imposed. Often, these efforts involve installing custom systems software and privacy-enhancing tools, efforts that clearly go beyond what companies intend when they provide APIs.

Building, in part, upon knowledge gained through unlocking, hackers associated with the OpenEZX project began to work towards "*the ultimate goal to replace all proprietary applications with 100% free software*" [92]. The project presented its efforts to develop system software, software controlling periphery, and calling and contact applications for a particular Linux-based smartphone at the CCC convention in 2005 [92], hoping that open source software replacements such as these would enable 'third party hacks,' i.e., extensions in functionality by others than the original manufacturer.

#### *(ii) Application-side hardware*

From 2006 to 2007, a member of the CCC, Harald Welte, became Lead System Architect for the Openmoko project.[13] The goal of this project was to develop and sell a completely free smartphone, i.e., a mobile phone all of whose soft- and hardware components are non-proprietary [88]. Founded and sponsored by the Taiwanese manufacturer First International Computer, Inc., the project released its first phone in 2007 and sold it in low quantities mostly to developers and open source enthusiasts.

As substantial parts of mobile phone software had been 'opened,' attention shifted to hardware components. In 2008, Welte presented an "*anatomy of smartphone hardware*" for the Openmoko phone and other Linux-friendly devices at the CCC's annual convention [93]. He detailed the individual major building blocks and overall architecture of contemporary smartphones, highlighting those components for which no 'open' alternative was yet available, i.e., baseband chip and other radio components essential for interfacing with the GSM network.

#### *(iii) Baseband-side hard- and software*

During 2008 and 2009, hackers gained substantial knowledge about the functioning and the operating of GSM networks. These developments, upon which we elaborate in the next section, facilitated a deeper-reaching investigation of mobile phones and their baseband chips, helping to create an open source baseband chip operating system, reverse engineering an existing operating system, and developing a reverse-engineered System-on-a-Chip for applications beyond telephony.

At the CCC convention in 2010, hackers, among them Welte, presented OsmocomBB, an open source baseband chip operating system implementing the GSM protocol [95]. OsmocomBB replaces the proprietary operating system of a partic-

[13] See http://wiki.openmoko.org/

ular type of baseband chip that occurs in a range of mobile phone models. Because the operating system of the baseband chip handles radio communication with the GSM network, OsmocomBB enabled hackers to send arbitrary data at all levels of the GSM protocol stack. (Previously, hackers were only able to control high-level, predefined actions, such as sending a text message or calling a number.) To send arbitrary data at all levels of the protocol stack is important for network security analyses and further network experiments. As hackers phrase it, with OsmocomBB "*the security researcher finally has a tool equivalent to an Ethernet card in the TCP/IP [internet] protocol world: A simple transceiver that will send arbitrary protocol messages to a GSM network*" [95]. Building upon OsmocomBB, hackers demonstrated, in 2012, how mobile phones themselves could act as a GSM base transceiver station (a crucial component, usually, of cell towers) to broadcast their own network [89].

In 2011, hackers reverse engineered a different baseband chip operating system, taken from a 3G USB mobile internet stick [84]. While the OsmocomBB project replaced a 'closed' operating system with an 'open' one, this instance of reverse engineering aimed at what an involved hacker called "*the final exploitation*" [84]—i.e., the exploitation of security flaws in order to execute one's own code within the existing operating system. The ability to execute own code on a foreign system is a prerequisite for further analysis and experimentation.

In 2014, another hacking project reverse engineered a cheap System-on-a-Chip (SoC) that is widely available in China and used in many of the *shanzhai* (i.e., knock-off or counterfeit) value phones [87]. A SoC integrates the baseband chip with most other phone capabilities from the application-side. Instead of focusing on telephony, however, the project built an open development kit with the reverse-engineered SoC at its core, making GSM available for applications beyond telephony. The project demonstrates, involved hackers argue, how "*to lawfully import IP [intellectual property] from the Shanzhai ecosystem into the Maker ecosystem*," thereby "*establish[ing] a repeatable [..] model for opening up previously closed IP [..], leveling the playing field for lawful Makers*" [87].

**Focus 3: Operating GSM networks**

Incrementally, hackers succeeded in building up the expertise required to run their own networks. Hackers affiliated with the CCC have, in fact, been setting up local GSM networks on various occasions since at least 2009.

To operate a mobile phone network, hackers needed to acquire, understand, and control crucial hardware components: the base transceiver station (BTS), typically mounted on the cell tower; the base station controller (BSC) that controls up to hundreds of BTS; and the mobile switching center (MSC). Such network equipment is, for non-corporate actors, hardly available and very expensive. After hackers obtained the necessary hardware, partly through loans from universities and companies [94], hackers began experimenting with test networks. In the following, we shortly describe how the independent operation of a GSM network was first demonstrated (i), and how GSM network operation subsequently evolved into a service that the CCC offered at its annual convention (ii).

*(i) Demonstrating network operation*

At the CCC convention in 2008, a group of hackers publicly released OpenBSC, a free software re-implementation providing the minimally necessary functionalities for GSM network equipment [90]. With two discarded, industry-grade BTS and other test equipment bought on eBay, the group demonstrated a live phone call between two mobile phones that were registered to a self-operated GSM base station. This represented the first successful step of the hacker community to run their own GSM network. The OpenBSC project also operated a temporary test network at the 2008 convention, serving at times more than 1,000 users simultaneously. The purpose of this test network was to gain experience with the OpenBSC software under real use conditions. Convention attendees were invited to join the network and experiment with it.

