Human-Computer Interaction and Cybersecurity Handbook

Abbas Moallem

Smart cities under attack

Publication details
Ralf C. Staudemeyer, Artemios G. Voyiatzis, George Moldovan, Santiago Reinhard Suppan, Athanasios Lioumpas, Daniel Calvo
Published online on: 24 Oct 2018

Accessed on: 17 Aug 2023

PLEASE SCROLL DOWN FOR DOCUMENT

Full terms and conditions of use: https://www.routledgehandbooks.com/legal-notices/terms

This Document PDF may be used for research, teaching and private study purposes. Any substantial or systematic reproductions, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The publisher shall not be liable for an loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
chapter eleven

Smart cities under attack
Cybercrime and technology response

Ralf C. Staudemeyer, Artemios G. Voyiatzis, George Moldovan, Santiago Reinhard Suppan, Athanasios Lioumpas, and Daniel Calvo

Contents

11.1 Introduction ........................................................................................................................ 210
  11.1.1 Cybercrime and its cost............................................................................................... 210
  11.1.2 Smart city evolution................................................................................................. 210
  11.1.3 Smart cities and cybercrime ................................................................................. 211
11.2 Smart city threat landscape ..............................................................................................211
  11.2.1 Use cases for IoT systems ......................................................................................211
  11.2.2 Infrastructure risks ............................................................................................... 213
  11.2.3 Real-world incidents ..............................................................................................213
11.3 Security and privacy requirements for smart cities ..................................................... 215
  11.3.1 Requirements for confidentiality, integrity, and availability .......................... 215
  11.3.2 Authentication, authorization, and accountability requirements .................. 217
  11.3.3 Credential management and end-to-end design .............................................. 218
  11.3.4 Privacy for data-in-transit and data-at-rest ........................................................ 218
  11.3.5 Citizen considerations and concerns .................................................................. 220
11.4 Recommendations ............................................................................................................. 222
  11.4.1 Network security in constrained environments ............................................... 222
  11.4.2 Security architecture recommendations ............................................................ 223
  11.4.3 Secure over-the-air programming ...................................................................... 224
  11.4.4 Integration of privacy ............................................................................................ 225
    11.4.4.1 Privacy trainings for system developers ............................................. 226
    11.4.4.2 Purpose definition and data minimization ........................................ 226
    11.4.4.3 Evaluation of privacy threats and risks ............................................... 227
    11.4.4.4 Privacy by design .................................................................................... 227
    11.4.4.5 Implementation and verification .......................................................... 228
    11.4.4.6 Release of system and education of stakeholders .............................. 229
    11.4.4.7 Technology and organizational response ........................................... 229
11.5 Acknowledgments .............................................................................................................. 230
11.6 References ..................................................................................................................................... 230
11.1 Introduction

11.1.1 Cybercrime and its cost

Cybercrime is crime committed using computing and communication systems such as computers and networks. When cybercrime occurs, computer systems are either the direct target of the crime or the technological means that enabled it. Many reports have been commissioned to estimate the size of the cybersecurity market and the cost of cybercrime. Hemanshu “Hemu” Nigam, an expert in online safety and privacy, estimated* that the market will reach 170 billion USD by 2020.

Allianz Global Corporate & Specialty suggested† that cybercrime costs the global economy 445 billion USD every year. According to Juniper Research, cybercrime costs are projected‡ to reach 2.1 trillion USD by 2019, while Cybersecurity Ventures predicts§ the annual costs to grow from 3 trillion USD in 2015 to 6 trillion USD by 2021. The 2016 Norton Cyber Security Insights Report found¶ that more than 689 million people in 21 countries experienced cybercrime in 2015. In other words, one out of five Internet citizens** experienced some form of cybercrime already that year.

11.1.2 Smart city evolution

Typically, the desktop computer was the device exposing common users to potential cybercrime activities. With the advent of pervasive computing, however, users do not need to be explicitly or directly interacting with computing devices, as before. Many built-in and mobile sensors, smart appliances, and existing public monitoring infrastructure can track, report, and trigger events according to the needs of users in specific locations (e.g., at home, in transit, those using public transportation services, or utilities), and these are only the current incarnations which are supposed to further develop into smart cities, an even more networked infrastructure providing even broader attack vectors.

What makes a city “smart”? While there are numerous definitions, here we adopt the one provided by Cesar Cerrudo [1]: “A city that uses technology to automate and improve city services, making citizens’ lives better.”

A systematic literature review on smart city research for the period of 2008–2016 is provided in by Raaijen and Daneva [2]. As cities evolve, they are assumed to use more and more technology to improve the quality of life for their inhabitants. To quote†† Eberhard van der Laan, Mayor of Amsterdam:

Smart Cities are a bit like football: Every city has a team working on a “Smart-city” and wants to be the Smartest City in the world, and at the start of every season every supporter thinks his or her team will be the global champion. Various ranking systems exist, comparing cities on indicators ranging from energy consumption per capita to life expectancy; from WiFi coverage to crime-rates. In other words: Smart

---

Cities are about everything and therefore about nothing. To be frank: I do not really believe all this Smart-city marketing. I do believe that innovation and technology gives us the opportunity to improve the quality of life of the citizens and make our cities more competitive.

Frost & Sullivan Research estimates* a combined market potential of 1.5 trillion USD globally for the smart city market in segments of energy, transportation, healthcare, building, infrastructure, and governance. Cities around the world are investing big budgets to become “smarter,” such as a 7.4 billion USD smart city project recently announced in South Africa. Other cities, such as New York; San Francisco; Los Angeles; Washington, DC; Seattle; and Miami in the United States, are already there. Similar trends are observed in Europe: London, Barcelona (ranked as the world’s smartest city), Amsterdam, Paris, Stockholm, and Berlin; in the Asia-Pacific: Singapore, Seoul, Tokyo, Sydney, Melbourne, and Hong Kong; in the Middle East: Abu Dhabi, Saudi Arabia (70 billion USD investment), Dubai, and Qatar; and in South America: Rio de Janeiro and Santiago [1].

11.1.3 Smart cities and cybercrime

Smart cities rapidly deploy infrastructure for information and communication technologies (ICTs) and digitize the available information, creating models reflecting specific interactions and domains (e.g., transportation models, relevant rules, and expectations). The “Internet of Things” (IoT) is a prominent key technology that enables connections and communications with a vast number of smart city “things.” These things are used to collect, process, share, and distribute critical information for the sustainable and livable operation of a city. This is to allow the whole city to be managed more efficiently and effectively. At the same time, the ICT-enhanced infrastructure and all of these smart things introduce a huge attack surface for potential cyber-attacks. What does it take for a smart city to be safe and secure for its inhabitants and respect privacy?

