"Appropriate Technical and Organizational Measures": Identifying Privacy Engineering Approaches to Meet GDPR Requirements

Completed Research

Dominik Huth
Technical University Munich
dominik.huth@tum.de

Florian Matthes
Technical University Munich
matthes@tum.de

Abstract

The General Data Protection Regulation requires, inter alia, the establishment of technical and organizational measures to ensure privacy properties. Software developers face the challenge of identifying these properties and suitable privacy enhancing techniques (PET). We conduct a literature study and identify eight privacy engineering approaches, which we analyze for their coverage of the GDPR privacy properties and for their support in software development phases. We conclude that recent privacy engineering approaches have the conceptual background to cover the GDPR, but advocate research on the integration of privacy concerns in software development processes.

Keywords

Privacy Engineering, GDPR, literature review, privacy properties

Introduction

In its 1998 report "Privacy Online", the Federal Trade Commission warned that "if growing consumer concerns about online privacy are not addressed, electronic commerce will not reach its full potential" (Landesberg et al. 1998). Despite the bursting of the Dotcom bubble only a couple of years later, most readers will agree that our everyday lives are hard to imagine without electronic commerce. Whether or not the report's Fair Information Practice Principles (FIPP - *Notice/Awareness*, *Choice/Consent*, *Access/Participation*, *Integrity/Security* and *Enforcement/Redress)* contributed to this success story remains an open question.

The state of online privacy is remarkably different today: As suggested by (Acquisti et al. 2015), data subjects do not have the power and knowledge to take up data holders like governments and corporations. Therefore, a baseline framework should be established that protects individual's privacy, regardless of the individual's potentially less-than-optimal decisions.

Such a baseline framework for the protection of European citizens has been established with the General Data Protection Regulation (GDPR) in 2016 (European Union 2016). The regulation adapts terminology to today's technological environment, establishes a unified territory and new data subject rights, and introduces additional documentation responsibilities for data controllers and data processors (Tikkinen-Piri et al. 2017). Most importantly, though, it dramatically increases possible fines for noncompliance and thus creates a business case for privacy.

Implementing legal requirements in the specification and development of software is a challenging task (Breaux et al. 2006). A recent study interviewed 27 software developers for their perceptions towards privacy (Hadar et al. 2018). The results show that developers most often define privacy in security terminology and tend to see privacy as a social concern, rather than an engineering concern.

The discipline of Privacy Engineering addresses privacy concerns in a more holistic way. As defined in (Gürses and Del Alamo 2016), the field "focuses on designing, implementing, adapting, and evaluating

theories, methods, techniques, and tools to systematically capture and address privacy issues in the development of socio- technical systems". In this work, we conduct a literature review to identify established privacy engineering approaches. The approaches are selected based on coverage of multiple aspects of privacy and on support during different stages of a typical software engineering process. We then assess the suitability of these privacy engineering approaches to address the technical requirements that are stated explicitly in the GDPR. Our contributions are as follows:

- We briefly review other work comparing approaches for operationalizing privacy requirements.
- We analyze the GDPR for references to "organizational and technical measures" for specific privacy properties. From these references, we create a consolidated list of privacy properties that must be addressed according to the GDPR.
- We conduct a literature review to identify Privacy Engineering approaches. We evaluate which of the identified requirements they address and in which phases of the software development cycle they provide support.

Related Work

(Kalloniatis et al. 2009) present ten methods for designing privacy aware systems. These include one method for modeling non-functional requirements (NFR), two methods for agent-based modeling (i*, Tropos), three methods for goal modeling (KAOS, GBRAM, M-N), one method for role-based access modeling (RBAC) and three broader privacy engineering approaches, which we also include in our work (Bellotti and Sellen 1993; Jensen et al. 2005; Kalloniatis et al. 2008). Despite the diverse nature of the approaches, the authors provide a valuable evaluation framework to assess the support that they provide. The areas of assessment are the requirements engineering process, the type of privacy issues that are addressed, the representation of the method, and the support for development. Of the presented methods, only the PriS method provides technical implementation guidance. According to the authors, formalization is an important aspect in privacy models, because it is the only way to prove privacy properties and quantify threats.

(Beckers 2012) defines a conceptual framework for privacy requirements engineering. This comparison framework includes the notions of individual stakeholder views, system requirements and threat analysis. The author then evaluates three privacy engineering approaches (Deng et al. 2011; Kalloniatis et al. 2008; Spiekermann and Cranor 2009) for completeness with respect to his framework. As privacy properties, he uses anonymity, unlinkability, undetectability and unobservability, but also mentions the additional properties that are reflected in the individual approaches. As a benefit of the comparison in (Beckers 2012), the author states the support in identifying a suitable privacy engineering approach for a specific project and the extensibility of the comparison with additional privacy engineering approaches or additional properties.