Because GSM networks operate in a licensed radio spectrum, the OpenBSC project cautioned convention attendees against operating their private GSM networks: "*don't try this at home!*"—and if, then "*don't interfere with the operators!*" [90], a warning frequently echoed by other presenters, e.g., [89]: "*act responsibly*."

Since 2010, CCC affiliates have been able to fully control the two major components of GSM. OpenBSC enables to control the network itself; and OsmocomBB, the open source baseband chip operating system, enables to control mobile phones. Together, control of these two components allows to send arbitrary data from phone to network and back, enabling hackers to govern the operation of GSM networks fully.

*(ii) Productive network operation*

At the CCC convention in 2009, the group of hackers associated with OpenBSC operated a temporary GSM test network open to convention attendees, this time with an official spectrum license for research purposes granted by the Bundesnetzagentur, the responsible regulatory authority [82]. Already a few months earlier, the group had operated a GSM network at a Dutch outdoor hacker camp, with antennas and BTS strapped to trees and the computer running OpenBSC placed under a tent [94].

From 2009 on, local GSM networks have been in operation during the convention, and new network features have been implemented year after year. The convention has successively integrated the operation of its GSM network into network management practices that evolved around LAN, WLAN, and landline phone networks earlier. A 'mobile operation center' coordinates network operation, organizes the registration of users, and offers technical support. Convention attendees have to bring a SIM card and receive a temporary phone number that they can use during the event. Network users can use the internal network as well as call, and be called by, external landline and mobile phone numbers in commercial telephone networks. They can use the GSM network at the convention in roughly the same way as they use any other mobile phone network—the network finally has become a genuine *infrastructure*, as unstable and primitive as it may be. At the same time, the convention's GSM network remains a site of experimentation for many convention attendees [83], a toy network for playful exploration, occasional breakdowns included.

| hacking strategy | form(s) of appropriation |
|---|---|
| reverse engineering | intellectual |
| re-implementation | intellectual, legal, functional |
| parallel operation | operational |

**Table 1. Hacking strategies and their dominant forms of appropriation.**

## HACKING STRATEGIES FOR APPROPRIATION

We identify three strategies that hackers have made use of in exploring and rebuilding GSM networks—reverse engineering, re-implementation, and parallel operation.

As we observe it, these strategies require substantial expertise and sustained collaborative effort. These strategies are transgressive: They require hackers to violate if not laws and property rights, so conventions of use and established configurations of economical and technological power. Over the course of years, these strategies have supported one another in an iterative cycle; and taken together, they have enabled hackers to 'open' proprietary, corporately owned and operated technologies.

All three strategies facilitate, as we will show, different forms of appropriation of the GSM network. These forms of appropriation do not point to entirely novel aspects, and facets of them have been discussed in the literature before [2, 20, 48, 60, 70, 79]. What we contribute with this distinction, however, is a systematic framework for studying appropriation—in the context of hacking, with the analytic lens of infrastructure.

We distinguish three forms of appropriation: intellectual appropriation, legal appropriation, functional appropriation, and operational appropriation (see Table 1). With *intellectual appropriation* we refer to the acquisition of knowledge about the functioning of an adopted system or artifact. With *legal appropriation* we refer to the acquisition of lawful proprietorship over a system or artifact, and, crucially, over the knowledge required to build, repair, and maintain a system or artifact autonomously. And while *functional appropriation* allows to technically control system functionalities, i.e., to control what a system does and does not do, *operational appropriation* allows to control where and how the system is put to use, in which contexts or settings, and by whom. As we will see, these forms of appropriation are both means and end. Through capacity building, they help working towards the vision of 'open' technology and are part of that vision at the same time.

### Reverse engineering

Much of the knowledge that hackers need to set up local GSM networks stems from reverse engineering efforts, which initially focused on mobile phones and were driven by the desire to unlock or 'jailbreak' them. As we have shown, GSM hacking has applied reverse engineering techniques also to, e.g., the operating system of baseband chips and GSM network equipment. Through reverse engineering, hackers have been able to familiarize themselves with 'closed' systems, and we thus characterize reverse engineering as a strategy of *intellectual appropriation*.

Reverse engineering is a strategy to investigate complex technical systems whose functioning, although known to its designers, is unknown to the investigator. As Rekoff explains, reverse engineering means "developing a set of specifications for a complex hardware system by an orderly examination of specimens of that system," an effort undertaken "by persons other than the original designers" and "without the benefit of any of the original drawings or other documentation" ([58]: 244). The purpose of reverse engineering is to "mak[e] a clone of the original hardware system" ([58]: 244).

Reverse engineering, originally applied to hardware and later to software, is a commonplace practice of manufacturers to improve their own products and to analyze the products of corporate competitors or military adversaries [13]. It has also been recognized as a prime strategy of hackers [33]. Surprisingly, however, accounts of reverse engineering practices appear to be largely absent in HCI and CSCW research.

As we have observed for the case of GSM hacking, reverse engineering commonly involves several steps of investigation: After an identification of relevant components and subcomponents, concerning which all available information is collected, the investigator observes and experiments with input/output relations. This is a process of trial and error that explores 'what happens when' an input variable is changed. When hackers reverse engineer, they may also use a system's security flaws in order to inject own code or reprogram parts of the system such as to further elucidate its functioning.