This chapter provides a primer on cybercrime in smart cities and possible technological countermeasures to it. Section 11.2 introduces the threat landscape, both for citizen-facing systems and smart infrastructures, and a survey of real-world incidents of attacks against smart cities. Section 11.3 discusses the security and privacy requirements for IoT-based smart city components and the integration of citizen concerns and considerations in the design. Finally, Section 11.4 describes a list of recommendations for designing and deploying citizen-centric IoT-based systems for future smart cities.

11.2 Smart city threat landscape

11.2.1 Use cases for IoT systems

Smart cities employ ICT mechanisms to manage and improve on livability and cost concerns such as pollution, traffic congestion, safety, and the functioning of strained utilities. IoT-based systems are a key technology to improve the efficiency of handling available resources, improving the services provided to its citizens, and improving the living standards of their residents. The role of IoT-based systems is not only to monitor various changes and to report on them, but also to react to certain conditions and adjust accordingly. This enables a more efficient management and utilization of the resources within the cities.

* http://www.frost.com/sublib/display-report.do?id=M920-01-00-00-00
A smart city can monitor, for example, pollution levels and traffic changes. It can also detect certain threshold conditions and prioritize traffic to specific sections to avoid traffic jams, excessive air and noise pollution, or vibration development in certain infrastructure points. It can also equip buildings with structural and metering sensors measuring physical building dynamics, humidity, and energy consumption. Such capabilities will enable proactive maintenance and minimize energy costs for heating and/or cooling. Some further IoT-based examples of a smart city are the following:

- **Smart structural monitoring**: Sensors mounted to specific structures prone to deterioration, such as old buildings or buildings exposed to vibrations due to nearby construction, heavy traffic, or seismic movement, can report and alert on possible structural changes such as mechanical stress [3,4].
- **Smart waste management**: Intelligent waste bins and containers are able to detect their load level so that waste collection services can plan and optimize the routes used in order to collect the waste, minimizing unnecessary trips [5,6].
- **Smart water management**: Built-in sensors within water distribution systems (WDS) can detect and report minor leaks, adjust water pressure to protect the pipes, and prioritize distribution emergencies (e.g., draught). Fine-grained historical information for consumption can also result in significant cost savings for infrastructure operation [7,8].
- **Smart environmental monitoring**: Air quality and noise sensors can provide near-real-time measurements detailing the environmental conditions within the city, automatically adjust traffic restrictions, and suggest healthier routes for pedestrians, people performing physical activities, and citizens with chronic illnesses (e.g., asthma) [9,10].
- **Smart traffic management**: Traffic congestion can be detected by using road- and camera-based sensors. Traffic lights, specific road restrictions, and traffic police can all adjust and react in a timely and semiautonomous manner [11–13].
- **Smart energy metering**: Both service providers and consumers (citizens and the city itself) can adjust and optimize their energy behavior, toward reducing their energy costs and increasing their ecological profile [14–17].
- **Smart lighting**: Combined sensor measurements (e.g., weather, acoustics, and movement detection) can be utilized to adjust the light intensity in public spaces and roads based on weather conditions, physical presence, and historical data (e.g., no pedestrians or cars present) [18,19].
- **Smart parking**: Different kinds of sensors (e.g., cameras, radio-frequency identification, and magnetic) can collect parking information, especially for open spaces (e.g., street sideways) and provide near-real-time updates for free spots or even estimations for near-future availability [20–22]. This results in less idling, less drive time, less emissions, and improved quality of life for citizens [23]. Such systems are expected to become more prevalent with the advent of electric cars and the need to charge them for long hours.
- **Smart public transport**: Citizens schedule their trips better when up-to-date information is available regarding trip duration and recommendations for alternative means and routes in case of service disruption or personalized needs (e.g., carrying a bicycle or opting for more environment-friendly transport means) [24,25].

The list of services provided by a smart city to its citizens and visitors provides a set of use cases for building threat models [26]. A common theme underlying all these services is the blend of existing, physical entities of the private and public spaces with digital technology. There is no longer a clear boundary between the “physical” and the “digital” part. Rather, these systems evolve into “cyberphysical systems” (CPS) that are deeply integrated...
and interconnected. As such, the threat landscape becomes complex. We can no longer rely on the existence of “ground truth,” as it was the case of solely virtual worlds and services (e.g., e-commerce platforms). A possible consequence is the emergence of malicious interventions in the physical realm itself. Consequently, novel protection mechanisms must be developed that protect the physical space of a smart city from digital threats.

11.2.2 Infrastructure risks

The US Department of Homeland Security identifies three crosscutting themes for security considerations of integrating CPS into existing city infrastructures [27]. The first relates to the changing seams. The physical and virtual seams are becoming increasingly permeable as cybersystems and physical systems become networked and remotely accessible, changing and stretching the borders that smart cities must secure. The second relates to inconsistent adoption. Different cities or city parts adopt and migrate to new technologies at different paces. Such inconsistencies pose challenges to developing consistent security policies for all of them.

The third relates to the increased automation. We are experiencing an era where cyberphysical infrastructure migrates control from human beings to algorithm-based systems. This introduces a level of security and resilience into a system by mitigating potential human errors. However, at the same time, algorithmic responses can be hard to predict and comprehend and can result in cascading failures that were not considered. The trillion-dollar stock market “Flash Crash” on May 6, 2010, in the United States, serves as a very good example of how fully automated, high-frequency decision systems can cause significant damage beyond our control [28,29].

11.2.3 Real-world incidents

The so-called Northeast blackout of 2003 was one of the first software-related incidents that affected the operation of whole cities and a total population of 55 million people [30]. A software bug in the alarm system at a control room of the FirstEnergy Corporation, located in Ohio, United States, was the primary cause. A sequence of events in a different order from that expected by the system was produced by exploiting a race condition in the control software, and this resulted in a series of cascading effects way beyond a local energy blackout. Infrastructures such as power generation, water supply, transportation, communication, and factories were severely affected in the Northeastern and Midwestern United States and the Canadian province of Ontario. During another event in 2012, a software bug summoned some 1200 people to jury duty on the same morning, resulting in a tie-up on Interstate 80, California, United States, as people drove to court.*

“Major computer problems” caused the San Francisco Bay Area Transit system to shut down, affecting hundreds of thousands of daily riders while some 500–1000 passengers were trapped on trains in the late evening and early morning hours†. A software update in December 2016 caused a four-hour service outage affecting travelers entering the United States in the middle of the holiday week‡. A 14-year-old teenager turned the train system of Lodz, Poland, into a personal train set, “triggering chaos and derailing four vehicles in the process,” as covered by Schneier on Security§.

---

* http://www.npr.org/2012/05/03/151919620/computer-glitch-summons-too-many-jurors
§ https://www.schneier.com/blog/archives/2008/01/hacking_the_pol.html
television remote control unit to succeed in this, demonstrating how easy accessing the transport infrastructures of some city might be. Although the aforementioned events were merely proofs of concept and not caused by an intentionally malicious action, they still serve as a good reminder on the complexity of all these interconnected smart city infrastructures and how vulnerable systems can be. Turning to malicious incidents, it was officially acknowledged that Iranian hackers remotely breached a water dam outside of New York, United States, in 2013. The attack did not cause any damage but still demonstrates that attackers can control physical infrastructure even when not physically close to their target.