Both comparisons of privacy engineering approaches provide valuable concepts for our work, such as the comparison against a set of privacy requirements. However, they fall short of our objective in two ways:

- Their privacy requirements are not fine-grained enough or do not cover all requirements that are stated in the GDPR. (Kalloniatis et al. 2009) only checks for the abstract, overarching categories *privacy requirements* and *privacy goals*. (Beckers 2012) derives a complete list of privacy requirements, but does not compare these requirements to specific privacy regulation.
- Since the publication of the comparisons, additional privacy engineering approaches have been developed (Hoepman 2014; Notario et al. 2015). These approaches differ from the previous work by including support on how to implement solutions to identified privacy requirements.

Requirements from the Regulation

Unfortunately, there is no easy answer to the question "what has to be done in order to comply with the GDPR". Since the GDPR is a legal document, many of its requirements are interpretable in the sense that they have to account for future developments and court decisions. This interpretable nature of the rules is

stated as "the appropriate organizational and technological measures" to fulfil a certain privacy property. In this work, we focus only on these measures.

To this end, we conducted text analysis in the legal document for references to "technical and organizational measures" and extracted the particular privacy properties that they referred to within the text. The analysis resulted in Table 1. We defined three categories for the identified privacy properties: properties that serve as umbrella terms and would have to be defined in more detail ("General"), terms that name specific, well established privacy properties, e.g. from (Pfitzmann and Hansen 2010) ("PP"), and properties that refer specifically to the fulfilment of data subject requests ("DSR"). The reason we chose to define a separate category for data subject requests is the specificity of the regulation in this regard. We do not cover the DSR category, because we did not find meaningful support on how to handle these requests within the identified Privacy Engineering approaches.

Reference	**Properties**	**Category**
Recital 29	Pseudonymisation, unlinkability, authorization	PP
Recital 66	Distribute data subject requests to processors	DSR
Recital 67	Restriction of processing	DSR
Recital 68	Data portability request	DSR
Recital 71	Accuracy of data	PP
Recital 78	Data minimization, pseudonymization, information	PP
Recital 81	Security	General
Recital 88	Protect data	General
Recital 156	Data minimization	PP
Art. 4 (5)	Pseudonymity	PP
Art. 5 (1) e	Non-identifiability	PP
Art. 5 (1) e	Storage limitation	PP
Art. 5 (1) f	Integrity and confidentiality	PP
Art. 17 (2)	Distribute data subject requests to processors	DSR
Art. 24 (1)	Demonstrate compliance	PP
Art. 24 (2)	Purpose limitation	PP
Art. 25 (1)	Pseudonymisation	PP
Art. 25 (2)	Data minimization	PP
Art. 28 (1)	meet the requirements of this regulation	General
Art. 28 (3) e	Distribute and execute data subject requests	DSR
Art. 28 (4)	meet the requirements of this regulation	General
Art. 32 (1) a	Pseudonymization	PP
Art. 32 (1) a	Encryption	PP
Art. 32 (1) b	Confidentiality, integrity, availability, resilience	PP
Art. 32 (1) c	access	PP
Art. 34 (3) a	render data unintelligible – (encryption, unlinkability)	PP
Art. 83 (2) d	Technical measures will be taken into account when determining fines	General

Table 1: GDPR references to "technical and organizational measures"

We derive 12 privacy properties from the list and define them as follows, adopting the definition according to the GDPR if possible. Otherwise, we refer to the definitions in ("ISO/IEC 27000:2018" 2018) or the privacy terminology in (Pfitzmann and Hansen 2010). In the course of this paper, we look for these privacy properties in the identified privacy engineering approaches. This will help us to determine their suitability for fulfilling the GDPR requirements.

- **Pseudonymity/Non-identifiabiliy:** *"Processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information"* (GDPR).
- **Unlinkability:** *"Inability of an attacker to determine whether two items of interest (IOI) are related or not."* (adapted from (Pfitzmann and Hansen 2010))
- **Access control/Authorization:** *"means to ensure that access to assets is authorized and restricted based on business and security requirements*" (ISO)
- **Integrity:** *"protection against accidental loss, destruction or damage"* (GDPR**)**
- **Confidentiality:** *"Protection against unauthorized or unlawful processing"* (GDPR), *"property that information is not made available or disclosed to unauthorized individuals, entities, or processes"* (ISO)
- **Availability/Access:** *"property of being accessible and usable on demand by an authorized entity"* (ISO)
- **Data minimization:** *"adequate, relevant and limited to what is necessary"* (GDPR)
- **Information/transparency:** *"processed in a transparent manner"* (GDPR).
- **Storage limitation:** storing data *"no longer than is necessary for the purposes for which the personal data are processed"* (GDPR)
- **Purpose limitation:** *"collected for specified, explicit and legitimate purposes"* (GDPR)
- **Accountability:** *"demonstrate that processing is performed in accordance with the regulation"* (GDPR)
- **Encryption:** *"protection measures that render data unintelligible to any person who is not authorised to access it"* (GDPR)