Given the complexity of GSM technology, reverse engineering efforts have to be strategically focused. For this reason, hackers have adopted what they described as a "*lazy approach*" [95]. To implement OsmocomBB, their open source baseband chip operating system, hackers decided not to build GSM hardware from scratch. Instead, they implemented their operating system upon an existing chip. To do so, however, they had to have precise knowledge about the chip and the way it is integrated into the mobile phone that carries it. They could gain this knowledge more efficiently by focusing their reverse engineering efforts upon a particular type of mobile phone, a 'model system.' The selection of this model phone, however, had to take into account two different concerns regarding technical complexity and economic availability. On the one hand, hackers sought to minimize technological complexity. They were inclined to choose a phone model that was older and as simple as possible, and for which crucial information about its baseband chip had already been leaked. On the other hand, hackers also sought to maximize physical availability and therefore considered phone models that are cheap and widely sold—choosing, eventually, the Motorola C123.

As this example shows, reverse engineering prepares for the integration of 'own,' new soft- or hardware into an existing system. As a strategy of intellectual appropriation, thus, reverse engineering provides some of the technical understanding required by re-implementation and parallel network operation.

### Re-implementation

With re-implementation we describe the development of 'open' hard- or software components that substitute proprietary components of existing infrastructures. To achieve this goal, re-implementations must be, to some degree, congruent in func-

tionality with those systems they substitute and conform to existing standards and specifications. Without a minimum of congruence and conformity re-implementations cannot be embedded into existing systems and integrated successfully with other infrastructural components. For this reason, re-implementation builds on knowledge gained through the reserve engineering of existing systems; and the tentative, selective re-implementation of single sub-systems is, in turn, an important element in reverse engineering technological systems.

When the aim is to understand and control a complex system, re-implementation is an iterative process in the course of which the functionalities of different components are gradually fine-tuned to one another. With time, some components will be stabilized and tentative functionalities morph into viable solutions—solutions that may have to be revised at a later point when further components of the system are modified, when the system is put to use, or when the conditions of use change. Re-implementations in a GSM network that work well for a small number of users may fail with larger user numbers. This is one of the reasons why it is important for the hacker community to have a large-scale temporary test network at the CCC's annual convention.

Two important examples of free software re-implementation in the context of GSM are OsmocomBB and OpenBSC. Both components are 'open' insofar as their code is published online and intellectually accessible to the hacker community as well as interested non-hackers. But re-implementation is a strategy not only of intellectual, but also of *functional* and, possibly, *legal appropriation*. The purpose of re-implementing OsmocomBB and OpenBSC is not only to understand but also to control the functioning of crucial elements of a GSM network. Because OsmocomBB and OpenBSC are pieces of software that have been created independently of corporate expertise, the hacker community may be able to establish intellectual ownership of crucial network elements legally (cf. [87]). OsmocomBB and OpenBSC are licensed as 'free.' Free soft- or hardware usually allows anyone to re-use, adapt, and distribute source material as well as products (sometimes under certain conditions) [14]. We observe that to legally appropriate elements of communication infrastructure through the re-implementation of free soft- or hardware may put intellectual and functional appropriation on a much more solid foundation than the (illicit) exploitation of security flaws in existing, proprietary software would.

#### Parallel operation

With parallel operation we refer to the autonomous operation of functionally similar infrastructures in parallel to existing ones. Our example is the temporary local GSM network that members of the CCC have been operating at every annual convention since 2009, and that all convention attendees are invited to join. As we have pointed out, by providing real-use conditions on growing scales, parallel network operation complements strategies of reverse engineering and re-implementation.

Through parallel operation, hackers seek to install a network that can be *operationally appropriated*, gaining control over the conditions of where, how, and by whom the network is put to use. In contrast to professional engineering and HCI research, the primary purpose of parallel operation is not necessarily to build a functionally better or different network, or to innovate new forms of communication. In fact, in the case of GSM hacking, parallel network operation is characterized by a high level of compatibility with existing, corporately-operated networks, one reason being that hackers try to reuse and repurpose many hard- and software components, another reason being the intention to facilitate inter-network communication.

## DISCUSSION: HACKING AS INFRASTRUCTURING

GSM hacking, as we observe it, consists in experimenting with proprietary systems, claiming corporate knowledge, challenging market structures, negotiating with regulators, and operating independent mobile phone networks. Faced with the range and the subversive thrust of these activities, activities that concern themselves with 'the rules of the game,' we have mobilized the notion of infrastructure as a theoretical lens for our analysis.

GSM hacking is infrastructuring, we argue, not just because involved hackers accumulate enough knowledge to interfere with existing GSM networks and acquire the ability to set up independent networks—and not because GSM hacking has played a major role in developing an open source mobile phone with the Openmoko project. Rather, GSM hacking is infrastructuring because it tackles the infrastructure of mobile telephony in its full scope, exploring diverse network components with regard to their infrastructural function. GSM hacking is infrastructuring because it renders invisible network components visible and amenable to experimentation, problematizing implicit configurations of economic and technological power. Moreover, GSM hacking disseminates the knowledge crucial to set up and operate independent networks, thereby reverberating with infrastructural landscapes in far-reaching ways.