In coordination with the involved authorities, researchers demonstrated that smart traffic control systems often used in intersections in the United States are vulnerable [31]. Three major weaknesses contribute to this: (1) lack of communication encryption at the network level, (2) lack of secure authentication at the system level (default usernames and passwords), and (3) controller software unpatched for known exploits. The attack scenarios showed that as many as 100,000 intersections in the United States and Canada could potentially be taken over maliciously. The researchers recommended defenses at the organizational and technical levels. The state of adoption is unreported to date [31].

On December 23, 2015, Ukrainian power companies experienced unscheduled power outages impacting 230,000 residents in Ukraine.† This well-prepared attack was the first confirmed hack to take down a power grid. The availability of rich system and firewall logs allowed the investigators to reconstruct the attack timeline and link it with the BlackEnergy3 malware [32]. A few months later, a nuclear power plant in Germany was infected with malware that could give remote access to attackers.‖ The threat was considered low, as the systems were isolated from the Internet. Still, some 18 removable data drives were detected to contain malware, including portable universal serial bus (USB) storage drives (e.g., thumbdrives and sticks). Such devices are a known attack conduit; there are already attack examples that use USB drives to bypass even air-gapped systems [33,34].

On April 7, 2017, all 156 Dallas storm warning systems started blaring across the city at around midnight. It took more than 90 minutes to silence them by shutting down their radio system.§ While initially considered a malfunction, it was acknowledged later that a malicious actor had penetrated the radio system and initiated the alarm. Soon after, on May 12, 2017, the WannaCry ransomware [35] attack was launched worldwide, affecting more than 230,000 computers in 150 countries.¶ The attack exploited a Microsoft Windows operating system vulnerability, encrypted data, and demanded ransom payments in the form of Bitcoins [36]. The effects of WannaCry were felt globally**:

- In the United Kingdom, hospitals and doctors were unable to access patient data and medical appointments, and operations had to be cancelled.
- In Germany, electronic boards displaying train route information were disrupted.
- In France, the carmaker Renault was forced to stop production at a number of sites.
- In the United States, FedEx, a delivery company, was affected.
- In South Korea and Indonesia, hospitals suffered.
- In South America, the Brazilian telecom operator Viva and the LATAM Airlines Group reported effects.

---

† https://ics-cert.us-cert.gov/alerts/IR-ALERT-H-16-056-01
‡ http://www.telegraph.co.uk/news/2016/04/27/cyber-attackers-hack-german-nuclear-plant
§ http://www.reuters.com/article/us-texas-sirens-idUSKBN17B001
**https://www.wired.com/2017/05/ransomware-meltdown-experts-warned
Chapter eleven: Smart cities under attack

The wide range of targets affected by this attack demonstrates the strong dependence on ICT of modern and evolving smart cities. Even more worrying, a postincident scan two months later revealed that more than 60,000 hosts that spread in 130 countries were still vulnerable to this attack.*

A city-wide attack can happen without penetrating air-gapped systems or infecting servers and large computers. Constrained devices that form the IoT and the smart city of the future, such as vulnerable smart light bulbs, could also be used to spread malware across a whole smart city. It would take only 15,000 randomly located light bulbs to control an area as big as Paris, France [37].

Finally, in this set of examples, The Devil’s Ivy† (officially: CVE-2017-9765) vulnerability was discovered in July 2017. Devil’s Ivy again vividly reminded us that many IoT systems share the same software codebase for implementing communication and network stacks. This makes them equally vulnerable, despite their apparent diversity in function and shape.

11.3 Security and privacy requirements for smart cities

The impact of vulnerabilities on smart city operations can be significant. The numerical and geographical scale of computing makes postincident reaction and fixing time-consuming and costly or sometimes—infeasible. Hence, it is necessary, to integrate appropriate security and privacy mechanisms of a smart system, preferably beginning in the design and predeployment phases [38–40]. In this context, IoT-based systems are crucial to defend, as they are most likely to represent the first and most accessible link in the smart city security and privacy chain: IoT sensors are scattered through a vast area, left unsupervised, easily accessible, and usually without complex software and hardware mechanisms meant to limit their exposure to tampering. Although the adoption of smart technologies is done in a fast pace, we consider that there is the need for integrating proper security and privacy. In the following, we review the security and privacy requirements for smart cities.

Gaining access to an IoT computing system can take place by either active or passive attacks. Active measures might involve manipulating devices physically (e.g., by analyzing specific components and reverse engineering them) or simply infiltrating them by exploiting software vulnerabilities, the end goal being to gain logical control over the system. In addition, active attacks can also be formed on communication channels by the insertion of messages or by the downgrading of the parameters of the communication channel (e.g., bandwidth). In contrast, the eavesdropping of communications is an example for passive attacks [39–41].

It is of great importance to acknowledge that information security is not limited to the protection of the content that is stored and transmitted. Other sensitive (meta-) information are revealed when simply observing information flows and patterns. The protection of meta-information is essential in smart city scenarios for the security and privacy of citizens [39,42].

11.3.1 Requirements for confidentiality, integrity, and availability

Confidentiality is the property that protects information from being disclosed to unauthorized parties. It denies access to those not entitled—no matter whether the data are stored or transmitted. On a communication channel, an attacker is assumed to be able to eavesdrop on messages that are being exchanged. For data-at-rest, the attacker is assumed to have physical access to the device.

Citizens may assume that data flowing between devices at their private home or in their immediate vicinity are confidential, that it is inaccessible to any other parties without consent or warrant, and that at least not everyone is able to collect and process this kind of data at will. In a smart city environment, this rather clear separation between private and public environments blurs.

To give an example, the city of Amsterdam, in the Netherlands, supports more than 40 smart city projects ranging from smart parking to the development of home energy storage for integration into the smart grid.* One of these projects concerns the installation of smart energy meters with incentives provided to households who plan to actively save energy. Smart meters record energy consumption in households and report these in short intervals (e.g., 2 s) to the provider. The benefit for the energy provider is that frequent reports allow demand management. Consumers could then benefit from possible lower rates during off-peak times. The downside is that these frequent measurements reveal detailed information on household activities, including presence, electrical devices in use, and even what content consumers watch on television.

Integrity is violated whenever data or a system is modified in an unauthorized way. Modification can occur due to transmission errors or due to an active attack. To ensure a correct and expected behavior of an information technology (IT) system, it is necessary that any modified data must become reliably distinguishable from unchanged data.

The protection mechanisms against malicious modifications differ from those that detect random transmission errors. Integrity protection is basic security functionality. Integrity protection mechanisms can be used to authenticate software and support, securing the distribution channels. Integrity protection detects erroneously or maliciously modified information and should support using it as input for further analysis. Thus, integrity violations can be used as early detectors for a fail-safe behavior.