Privacy Engineering Approaches

We conducted our literature search on *Scopus*, using a combination of the terms *privacy requirement engineering* and *technical solution*. The search resulted in 140 publications, of which we analyzed the titles and abstracts, reducing the selection to 49 and 27 publications, respectively. Forward and backward search added another 14 publications, resulting in 41 publications to review. In this paper we present and analyze eight publications that include approaches to privacy engineering. We define *privacy engineering approaches* as approaches that cover more than one privacy property (such as pseudonymity) and support more than one phase in a generalized software development cycle. Various cross-references among these scientific publications confirm the validity of this selection.

For each of the approaches, we conducted a critical analysis in order to find the covered privacy requirements and the covered phases of a software development cycle. We present a short summary of the approaches in chronological order in this section.

Belloti & Sellen (1993)

Motivating their work with increasing information storage capabilities, the work of (Bellotti and Sellen 1993) is still relevant today. The authors report on an experiment within a research project. There are cameras and microphones in every participant's office, which allows other participants to either establish video conversation or check on someone else's availability without any interaction. Despite the low concern for privacy among the system users, the authors identify the principles *control* and *feedback* to be important in the experimental setup. In the design framework, these two principles have to be addressed for the four system behaviors *capture*, *construction*, *accessibility* and *purpose*.[1]

[1] Interestingly, these four behaviors align well with Solove's taxonomy (Solove 2006).

For the resulting eight design questions, the authors propose evaluating a possible design for eleven further criteria. The authors do not claim to support in generating a design solution, but help point to a solution by clarifying problems.

Hong et al. (2004)

(Hong et al. 2004) point out that privacy is not an absolute value, but a matter of values and personal preferences. In the context of ubiquitous computing, the authors propose a two-step method of (1) identifying privacy risks and (2) prioritizing the identified risks, in order to provide a reasonable level of privacy that depends on the stakeholders of the intended system. The prioritization is intended to help designers develop suitable architectures and interaction techniques.

The privacy risks are identified through a series of questions about the organizational context and the technology to be used. For the organizational context, it is important to identify the users and their relationships, the type of personal information and the value proposition for sharing it, and the potential for malicious observers. For the technology, questions about should be asked about how the information is collected, whether the user has control over how the information is used, how precise the collected information is, and for how long it will be stored. For assessing the risks, the authors suggest using the three factors probability of event P, damage of event D and cost of preventing the event C. In general, the countermeasures should be applied when the expected damage $P*D$ exceeds the cost C. More specifically, the decisions can be based on the disclosure scenario, feedback and control mechanisms, possibilities for data protection of breach discovery, and the ability to maintain plausible deniability. The presented case studies mention how countermeasures for some risks were implemented, but the framework does not propose specific countermeasures.

Jensen et al. (2005)

(Jensen et al. 2005) analyze previous approaches for privacy aware design. They conclude that they do not address specific implementation issues and do not take into account the iterative nature of software development. These concerns are addressed with the framework STRAP. The framework has four steps: analysis, refinement, evaluation and iteration.

The authors argue that a goal-oriented analysis of the system should be the starting point for privacy risk analysis, which is conducted in a similar manner as (Bellotti and Sellen 1993) and (Hong et al. 2004). In the design refinement step, the authors propose eliminating or mitigating risks by enabling technical measures, such as database encryption, or changing the goal model. However, they stress that the privacy measures should not interfere excessively with the task the user wants to accomplish. The evaluation step includes comparing multiple independent designs for their fit to the Fair Information Practice Principles. To account for changes in system features, the iteration step involves adapting the (already existing) goal model and starting over with the analysis.

The framework is evaluated in a lab experiment against the Bellotti & Sellen framework, indicating that more vulnerabilities are identified in the same amount of time.

Kalloniatis et al. (2008)

According to (Kalloniatis et al. 2008), research efforts in the privacy domain focus either on requirements engineering or privacy enhancing techniques (PET), but do not link the identified requirements to possible implementations. To bridge this gap, they propose the method PriS.

The method draws from concepts of enterprise modeling, namely organizational goals and goal models, the processes that operationalize these goals, and the software systems to support them. In this context, the authors derive eight privacy goals from literature: identification, authentication, authorization, data protection, anonymity, pseudonymity, unlinkability and unobservability. Further, they propose so-called privacy-process patterns (modeled in BPMN) that address each of these goals.