The outcomes of GSM hacking are not confined to the operation of a temporary network at the CCC's convention. As hitherto proprietary knowledge is disseminated, GSM hacking has helped communities operate mobile phone networks where corporate network providers are unwelcome or where corporations have no economic interests (e.g., in rural Mexico [78]). Furthermore, as independent GSM networks, licensed or not, use signal space and interfere with other network providers, they force regulatory authorities to react (cf. [80]). And, as the infrastructure of GSM is rendered more visible and accessible, security vulnerabilities emerge in systems that are, at first glance, not related to mobile telephony (such as, e.g., the unlocking mechanism in recent BMW cars that utilizes GSM capabilities [64]).

#### How to infrastructure?

Given the reverberations of GSM hacking, let us return to the core of hacking activities that originally set these effects off. The existing literature on hacking in CSCW and HCI has described various hacking techniques such as, e.g., ad-hoc tool making through ingenious modifying and 'kludging' of readily available resources [5, 53]. These are fine-grained descriptions

that we, in the result of our analysis, complement with a wider angle when we distinguish three hacking strategies—reverse engineering, re-implementation, and parallel operation. In the case of GSM hacking, these three strategies have enabled hackers to explore and modify the 'closed,' proprietary components of GSM networks, explorations that are if not violations of property rights so transgressions of market conventions and design intentions. However, it is precisely through transgression that hacking, as we have observed it, seeks to initiate change on infrastructural level.

Parallel operation, in particular, is a strategy of infrastructuring that supplements the ways in which recent activist research in HCI has sought to address infrastructural issues [42]. Such research has proposed to fill infrastructural 'gaps' [51], or to 'graft' a communication infrastructure 'on top' of an existing one [34], even parasitically [30, 31]. Parallel operation, instead, circumvents the existing infrastructure to a large degree, offering more independence but also requiring substantially more work.

#### The 'open' imaginary of GSM hacking

GSM hacking, as a transgressive 'playing with the rules of the game,' is underscored by the sociotechnical *imaginary* of 'open' technology. Following Jasanoff and Kim, we understand imaginaries as "collectively imagined forms of social life and social order reflected in the design and fulfillment of [..] technological projects" ([36]: 120). The imaginary of 'open' technology has emerged from the open source or free code movement and rejects exclusive intellectual property rights, particularly for vital components of infrastructure (cf. [14]). 'Open' technology is transparent, its blueprints accessible—a condition which renders technology 'open' for experimentation and change. In the case of GSM hacking, we can observe how this imaginary motivates the transgressive appropriation of corporately-owned technology.

As our case study shows, many hackers perceive the socioeconomic configuration of mobile telephony as problematic because it is 'closed.' A handful of international corporations dominate network operation and hold exclusive intellectual property rights for network equipment and phone components, an oligopoly that secures them technological power and financial profit. Exclusive intellectual property rights prevent actors without these rights, i.e., 'the public,' from gaining insights in the functioning of mobile phone technology and limit the ways in which mobile telephony can be used. In this situation, GSM hacking challenges the for-profit, corporate, and proprietary character of much of mobile phone technology.

As Coleman and Golub [15] have argued, the transgressive 'opening' of technology is motivated by a liberal notion of freedom. In the trajectory of GSM hacking, we notice a shift from emphasis on negative individual freedom (i.e., the freedom of the device-owning user from service limitations) to collective positive freedom (i.e., the freedom of communities to operate their own mobile networks independent of corporate providers). Claiming the freedom to operate GSM networks independently is, as we see it, essentially a claim for operational appropriation.

#### Control shift: Who infrastructures? For whom?

Hacking, framed as infrastructuring, can transform the research questions we discuss. While Star and Ruhleder asked 'for whom is infrastructure?' [66], the case of hacking forces us to reconsider a different version of this question: *Who* infrastructures—and to the benefit, or detriment, of whom?

When hackers embed the practice of GSM hacking in the imaginary of 'openness,' they create a narrative about 'closed,' proprietary technological knowledge that should be 'opened,' benefitting thereby not only hacking communities but the public at large. Yet, research in CSCW and HCI has questioned the visions of public empowerment conjured up sometimes by hackers, sometimes by their observers, and it has pointed out that hacking, along with related practices of making and DIY, creates new dynamics of social exclusion [69]. As Tanenbaum and collaborators [70] note, practices of making require access to economic, technical, social, and intellectual resources that are unevenly distributed—a requirement that may make making practices a "hobby for the privileged" that is "increasingly co-opted by corporate interests" ([3]: 1087).

Beyond the imaginary of 'openness' and narratives of empowerment, however, there are other ways to conceive of hacking. Social theorist Plönges [56] suggests to understand hacking as practices that seek to disturb control(led) systems, causing disturbances that re-allocate possibilities of control. Communication infrastructures are such control(led) systems. They are controlled insofar as they are highly regulated and contracted; and at the same time, they strictly control the ways in which they can be made use of. In fact, what we observe in the case of GSM hacking is the attempt to gain control—control over the operation of mobile phone networks—and to shift control away from the oligopoly of manufacturers and network providers. Still, for the time being, it remains an open question whether GSM hacking entails an effective, long-term control shift and which actors eventually benefit from it.

#### Hacking as subversive appropriation

In the context of GSM hacking, we study appropriation as the processes through which individuals or collectives acquire the capacity to act upon and through technology as desired. As we have elaborated, there are four dimensions to this capacity. With intellectual appropriation we describe the acquisition of technical knowledge, and we describe the acquisition of proprietorship over technology and/or knowledge as legal appropriation. What we describe as functional and operational appropriation concerns the technical control of a technology's features and the control of its functioning in varying contexts of use.