For example, in smart cities, sensed data are gathered and used by algorithms to enable decision-making. Thus, the decisions are based on data gathered, and bad quality input data can lead to bad decisions. Imagine a smart city without integrity-protected messages: every sensor value can be potentially tampered with. A faulty or misbehaving air pollution sensor in one city area might cause that area to be declared a “zero emission zone” (ZEZ).

To visualize the consequences, assume that a high risk of pollution is detected in a smart city. As a response, the access to the inner city area is restricted. The inner city is declared a ZEZ, and only electric cars are allowed. This leads to the cars in the neighborhood being denied access to the inner city area. Citizens in the area are expected to experience far less noise due to the decreased traffic. The surrounding areas can however experience congested roads due to the increased traffic—caused by noncompliant cars having to avoid the ZEZ.

Likewise, all control messages could be manipulated, e.g., by changing an “access denied” message from the barrier control system into an “open-barrier” one.

A defective sensor could also send erroneous readings. In such a case cryptographic mechanisms would still recognize the data as being correct—which, looking purely at the security perspective, would be true. Hence, smart cities need to deploy additional processing logic to detect erroneous sensor readings and possibly send maintenance crews to investigate the actual sensor.

Availability ensures timely and reliable access to devices and services. The availability property is violated if an attack succeeds by degrading a computer resource or rendering it unavailable, i.e., a denial-of-service attack.

* http://amsterdamsmartcity.com/projects
When monitoring critical environments, sensor values and alert messages need to arrive in a timely manner; otherwise, detection fails and no alarm is triggered (recall the Northeast blackout of 2003 incident in Section 11.2). Critical values might be related to industrial contamination with potentially hazardous consequences; for example, air pollution, high radiation levels, or a decrease of water quality. For example, the city of Tarragona, in Spain, is next to chemical factories. Here, constant and reliable monitoring of air pollution for critical substances is vital to protect citizens. It is essential that potential air contamination can be detected before it reaches the closest households. To achieve this, the deployed detection sensor systems need to send their data to a monitoring server. The city can then detect and potentially react timely in any manner whatsoever. Should parts of the infrastructure, such as the sensors, the communication networks, or the monitoring servers, become unavailable, the detection will fail and the population may not receive the timely warning the system was designed to provide.

11.3.2 Authentication, authorization, and accountability requirements

Successful authentication and authorization enables accountability to be achieved for a certain action by a certain entity. The three goals are often grouped together and referred to as “AAA” or “triple A”: authentication, authorization, and accountability as defined by Shirey [43].

Authentication is the “process of verifying a claim that a system entity or system resource has a certain attribute value.” Authorization is the “process for granting approval to a system entity to access a system resource.” Authorization controls who can do what to which objects, while authentication involves identifying who is seeking the access, often being a specific part of the authorization process. Accountability enables “the detection of actions to be traced to the potentially responsible entity.” To achieve accountability, an entity first needs to be authenticated, and then the request for access is subject to authorization. To control access, systems must check if an entity is authorized to carry out a certain action.

There are authentication challenges at different layers of a system as complex as a smart city. Returning to a previous example, assume that today, only electric cars are allowed because the inner city is declared a ZEZ. In this setting, the first question is, “who is the entity of the system that you want to authenticate?,” which can be hard to answer in technical detail, as discussed by Gollmann [44]. Do we want to authenticate a single car, the on-board device of the car, or the car passengers? Here it becomes obvious that peer entity authentication can happen on various layers.

As reported in 2004 by the World Health Organization, the city of Milan, in Italy, is one of Europe’s most air-polluted urban centers. The problem is caused by very high downtown traffic volumes [45]. In 2008, the city introduced electronic road pricing to address traffic congestion, to promote sustainable mobility and public transport, and to decrease the smog levels. The so-called inner city Area C is a restricted zone.* The toll revenues from cars entering the area are reinvested into sustainable energy projects. The area is accessible via gates monitored by video cameras equipped with automatic number plate recognition technology.

In this use case, the attribute value is the car license plate number. If the road toll system recognizes the license plate of a car as registered, the system can charge the owner’s account for its use of the toll road. Admittedly, one weakness of the system design is that it is not accounting for privacy requirements. The plate number data are stored and correlated, potentially enabling unauthorized citizen surveillance.†

---

* https://www.comune.milano.it/wps/portal/ist/en/area_c
† The interested reader can consult an example case reported in New York in 2013: https://www.forbes.com/sites/kashmirhill/2013/09/12/e-zpasses-get-read-all-over-new-york-not-just-at-toll-booths
There is no need for an application to realize authentication based on such unique identifier such as license plate numbers. For example, in the aforementioned case, it suffices to authenticate and distinguish between electric (allow) and fossil-fueled (deny) cars. This means that the relevant attribute value is “I am an electric car.” The claim needs to be proven, so that the entrance control system can ensure that it only grants access to an electric cars. The disclosure of the related attribute for the aim of inner city access control should be realized in a privacy-preserving means, using, for example, anonymous credentials [46].

11.3.3 Credential management and end-to-end design

Each entity in an IoT system needs to have credentials required to prove its own claims to the authorization components. This means that all participating systems need some form of keys and mutual trust associated with them. For example, would the smart city trust the car manufacturer to vouch for the “I am an electric car” claim?* Or would the car need to be regularly inspected and be issued with a token issued by a state-trusted institution? No matter how it is implemented, there is a need for secure key management and secure key distribution. If they are not in place, attackers might disguise their identities and credentials for their benefit. For example, an attacker can present their hybrid car on demand to appear as an electric car and freely cruise the aforementioned Area C of Milan.

Depending on the involved systems and communication links, transmitted data in the communication system can be protected by different means. One option is to protect the transport link, the other is to protect every message separately. Both options have serious disadvantages, suggesting that a layered approach is required to thwart intrusions. To make the differences obvious, let us consider confidentiality protection by encryption. Hop-to-hop, or so-called link-level, protection, encrypts data between neighboring network nodes. In end-to-end security, the confidentiality and the integrity protection is between the endpoints of the communication. As such, authentication (and finally the authorization) can be performed between the endpoints as well [39].

For example, to logically authenticate a specific car, one needs to be sure that what you technically authenticate is affixed to that specific car, so that it cannot be easily removed and placed into another car. Achieving end-to-end protection means that the need to trust the intermediate systems is removed. While this is preferable, it cannot always be achieved due to layered approaches and independent subsystems.

11.3.4 Privacy for data-in-transit and data-at-rest

The citywide communication networks of smart cities are very hard to physically secure against unauthorized access. In the IoT domain, the local network access is predominantly wireless and is therefore prone to eavesdropping. To protect citizens from any kind of hidden loss of personal information when accessing public resources, the communication also needs to be protected against traffic analysis [39,41,42].