As we conceive of appropriation as an aspect of infrastructuring, we argue that our analysis of hacking as transgressive infrastructuring fruitfully complements existing studies of appropriation in the fields of CSCW and HCI in several respects. First, existing research discusses appropriation as the adaptation of artifacts and particularly end user-facing interfaces of, e.g., organizational information systems or software platforms (see, e.g., [4, 21]). In contrast, we observe appropriation as referring to elements of large-scale communication networks. Second, the forms of technology appropriation we observe are

subversive, i.e., they are not only unanticipated [62, 73] but illicit from the point of view of those who design and provide the technology. Third, the forms of appropriation we observe are less about mediating between technological innovation and existing practices of use; rather, they are about shaping the infrastructural configuration of existing technologies such as to comply with a not-yet-realized sociotechnical imaginary, thereby highlighting the cultural dimension of appropriation (cf. [48]).

## CONCLUSION

In this paper, we suggest to mobilize the analytic lens of infrastructure to study hacking practices that take issue with large-scale communication networks. As a processual concept, 'infrastructure' helps us account for the changing technical foci of GSM hacking, for the strategies by which it tackles these, and for the forms of infrastructural appropriation it seeks to achieve. As a relational concept, 'infrastructure' helps us to understand how the character of technological systems changes through hacking practice—GSM hacking turns mobile phone infrastructure into an object of manipulation and experimentation (and converts it back into infrastructure again). And as a sociotechnical concept, 'infrastructure' sensitizes us for the role that the imaginary of 'open' technology plays in the motivation and justification of transgressive hacking practice.

## REFERENCES


1. Syed Ishtiaque Ahmed, Nusrat Jahan Mim, and Steven J. Jackson. 2015. Residual Mobilities: Infrastructural Displacement and Post-Colonial Computing in Bangladesh. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 437–446.
2. Binaebi Akah and Shaowen Bardzell. 2010. Empowering Products: Personal Identity Through the Act of Appropriation. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems (CHI EA '10)*. ACM, New York, NY, USA, 4021–4026.
3. Morgan G. Ames, Jeffrey Bardzell, Shaowen Bardzell, Silvia Lindtner, David A. Mellis, and Daniela K. Rosner. 2014. Making cultures: empowerment, participation, and democracy—or not?. In *Proceedings of the Extended Abstracts of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 1087–1092.
4. Ellen Balka and Ina Wagner. 2006. Making Things Work: Dimensions of Configurability As Appropriation Work. In *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work (CSCW '06)*. ACM, New York, NY, USA, 229–238.
5. Jeffrey Bardzell, Shaowen Bardzell, and Austin Toombs. 2014. Now that's definitely a proper hack: self-made tools in hackerspaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 473–476.
6. Shaowen Bardzell. 2010. Feminist HCI: Taking Stock and Outlining an Agenda for Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 1301–1310.
7. Colin J. Bennett. 2008. *The Privacy Advocates: Resisting the Spread of Surveillance*. MIT Press, Cambridge.
8. Erling Björgvinsson, Pelle Ehn, and Per-Anders Hillgren. 2010. Participatory Design and "Democratizing Innovation". In *Proceedings of the 11th Biennial Participatory Design Conference (PDC '10)*. ACM, New York, NY, USA, 41–50.
9. Glenn A. Bowen. 2009. Document analysis as a qualitative research method. *Qualitative Research Journal* 9, 2 (2009), 27–40.
10. Geoffrey C. Bowker and Susan Leigh Star. 1999. *Sorting Things Out: Classification and its Consequences*. MIT Press, Cambridge.
11. Richard E. Boyatzis. 1998. *Transforming Qualitative Information: Thematic Analysis and Code Development*. Sage, Thousand Oaks.
12. Giuseppe Cattaneo, Giancarlo De Maio, and Umberto Ferraro Petrillo. 2013. Security Issues and Attacks on the GSM Standard: A Review. *Journal of Universal Computer Science* 19, 16 (2013), 2437–2452.
13. Elliot J. Chikofsky and James H. Cross II. 1990. Reverse engineering and design recovery: a taxonomy. *IEEE Software* 7, 1 (1990), 13–17.
14. Gabriella Coleman. 2013. *Coding Freedom. The Ethics and Aesthetics of Hacking*. Princeton University Press, Princeton and Oxford.
15. Gabriella Coleman and Alex Golub. 2008. Hacker practice: Moral genres and the cultural articulation of liberalism. *Anthropological Theory* 8, 3 (2008), 255–277.
16. Lyell Davies and Elena Razlogova. 2013. Framing the contested history of digital culture. *Radical History Review* 2013, 117 (2013), 5–31.
17. Lorenzo Davoli and Johan Redström. 2014. Materializing infrastructures for participatory hacking. In *Proceedings of the 2014 Conference on Designing Interactive Systems (DIS '14)*. ACM, New York, NY, USA, 121–130.
18. Kai Denker. 2014. Heroes Yet Criminals of the German Computer Revolution. In *Hacking Europe. From Computer Cultures to Demoscenes*, Gerard Alberts and Ruth Oldenziel (Eds.). Springer, London, Chapter 8, 167–187.
19. Yvonne Dittrich, Sara Eriksén, and Christina Hansson. 2002. PD in the Wild: Evolving Practices of Design in Use. In *Proceedings of the 7th Biennial Participatory Design Conference (PDC '02)*. CPSR, Palo Alto, 124–134.
20. Paul Dourish. 2003. The Appropriation of Interactive Technologies: Some Lessons from Placeless Documents. *Computer Supported Cooperative Work* 12, 4 (2003), 465–490.

21. Sebastian Draxler and Gunnar Stevens. 2011. Supporting the collaborative appropriation of an open software ecosystem. *Computer Supported Cooperative Work* 20, 4-5 (2011), 403–448.

22. Nicolas Ducheneaut. 2005. Socialization in an Open Source Software Community: A Socio-Technical Analysis. *Computer Supported Cooperative Work* 14, 4 (2005), 323–368.

23. Pelle Ehn. 2008. Participation in Design Things. In *Proceedings of the Tenth Anniversary Conference on Participatory Design (PDC '08)*. ACM, New York, NY, USA, 92–101.

24. Stephen Flowers. 2008. Harnessing the hackers: The emergence and exploitation of Outlaw Innovation. *Research Policy* 37, 2 (2008), 177–193.

25. Meg Foster. 2014. Online and Plugged In?: Public History and Historians in the Digital Age. *Public History Review* 21 (2014), 1–19.

26. Sarah Fox, Rachel Rose Ulgado, and Daniela K. Rosner. 2015. Hacking Culture, Not Devices: Access and Recognition in Feminist Hackersaces. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM, New York, NY, USA, 56–68.

27. Christiane Funken. 2010. Der Hacker. In *Diven, Hacker, Spekulanten – Sozialfiguren der Gegenwart*, Stephan Moebius and Markus Schroer (Eds.). Suhrkamp, Frankfurt, 190–205.

28. Katie Hafner and John Markoff. 1991. *Cyberpunk: Outlaws and Hackers on the Computer Frontier*. Simon and Schuster, New York.

29. Bjorn Hartmann, Scott Doorley, and Scott R. Klemmer. 2008. Hacking, mashing, gluing: Understanding opportunistic design. *IEEE Pervasive Computing* 7, 3 (2008), 46–54.

30. Tad Hirsch. 2009. Communities Real and Imagined: Designing a Communication System for Zimbabwean Activists. In *Proceedings of the Fourth International Conference on Communities and Technologies (C&T '09)*. ACM, New York, NY, USA, 71–76.

31. Tad Hirsch and John Henry. 2005. TXTmob: Text Messaging for Protest Swarms. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1455–1458.

32. Lara Houston. 2013. *Inventive Infrastructure: An Exploration of Mobile Phone 'repair' cultures in Kampala, Uganda*. Ph.D. Dissertation. Lancaster University.

33. Gary R. Ignatin. 1992. Let the Hackers Hack: Allowing the Reverse Engineering of Copyrighted Computer Programs to Achieve Compatibility. *University of Pennsylvania Law Review* 140, 5 (1992), 1999–2050.

34. Lilly C. Irani and M. Six Silberman. 2013. Turkopticon: Interrupting Worker Invisibility in Amazon Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 611–620.

35. Steven J. Jackson and Laewoo Kang. 2014. Breakdown, obsolescence and reuse: HCI and the art of repair. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 449–458.

36. Sheila Jasanoff and Sang-Hyun Kim. 2009. Containing the atom: Sociotechnical imaginaries and nuclear power in the United States and South Korea. *Minerva* 47, 2 (2009), 119–146.

37. Tim Jordan. 2008. *Hacking. Digital Media and Technological Determinism*. Polity Press, London.

38. Helena Karasti. 2014. Infrastructuring in Participatory Design. In *Proceedings of the 13th Participatory Design Conference (PDC '14)*. ACM, New York, NY, USA, 141–150.

39. Helena Karasti and Karen S. Baker. 2004. Infrastructuring for the long-term: ecological information management. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS '04)*. IEEE, 10 pp.

40. Helena Karasti, Karen S. Baker, and Florence Millerand. 2010. Infrastructure Time: Long-term Matters in Collaborative Development. *Computer Supported Cooperative Work* 19, 3-4 (2010), 377–415.

41. Helena Karasti and Anna-Liisa Syrjänen. 2004. Artful Infrastructuring in Two Cases of Community PD. In *Proceedings of the Eighth Conference on Participatory Design (PDC '04)*. ACM, New York, NY, USA, 20–30.

42. Matthias Korn and Amy Voida. 2015. Creating Friction: Infrastructuring Civic Engagement in Everyday Life. In *Proceedings of the 5th Decennial Aarhus Conference: Critical Alternatives*. ACM, New York, NY, USA, 145–156.

43. Daniel Kulla. 2003. *Der Phrasenprüfer: Szenen aus dem Leben von Wau Holland, Mitbegründer des Chaos Computer Clubs*. Pieper & The Grüne Kraft, Löhrbach.

44. Stacey Kuznetsov, Alex S. Taylor, Tim Regan, Nicolas Villar, and Eric Paulos. 2012. At the seams: DIYbio and opportunities for HCI. In *Proceedings of the Designing Interactive Systems Conference (DIS '12)*. ACM, New York, NY, USA, 258–267.

45. Christopher A. Le Dantec and Carl DiSalvo. 2013. Infrastructuring and the formation of publics in participatory design. *Social Studies of Science* 43, 2 (2013), 241–264.

46. Charlotte P. Lee, Paul Dourish, and Gloria Mark. 2006. The Human Infrastructure of Cyberinfrastructure. In *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work (CSCW '06)*. ACM, New York, NY, USA, 483–492.