Consider, for example, the case of encrypted voice-over-Internet protocol communication through a piece of software, e.g., Microsoft Skype. It sounds like eavesdropping on such communication would not be possible without the knowledge of proprietary (secret)

communications protocols, access to cryptography keys, and capabilities to decrypt the conversations. Despite the strong encryption, isolated phonemes can be classified and given sentences identified from vectors of packet sizes of Skype traffic. The reported accuracy can reach more than 80% under specific conditions [47].

Smart energy meters provide automatic meter readings in intervals defined by the electricity provider. The meter readings should remain confidential; highly frequent measurements can reveal detailed information on household activities (e.g., pattern of use for specific electric appliances) [48]. Network traffic containing meter readings can be mapped to communication partners and traced to a specific meter and household. Even if traffic is encrypted, natural changes in traffic patterns and volumes can reveal precious information about the household operations, rendering encryption obsolete. An overview of privacy-preserving data aggregation techniques is provided by Erkin et al. [49]. The need to further anonymize meter readings that are stored by the energy providers, once they are received by them, is discussed by Efthymiou and Kalogridis [50].

There is no reproducible public interest to leave citizens traceable and facilitate continuous surveillance. Nevertheless do many smart city applications heavily depend on citizen location information to provide personalized services (e.g., location-based services). The traditional communication security goals are unable to protect location information. This kind of information leaks from metadata and can be extracted by traffic analysis with little effort. The protection of metadata requires a different approach provided by so-called privacy-enhancing technologies (PETs). However, potentially suitable protection mechanisms do still suffer from a huge overhead in terms of resources.

All citizens of a smart city together form a giant sensor, continuously contributing datapoints from the public space. This includes e-tickets for public transport, automated payment of tolls for car commuters, traffic navigation through smart city, and even presence in public spaces while carrying digital devices. Omnipresent smart cameras can also record entrance into public spaces. Even innocent-looking waste bins can assist in serving targeted advertisements to passing citizens, as demonstrated in London, United Kingdom.*

It is not an exaggeration that citizens are constantly producing datapoints in a smart city. While this can significantly improve their quality of life, such data are long lasting. They are continuously collected, processed, and stored for long times or even indefinitely. These data-at-rest are the new gold for numerous stakeholders. Privacy-preserving techniques, algorithms, and mechanisms are more than necessary to defend against misuse of stored information.

As an example, in 2016, a healthcare organization in Johannesburg, in South Africa, went paperless†. This included systems for deploying digital media health records to improve record keeping and as well patient care‡. There is a high risk that sensitive medical information will leak at some point and be processed in unintended ways, most probably not for the benefit of citizens. On the one hand, it may be very useful for benevolent local governments to hold information about the medical conditions of its citizens and dedicate appropriate budgets to benefit all. On the other hand, governments are not all benevolent, and individuals can be targeted by dishonest (state) employees or external contractors. For a beginning, strict access control and logging, privacy-preserving data processing, and data minimization are essential first steps. Still, if the loss of highly confidential data happens in the most secure environments, such as the National Security

---

† http://www.htxt.co.za/2016/06/08/joburg-clinics-say-goodbye-to-paper-at-digital-ehealth-system-launch
‡ http://ehealthnews.co.za/joburg-invests-in-ehealth-to-benefit-patients
Agency of the United States, it is hard to imagine that any (smart) city will be in better position to defend its own citizens. In the context of highly confidential health data and the economic interests of medical insurance companies, the interested reader is referred to Xu and Cremers [51].

11.3.5 Citizen considerations and concerns

Security experts raised concerns early on regarding the security and privacy of Internet-connected smart-home devices [52–54]. They put significant effort into analyzing existing systems, including privacy leakage through pairing and discovery protocols [55], insecure communication protocols [56], and vulnerabilities that allow remote spying on residents in smart homes [57–59]. Even simple IoT devices, such as smart locks and light bulbs, have been shown to be vulnerable [37,60,61]. The trustworthiness of an IoT application is impacted by the implemented privacy and security practices [62].

Although an independent concept, the smart home is the most widespread example of how humans interact with the smart city environment as technology users. Consequently, privacy and security concerns of citizens and consumers become more evident and can drive the adoption of new technologies [58,63–67]. Security and privacy concerns evolve over time and with the audience, from the preprocurement phase of smart-home IoT products and services to the deep integration of those products and services in daily life.

The European Union (EU) FP7 project “RERUM”* conducted a survey to explore the views of future consumers of potential smart city IoT products and services coming out of the project. As depicted in Figure 11.1, the respondents were most interested in smart-home services in domains of home automation and remote control, while less priority was given for smart transportation and e-health applications. The least priority was on smart-home security and surveillance services. Respondents were concerned for their privacy, citing photos and video streams that are transmitted over public Internet to third-party operators for further processing. They fully trust neither the technical means used to secure the communications nor the human operators whom they fear are constantly monitoring their private space with malicious intentions. This finding is in accordance with Lee and Kobsa [68,69], which also report that privacy considerations affect people’s concerns about IoT services such as device tracking. Furthermore, Lee and Kobsa [68,69] highlight that the entity that monitors and collects personal information plays a crucial role in their decision whether they use an IoT service. Regarding the RERUM project findings, as depicted in Figure 11.2, respondents ranked the service price highest among their concerns when considering purchasing a smart-home service, along with the security and privacy, which are considerably more important factors compared to other criteria, such as the diversity of offered applications.

Differences between people living in the same smart home (e.g., technology enthusiasts and passive users—or teens and parents) may result in tensions about technology use and privacy considerations [70,71]. Privacy and security concerns are raised by consumers when they start to use smart-home devices and are in the novelty phase [72,73]. Even when people have lived in a smart home for many months and have had daily interaction with smart-home devices, it may be that their threat models is generally naive, aligned with the sophistication of their technical mental models. Consequently, in this knowledge state, their technology choices are likely to be based

* https://ict-rerum.eu
on requirements that may conflict with security and privacy (e.g., cost and interoperability) [74]. The comfort level of individuals with their IoT devices varies in different IoT data collection scenarios. The comfort level with IoT devices and device security is profoundly influenced by their perception on how their data are being used to be beneficial [64,75].
11.4 Recommendations

At first glance, IoT security challenges in the smart city domain (as for any IT system) could be considered already solved by applying good security practices, such as encrypting data end to end by default when communicating over the network. However, they become challenging to solve in large-scale distributed systems. IoT services may seem simple from the user point of view since the user experiences only the upper layer, i.e., how the information under consideration are displayed and utilized. However, the real picture is totally different. If we ask where we should expend efforts to enhance security in the IoT and, specifically, in the smart cities, the answer is everywhere! Because a system fails first at its weakest link [39].

11.4.1 Network security in constrained environments

As it was presented in the introduction of this chapter, IoT infrastructures that are typically deployed in a smart city integrate smart devices or smart things as the basic and fundamental components of the system. These devices provide sensing and actuation mechanisms that are the connecting links between the cyberworld and the physical world. They enable new services aiming to improve the lives of citizens.

Within the scope of smart city environments, a great diversity of use cases have been designed and implemented during the last decade. Each one of them targets different actors and users, establishes different objectives and requirements, and culminates in the design of different architectures and the integration of different technologies.

Despite this heterogeneity, the identification of a set of common requirements is possible [76]:

- Deployment of a huge number of devices to allow provisioning services to the majority of the citizens. For instance, more than 15,000 sensors were installed as part of the EU FP7 project “SmartSantander” to develop a first catalog of small applications in a relatively small city considering population and urban area.*
- Most devices must communicate data wireless due to their location and reduce installation complexity and costs. Some devices may be attached to mobile elements (e.g., vehicles).
- Distances between the different devices may be long, from hundreds of meters to several kilometers.
- Devices typically exchange small amounts of data over long periods.
- In many cases, devices are powered using batteries and may use energy harvesting. Thus, an efficient use of available resources is an essential nonfunctional requirement to optimize power consumption and extend the working life without supervision.
- Typical constraints also impose additional restrictions to the size and cost of each device.

Hybrid ICT architectures derived from these common requirements often lead to the deployment of low-power and lossy networks, which interconnect constrained devices with a variety of links, such as IEEE 802.15.4 or low-power Wi-Fi to backend infrastructures [35]. These devices are characterized by severe constraints on the central processing unit, memory, storage capacity, and communication bandwidth [77]. An example case is

* http://www.smartsantander.eu
the Zolertia Re-Mote platform*, which was developed in the aforementioned RERUM project. It is based on an ARM Cortex-M3 processor running at 32 MHz; it includes 512 KB of flash memory and 32 KB of random-access memory (aka RAM).

### 11.4.2 Security architecture recommendations

It is essential for information transfer and exchange to use constrained IoT systems. Smart cities must recognize this, since such communications of the IoT devices can control the behavior of city-critical systems such as water dams, traffic lights, or the power distribution network. At the same time, it is also essential, at a minimum, to ensure system resilience against most common attacks (e.g., packet replay attacks and denial-of-service attacks). Realizing security mechanisms for smart city IoT systems must also consider the impact on energy consumption. Security solutions that exist in enterprise environments (e.g., strong cryptography, control-flow integrity, and deep packet inspection for content filtering and intrusion detection) cannot be readily applied in citywide distributed low-powered devices that are in most cases physically hard to approach due to their special characteristics and constraints, e.g., strongly limited hardware resources, lack of advanced user interfaces, massive amount of them covering large areas, or even installed in moving or difficult-to-access locations.

All these goals combined result in very challenging requirements: mechanisms or algorithms that enforce privacy and security tend to be computationally intensive and consequently cost energy. It can be expected that processing times and energy consumption overheads for privacy and security are actually higher than the resources needed to process and transmit the data. Moreover, the integration of features to enforce security and privacy also increases the code size of the firmware, which also could be an important problem considering scarce memory resources.

Therefore, when designing the IoT architecture and to ensure an efficient and flexible implementation of authorization mechanisms, it is recommended that a distributed approach as proposed by the ACE (Authentication and Authorization for Constrained Environments) Working Group of the Internet Engineering Task Force be considered. In this approach, constrained nodes delegate authorization mechanisms to more powerful entities that are part of the overall IoT platform [78]. Constrained devices are known as resource servers, which host one or several resources, for instance, a sensor that measures the external temperature or humidity. The client represents an actor that tries to get access to the information or resources that are exposed by the resource server. We could think about a citizen that wants to retrieve the last measurements from his/her laptop or smartphone or even about another constrained device. Both actors, client and resource server, rely on more powerful entities for the authorization process: the authorization server and the client authorization server, respectively.

The following recommendations are proposed in order to ensure that security and privacy are preserved during the authorization process:

1. Secure protocols must be chosen for communications between clients and authorization server (e.g., implemented using transport layer security over transmission control protocol).
2. If the characteristics of the resource server permit it, secure protocols will be used for communications between clients and resource servers (e.g., end-to-end encryption with datagram transport layer security (DTLS) over user datagram protocol (UDP) [53]).

* http://zolertia.io/product/hardware/re-mote
3. If the characteristics of the resource server do not enable the use of secure communication protocols due to energy consumption constraints or limited computing capabilities, additional recommendations are needed:

a. Tokens generated by authorization servers will be composed of two parts: one destined to the client and another to be attached to each request sent to the resource server. The former one must never be sent through an unsecured channel (e.g., constrained application protocol [53,79,80]).

b. Resource servers shall be able to prevent typical attacks and to validate the integrity of the access token using mechanisms such as proof of possession [81].

c. Confidentiality of exchanged data shall be protected applying lightweight encryption algorithms (e.g., pseudorandom functions based on ChaCha/Poly1305 [82]) or with built-in hardware support if unavoidable.

11.4.3 Secure over-the-air programming

The firmware embedded in IoT devices must be updated to ensure safe and reliable operation. The integration of over-the-air (OTA) programming mechanisms as part of the IoT ecosystem allows bugs to be solved or new functionalities to be added in deployed devices. Moreover, OTA is essential in order to ensure that existing infrastructures will be able to address future security flaws without requiring the substitution of massive amounts of old units or manual flashing procedures. In fact, the application of continuous development and continuous integration techniques that make use of underlying OTA technologies may result in mitigating and preventing some of threats and vulnerabilities previously described [83].

While OTA is a crucial requirement for massive IoT deployments in smart cities, it cannot be denied that it also constitutes a potential vulnerability:

- The size of the firmware grows with the increasing number of functionalities that are implemented in IoT devices. This is a problem since OTA may require using a relevant percentage of the network bandwidth and of the resources of the device to execute a noncore task that must not affect the availability and performance of the device from the client’s perspective. Denial-of-service attacks may also try to exploit unprotected resources that are exposed to receive the updates.

- OTA could be used to gain control of devices by replacing the existing firmware or installing malware applications to sniff data, to monitor the device and network activity, to corrupt or modify the provided data, or just to completely disable victims.

To avoid these threats and safety flaws that may be used to attack OTA vulnerabilities, the following recommendations are encouraged [84]:

1. The IoT platform must implement communication mechanisms and interfaces between the version control component and the gateway that ensure privacy and security and that enforce appropriate authorization policies (i.e., only gateways that are part of the system will be allowed to retrieve updates, and gateways will receive information only about nodes that are connected to them).
2. The version control component of the IoT platform must sign the firmware images to be deployed to devices (e.g., RSA with SHA256 [85]). The gateway will be able to validate the origin and integrity of updates before sending them.
3. Safe communication protocols must be chosen to send updates from gateways to devices, providing privacy and data integrity (e.g., DTLS [53]).
4. Authorization policies must be enforced by the target device, i.e., only gateways will have permissions to send updates. Access tokens and authorization servers may be used to implement this measure.
5. Error detection techniques will be applied to ensure reliable transmission of firmware updates. Cyclic redundancy check codes or cryptographic hash functions may be used depending on the available computing resources of the target devices.
6. Devices must store the new firmware in a separate memory space to preserve the integrity of the currently used version.
7. Mechanisms to recover the previous state must be implemented as a backup solution, if something fails during the startup of the new image.