47. Steven Levy. 1994 [1984]. *Hackers. Heroes of the Computer Revolution*. Dell Publishing, New York.

48. Silvia Lindtner, Ken Anderson, and Paul Dourish. 2012. Cultural Appropriation: Information Technologies As Sites of Transnational Imagination. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW '12)*. ACM, New York, NY, USA, 77–86.

49. Silvia Lindtner, Garnet D. Hertz, and Paul Dourish. 2014. Emerging sites of HCI innovation: hackerspaces, hardware startups & incubators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 439–448.

50. Maria Löblich and Manuel Wendelin. 2012. ICT policy activism on a national level: Ideas, resources and strategies of German civil society in governance processes. *New Media & Society* 14, 6 (2012), 899–915.

51. Scott D. Mainwaring, Michele F. Chang, and Ken Anderson. 2004. Infrastructures and Their Discontents: Implications for Ubicomp. In *Proceedings of the 6th International Conference on Ubiquitous Computing (Ubicomp '04)*. Springer, Berlin, 418–432.

52. Eric Monteiro, Neil Pollock, Ole Hanseth, and Robin Williams. 2013. From Artefacts to Infrastructures. *Computer Supported Cooperative Work* 22, 4-6 (2013), 575–607.

53. Joseph A. Paradiso, John Heidemann, and Thomas G. Zimmerman. 2008. Hacking is pervasive. *IEEE Pervasive Computing* 7, 3 (2008), 13–15.

54. Claus Pias. 2002. Der Hacker. In *Grenzverletzer. Figuren politischer Subversion*, Eva Horn, Stefan Kaufmann, and Ulrich Bröckling (Eds.). Kadmos, Berlin, 248–270.

55. Volkmar Pipek and Volker Wulf. 2009. Infrastructuring: Toward an Integrated Perspective on the Design and Use of Information Technology. *Journal of the Association for Information Systems* 10, 5 (2009), 447–473.

56. Sebastian Plönges. 2012. Versuch über Hacking als soziale Form. In *Shift. #Globalisierung, #Medienkulturen, #Aktuelle Kunst*, Christine Heil, Gila Kolb, and Torsten Meyer (Eds.). Kopaed, München, 81–91.

57. Neil Pollock and Robin Williams. 2010. e-Infrastructures: How Do We Know and Understand Them? Strategic Ethnography and the Biography of Artefacts. *Computer Supported Cooperative Work* 19, 6 (2010), 521–556.

58. M. G. Rekoff. 1985. On reverse engineering. *IEEE Transactions on Systems, Man and Cybernetics* 15, 2 (1985), 244–252.

59. David Ribes and Charlotte P. Lee. 2010. Sociotechnical Studies of Cyberinfrastructure and e-Research: Current Themes and Future Trajectories. *Computer Supported Cooperative Work* 19, 3-4 (2010), 231–244.

60. Daniela Rosner and Jonathan Bean. 2009. Learning from IKEA hacking: I'm not one to decoupage a tabletop and call it a day. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 419–422.

61. Daniela K. Rosner, Silvia Lindtner, Ingrid Erickson, Laura Forlano, Steven J. Jackson, and Beth Kolko. 2014. Making Cultures: Building Things & Building Communities. In *Proceedings of the Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW Companion '14)*. ACM, New York, NY, USA, 113–116.

62. Antti Salovaara, Sacha Helfenstein, and Antti Oulasvirta. 2011. Everyday appropriations of information technology: A study of creative uses of digital cameras. *Journal of the American Society for Information Science and Technology* 62, 12 (2011), 2347–2363.

63. Rita Shewbridge, Amy Hurst, and Shaun K. Kane. 2014. Everyday making: identifying future uses for 3D printing in the home. In *Proceedings of the 2014 Conference on Designing Interactive Systems (DIS '14)*. ACM, New York, NY, USA, 815–824.

64. Dieter Spaar. 2015. Beemer, Open Thyself! – Security vulnerabilities in BMW's ConnectedDrive. *c't Magazin* 15, 5 (2015), 86ff. Trans.: F. A. Scherschel, http://heise.de/-2540957.

65. Susan Leigh Star and Geoffrey C. Bowker. 2006. How to infrastructure. In *Handbook of New Media*, Leah A. Lievrouw and Sonia Livingstone (Eds.). Sage, Thousand Oaks, 230–245.

66. Susan Leigh Star and Karen Ruhleder. 1996. Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. *Information Systems Research* 7, 1 (1996), 111–134.

67. Falk Steiner. 2013. Die Enquete-Kommission "Internet und Digitale Gesellschaft": Zivilgesellschaftlich ein Erfolg – oder ein Ausfall? *Forschungsjournal Soziale Bewegungen* 26, 2 (2013), 83–88.

68. Kevin F. Steinmetz. 2014. Craft(y)ness. An Ethnographic Study of Hacking. *British Journal of Criminology* 55, 1 (2014), 125–145.

69. Yuling Sun, Silvia Lindtner, Xianghua Ding, Tun Lu, and Ning Gu. 2015. Reliving the Past & Making a Harmonious Society Today: A Study of Elderly Electronic Hackers in China. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM, New York, NY, USA, 44–55.

70. Joshua G. Tanenbaum, Amanda M. Williams, Audrey Desjardins, and Karen Tanenbaum. 2013. Democratizing technology: pleasure, utility and expressiveness in DIY and maker practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2603–2612.