11.4.4 Integration of privacy

Although requirements and technologies stand on their own, to make an IoT system or generally any ICT system privacy enhanced, one needs to understand how privacy can be integrated in the design and planning of that system.

In security, security development life cycles are a well-understood way of conducting a systematic approach for ensuring security in software engineering. A privacy development life cycle should be defined the same way: to systematically introduce a privacy methodology in system engineering. However, security and privacy development life cycles have significant differences.

To make the definition of a privacy development life cycle tangible, we look at popular security development life cycles for software development and try to derive goals that can be used for privacy: the Microsoft Security Development Lifecycle (SDL) [86] and the Open Web Application Security Project Software Assurance Maturity Model [87]. Beneficial synergies form both approaches can be recognized [88,89], and the following set of guidelines for the steps of both security life cycle frameworks can be defined.

In general, the following can be stated:

1. Train personnel or ensure that personnel is qualified.
2. Identify threats, evaluate risks (acceptable vs. mitigable threats), and elicit requirements.
3. Design the system according to the requirements.
4. Implement the system, fulfilling all requirements.
5. Verify that the system fulfills the requirements.
6. Deploy the system while making sure the requirements will still apply in the deployment environment.
7. Keep the system developers ready to respond to any conflicting or emerging situation.

It should be noted that phases may differ from the SDL counterpart, as system engineers may adapt the content and the focus of each step according to their needs. The reader
is referred to Suppan [88,89] for further details. Figure 11.3 summarizes the life cycle (also note that the life cycle consists of living, continuous processes).

In the following sections, we will consider each step to support the definition of a privacy development life cycle.

11.4.4.1 Privacy trainings for system developers
Precondition: Training developers in privacy topics is as essential as training them in security. Although all team members should understand why privacy protection is fundamental and be familiar with the adequate guiding rules (e.g., the EU Data Protection Rules or the applicable guidelines), we recommend that one person in the team should be responsible for privacy, while everyone on the team should be knowledgeable about it.

The responsible person should be well trained in the technical aspects of designing and implementing privacy friendly systems. We call this person the “privacy expert” of the team. The privacy expert should know a privacy engineering framework such as PRIPARE (Preparing Industry to Privacy-by-design by supporting its Application in Research) [90]. The most critical condition for achieving a privacy-friendly product is the presence of at least one, but preferably several, security and privacy experts in the team.

The expert is also responsible for data protection expertise in the development team and should be a participant in every phase of the life cycle. The expert also brings the knowledge as to where mature PETs and best practices can be found. The life cycle itself does not focus on developing new technologies, which could cost a considerable amount of time, research, and technical expertise, but on using existing building blocks and suitable PETs, such as those we gave a brief overview of in previous sections.

A privacy expert needs to be continuously enhancing its technical skills and state-of-the-art knowledge, in particular for IoT with new developments in PETs. Legal support may be needed to resolve privacy-related incidents. In these situations, a privacy expert should be as well aware when legal support is required.

11.4.4.2 Purpose definition and data minimization
The first phase of the privacy life cycle development is the specification of requirements for the system. Here, functional requirements of the system are analyzed by posing the following questions, which follow one of the most basic principles in privacy, the principle of purpose. To be more precise, what personal data do the system collect,
what is their specific purpose, and can the system reach its desired functionality with less personal data?

The following process is iterated stepwise to obtain answers to these questions and more concrete and operational privacy requirements (or PETs):

- Obtain or define the system data flow and the functional goals and requirements the system.
- Determine which personal data are needed to achieve the functional goals of the system.
- Analyze the functional requirements and determine if existing PETs can help minimize the data that are needed for the system.
- Define limits on data usage and data retention in the system.
- The privacy expert analyzes the proposed solution and suggests new possible technologies to reduce data usage in the system.

The reader is referred to Suppan [88,89] for further details on this process.

11.4.4.3 Evaluation of privacy threats and risks

After the definition of the required personal data in the system, privacy requirements, and privacy goals, this phase is used for privacy threat analysis. Several frameworks for privacy threat analysis have been proposed, such as LINDDUN (Linkability, Identifiability, Non-repudiation, Detectability, information Disclosure, content Unawareness, and policy and consent Non-compliance) [91], PriS [92] and FPFSD (Framework for Privacy-Friendly System Design) [93].

LINDDUN is especially well suited for the integration of a privacy development life cycle as it is based on STRIDE, a popular risk analysis methodology for security life cycle; see Microsoft Corporation [86]. System developers trained in an SDL and STRIDE should be able to learn the LINDDUN method easily, reuse existing system models (particularly data flow diagrams or DFDs) for their systems, and see synergies or problems of both security and privacy goals. LINDDUN follows STRIDE in defining six steps. The first three cover the “problem space,” focusing on the problems, identifying privacy threats and defining requirements of the system. The last three steps cover the “solution space,” which aim at fulfilling the requirements; see Microsoft Corporation [86]:

1. Define DFDs of the system (a graphical representation of the system).
2. Map privacy threats to elements of the flow diagrams.
3. Identify threat scenarios according to privacy threats.
4. Prioritize threats/risk analysis.
5. Elicit mitigation strategies.
6. Select PETs that support the mitigation strategies.

The last step is specially challenging for IoT environments. Existing PETs may need to be adapted for constrained and lossy environments or need to support a vast amount of fluctuating participants (e.g., due to devices that frequently change positions).

11.4.4.4 Privacy by design

The design phase develops strategies for implementation, verification, release, and response. Since this phase is also functional, security and privacy requirements are adjusted to one another. For example, functional requirements might need change to
respect policies, security procedures might need to be adapted to support unlinkability, and privacy requirements might turn out impractical due to core functional requirements and need to be reshaped.

Conflicts might appear between goals, and therefore, best practices can be useful. Best practices are strategies that have been employed by others with good results. For example, Hoepman [94] has defined eight design strategies for privacy, which can be realized using privacy patterns (i.e., best practice solutions), namely, the following:

1. **Minimize** states that the amount of personal data that are processed should be restricted to the minimum amount possible (most basic privacy design strategy).
2. **Hide** states that any personal data, and their interrelationships, should be hidden from plain view.
3. **Separate** states that personal data should be processed in a distributed fashion, in separate compartments whenever possible.
4. **Aggregate** states that personal data should be processed at the highest level of aggregation and with the least possible detail in which it is (still) useful.
5. **Inform** corresponds to the important notion of transparency: Data subjects should be adequately informed whenever personal data are processed.
6. **Control** states that data subjects should be provided agency over the processing of their personal data.
7. **Enforce** a privacy policy, compatible with legal requirements, should be in place and should be enforced.
8. **Demonstrate** requires a data controller to be able to demonstrate compliance with the privacy policy and any applicable legal requirements.

Hoepman provides procedures to address each of the strategy elements listed earlier to support their technical engineering. As some strategies have rarely been used before, Hoepman points out that new patterns are needed. The reader is referred to the privacy patterns database [95].

At this point, we would like to add the following principles to the ones just mentioned:

- **Early application of policies and filtering:** The processing of personal data should take place in devices under the control of the data subject or, if that is not feasible, as close to the point of origin as possible. This strategy takes advantage of increased processing power in personal devices.
- **Process data in a distributed fashion:** Hoepman describes a separation strategy to process data in a distributed fashion, but, e.g., storing data in separated databases is not enough, if it can be relinked across databases. This strategy helps avoid such cases by establishing a mechanism that actively checks for possible identifiers and allows proper separation without the possibility to reaggregate the data.
- **Ensuring the usability in privacy controls:** This has several objectives, such as making privacy controls usable for a variety of users with different skill levels, integrating privacy controls seamlessly into the system, and making the users understand what they are seeing and how they can affect it with the controls provided to them.

11.4.4.5  Implementation and verification

Proper documentation and by-default configuration are key elements in the implementation and verification phase. Users must be able to perform informed decisions in a manner...
that preserves their privacy easily. The system should be “privacy friendly” from the start. This is called “privacy-by-default” and is one of the most important fair information and privacy-by-design principles, as the majority of users will interact with a system in its lifetime with the default settings, as pointed out by Willis [96].

Secure coding procedures will be needed to avoid privacy issues, which could otherwise become visible later. PETs need to be securely implemented in the same way as security mechanisms, e.g., by coding experts, and frequently verified with code reviews. Implementation strategies, as defined in the design phase, help assure that the implementation effort is controllable and on time and reaches the desired quality. Software developers and privacy experts should work closely together in this phase to avoid problems such as an improper choice of libraries with unwanted effects (such as the use of logging of data including personal information or the presence of vulnerabilities or leaks).

It remains unclear if testing procedures, such as penetration testing and fuzzing, can be used for privacy purposes. Nevertheless, the methods used in code reviews also offer valuable information about data and information flow in programs and about the presence and enforcement of privacy-enhancing mechanisms.

11.4.4.6 Release of system and education of stakeholders
The release of system and education of stakeholders phase is used to develop strategies in case vulnerabilities are discovered on release, and these strategies are carried over to the next phase. Strategies cover the assignment of responsibilities, emergency response methods and emergency assessments, technical actions, and communication strategies. Privacy cannot be protected simply by technical components; this holds for security as well.

The education of system stakeholders takes a significant role in this phase. Stakeholders are system administrators, operators, and system end users. Operation stakeholders need to know which data are processed by the system and what kinds of implications this might have for users. Technical protection might be useless in certain scenarios that might seem unlikely, yet operators should know them to be able to react in case they occur.

Data subjects need to be informed about how their data are processed and which tools they are provided to exercise their privacy rights. The released system should be accompanied by an according privacy disclosure that describes the use of personal data of the system; documentation, which details the tools that the system provides; and user communication tools like a brief explanation to addresses likely user questions. System administrators, users, and other personnel who may interact with the system need to be informed about how their actions affect the privacy of others and which actions can lead to privacy violations.

The Microsoft SDL proposes to validate the privacy standards of the system by a privacy advisor or a privacy seal of quality prior to release. A legal privacy expert should review the documents and overview the release process.

11.4.4.7 Technology and organizational response
The last phase is one of the most significant in the life cycle. It carries over the results from the release phase for rapid response strategies. Breaches might have a significant impact, and the team must be prepared to respond efficiently and timely to them, as they can occur unexpectedly. A response team must therefore develop a response plan that includes preparations for potential postrelease emergencies. The Canadian Office of the Privacy Commissioner (OPC) [97] proposes four steps for this phase:
• **Breach containment and preliminary assessment:** In this step, immediate actions to stop the breach are carried out and the investigation leader and a response team are assigned. Legal action against the attackers is suggested as well.

• **Evaluation** of the risks associated with the breach: In this step, the risks associated with the breach are evaluated and first actions are triggered. In case of compromised data, for example, sensitive data in plaintext will trigger different actions than encrypted data. Assessments can help identify the root cause, the foreseen harm, and the individuals affected and to find adequate mitigation strategies.

• **Notification:** Here the users are notified of possible consequences the breach might have. The notification should be as soon as (reasonably) possible and personal, by phone, e-mail, etc. In this step, additional organizations can also be informed, such as cyberdefense centers and credit card companies (if credit card data were stolen).

• **Prevention:** A prevention plan is defined at this point. The OPC suggests that the level of effort should reflect whether it was a systemic breach or an isolated instance. This step aims at fast communication and support strategies between companies and users. They help users to understand what possible consequences the breach may have and give them a transparent view of the emergency response strategies form the company.

Legal support should be expected to be needed to handle consequences and to initiate legal actions against the attackers. A root cause mitigation team could be engaged to investigate why a breach was possible and develop a mitigation plan that can be realized rapidly. This is an especially crucial step in battling cybercrime.

The life cycle, once implemented, will change with every iteration according to the needs of organization. With a proper privacy engineering organization in place, a solid privacy-by-design approach can be reached in system design as well as the operation of the IoT system. For smart cities, a proper privacy life cycle is a quality differentiator, one that needs just a first step to start and considerable practice to mature.

**Acknowledgments**

Ralf C. Staudemeyer was partly supported by the research pool of the Faculty of Computer Science and Mathematics of the University of Passau. Artemios G. Voyiatzis was partly supported by the Christian Doppler Forschungsgesellschaft through Josef Ressel Center project u’smile. Daniel Calvo was partly supported by the Internet of Everything Laboratory and the Energy and Transport Market of Atos Research and Innovation. Ralf C. Staudemeyer, George Moldovan, Santiago R. Suppan, Athanasios Lioumpas, and Daniel Calvo were partially supported by the European Union 7th Framework Programme under grant agreement number 609094 (RERUM).

**References**


23. P. Ramaswamy, IoT smart parking system for reducing green house gas emission, in International Conference on Recent Trends in Information Technology (ICRTIT’16), pp. 1–6, 2016.
35. J. P. Vasseur, RFC7102—Terms Used in Routing for Low-Power and Lossy Networks, no. 7102. RFC Editor, Jan 2014.
Chapter eleven: Smart cities under attack


74. E. Zeng, S. Mare, and F. Roesner, End User Security & Privacy Concerns with Smart Homes, in Symposium on Usable Privacy and Security (SOUPS), 2017.