71. Paul Taylor. 1999. *Hackers. Crime in the Digital Sublime*. Routledge, London.

72. Douglas Thomas. 2002. *Hacker Culture*. University of Minnesota Press, Minneapolis.

73. Manfred Tscheligi, Alina Krischkowsky, Katja Neureiter, Kori Inkpen, Michael Muller, and Gunnar Stevens. 2014. Potentials of the "Unexpected": Technology Appropriation Practices and Communication Needs. In *Proceedings of the 18th International Conference on Supporting Group Work (GROUP '14)*. ACM, New York, NY, USA, 313–316.

74. Orly Turgeman-Goldschmidt. 2005. Hackers' Accounts: Hacking as a Social Entertainment. *Social Science Computer Review* 23, 1 (2005), 8–23.

75. Sherry Turkle. 2005 [1984]. *The Second Self: Computers and the Human Spirit*. MIT Press, Cambridge. Twentieth Anniversary Edition.

76. Eric von Hippel. 2005. *Democratizing Innovation*. MIT Press, Cambridge.

77. Eric von Hippel and Joseph A. Paradiso. 2008. User Innovation and Hacking. *IEEE Pervasive Computing* 7, 3 (2008), 66–69.

78. Lizzie Wade. 2015. Where Cellular Networks Don't Exist, People Are Building Their Own. *Wired* (Jan. 14, 2015). http://www.wired.com/2015/01/diy-cellular-phone-networks-mexico/.

79. Tricia Wang and Joseph Kaye. 2011. Inventive leisure practices: understanding hacking communities as sites of sharing and innovation. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11)*. ACM, New York, NY, USA, 263–272.

80. Richmond Y. Wong and Steven J. Jackson. 2015. Wireless Visions: Infrastructure, Imagination, and US Spectrum Policy. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM, New York, NY, USA, 105–115.

81. Thomas G. Zimmerman. 2008. Hacking in Industrial and Research Development. *IEEE Pervasive Computing* 7, 3 (2008), 16–23.

## SOURCE MATERIAL

All websites in this document last accessed May 2015. Audio and/or video recordings of presentations available from http://media.ccc.de/browse/congress/ and http://media.ccc.de/browse/conferences/camp20{03,07,11}/.

82. 26C3 Public Wiki. 2009. GSM Project / 26C3 GSM Network. (2009). http://events.ccc.de/congress/2009/gsm.html.

83. 27C3 Public Wiki. 2010. GSM Network at 27C3. (2010). https://web.archive.org/web/20130823060820/http://events.ccc.de/congress/2010/wiki/GSM, archived version.

84. Guillaume Delugré. 2011. Reverse-engineering a Qualcomm baseband. In *28C3*. Berlin, CCC. http://events.ccc.de/congress/2011/Fahrplan/events/4735.en.html.

85. Tobias Engel. 2001. SMS & all its features. In *18C3*. Berlin, CCC. http://events.ccc.de/congress/2001/fahrplan/event/323.en.html.

86. Achim Friedland and Daniel Kirstenpfad. 2005. 3G Investigations. In *22C3*. Berlin, CCC. http://events.ccc.de/congress/2005/fahrplan/events/567.en.html.

87. Andrew Huang and Sean Cross. 2014. Fernvale: An Open Hardware and Software Platform, Based on the (nominally) Closed-Source MT6260 SoC. In *31C3*. Hamburg, CCC. http://events.ccc.de/congress/2014/Fahrplan/events/6156.html.

88. Linux Devices. 2006. Cheap, hackable Linux smartphone due soon. In *LinuxDevices.com*. http://archive.today/PDw5, archived version.

89. Sylvain Munaut. 2012. Further hacks on the Calypso platform. In *29C3*. Hamburg, CCC. http://events.ccc.de/congress/2012/Fahrplan/events/5226.en.html.

90. Dieter Spaar and Harald Welte. 2008. Running your own GSM network. In *25C3*. Berlin, CCC. http://events.ccc.de/congress/2008/Fahrplan/events/3007.en.html.

91. Gerd Kramarz von Kohout. 2001. IMSI-Catcher: GSM Unsicherheit in der Praxis. In *18C3*. Berlin, CCC. http://events.ccc.de/congress/2001/fahrplan/event/340.en.html.

92. Harald Welte. 2005. Towards the first Free Software GSM Phone. In *22C3*. Berlin, CCC. http://events.ccc.de/congress/2005/fahrplan/events/768.en.html.

93. Harald Welte. 2008. Anatomy of smartphone hardware. In *25C3*. Berlin, CCC. http://events.ccc.de/congress/2008/Fahrplan/events/3008.en.html.

94. Harald Welte. 2011. *Report of OpenBSC GSM field test, August 2009, HAR2009, Vierhouten, The Netherlands*. Technical Report. http://openbsc.osmocom.org/trac/raw-attachment/wiki/FieldTests/HAR2009/har2009-gsm-report.pdf.

95. Harald Welte and Steve Markgraf. 2010. Running your own GSM stack on a phone. In *27C3*. Berlin, CCC. http://events.ccc.de/congress/2010/Fahrplan/events/3952.en.html.

96. Paul Wouters and Leigh Honeywell. 2006. Mobile phone call encryption. In *23C3*. Berlin, CCC. http://events.ccc.de/congress/2006/Fahrplan/events/1495.en.html.