In the first step of the method, the privacy goals that are relevant to the organization are elicited. These privacy goals are evaluated against other organizational goals in the second step. The objective is to identify the affected organizational goals and, thereby, the affected processes. Each process is modeled using the

proposed privacy-process patterns in step three. Finally, the knowledge of where in a process a PET has to be applied supports in selecting the appropriate technique. A detailed table with implementation techniques and their relationship to the privacy goals is given. The categories of techniques are administrative tools, information tools, anonymizer and pseudonymizer tools, track and evidence erasers and encryption tools. The method is also formally defined.

Spiekermann and Cranor (2009)

(Spiekermann and Cranor 2009) first frame privacy in order to support understanding of their framework. An important realization is that processing[2] takes place in three different spheres: The user sphere, where the data owner is in full control of the processing; the recipient sphere, where the data controller has full control; and the joint sphere, where the user can influence how data is being processed, although the controller has technical control over it. Additionally, the authors present three system activities that can take place between or within these spheres: transfer, storage and processing.

The authors argue that privacy friendly system design always depends on the privacy expectations of the user. With this in mind, the Framework for Privacy-Friendly System Design is presented. Its main dimensions are the level of identifiability, the level of linkability and the approach to privacy protection. The latter represents a spectrum between privacy by policy, mainly based on the principles of notice and choice, and privacy by architecture. Even though references to specific techniques are given (e.g. k-anonymity), there are no instructions on how privacy by architecture can be implemented. Therefore, the presented framework has only limited use for guiding developers in the selection of data protection measures. It contributes to the understanding of the domain and raises awareness for possible issues that must be considered.

Deng et al. (2010)

The authors of (Deng et al. 2011) argue that no comprehensive privacy threat modeling framework exists that provides guidance in modeling threats, eliciting requirements and identifying countermeasures in the *privacy* domain. In the security domain, the Microsoft STRIDE[3] framework serves these purposes. The approach of STRIDE is transferred to the privacy domain in the specification of the LINDDUN framework. To this end, the threats to the widely accepted privacy terminology in (Pfitzmann and Hansen 2010) are defined as ***l****inkability*, ***i****dentifiability*, ***n****on-repudiation*, ***d****etectability* and ***d****isclosure of information*. Additionally, the authors consider the threats *content* ***u****nawareness* and *policy and consent* ***n****oncompliance*. Together, these yield the acronym LINDDUN.

The approach consists in defining a data flow diagram and mapping the seven privacy threats to the data flow diagram elements, i.e. entities, data flows, data stores and processes. The privacy risks are identified with the support of an extensive set of threat tree patterns. After prioritizing the risks, the corresponding privacy requirements are elicited. The authors note that solution strategies for responding to the privacy risks can include simply removing features, but finding a countermeasure is typically the more desirable approach. The LINDDUN framework relies on catalogues of PETs, as specified in (Kalloniatis et al. 2008), to identify such solutions.

Hoepman (2014)

In line with other publications, (Hoepman 2014) asserts that developers have a large amount of PETs to draw from, but are missing support to incorporate privacy concerns in the early phases of a software development project. Hoepman points out the relationship between the software development concepts of *design strategies*, which he defines as "fundamental approach[es] to achieve a certain design goal", and *design patterns*, which he defines as "a scheme for refining […] a software system" and "a commonly recurring structure […] that solves a general design problem". The design goal of a *privacy design strategy*

[2] Here, processing refers to any action performed on personal data.

[3] An acronym of the security threats spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege.

is to achieve some level of privacy, while a *privacy design pattern* represents the abstract concept that is implemented by a PET.

The privacy design strategies are derived from the OECD privacy guidelines, the (then preliminary) draft of the GDPR, and the ISO 29100 privacy framework. Thus, there is a perfect fit of the privacy design strategies to the list we provide above. Specifically, the eight privacy design strategies are *minimize*, *hide*, *separate*, *aggregate*, *inform*, *control*, *enforce* and *demonstrate*.

Subsequent research (Colesky et al. 2016) enhances the privacy design strategies with tactics[4], which are refinements of how a particular privacy design strategy is applied. Tactics are an additional abstraction between privacy design strategies and privacy design patterns. In summary, privacy design strategies present a hierarchical pipeline: strategies specify a privacy goal, tactics contribute to an overarching strategy, patterns[5] describe an abstract implementation of a tactic, and PETs implement privacy patterns. Privacy design strategies do not specify a process in its own, but provide the supporting terminology to address privacy concerns along the software development process.

Notario et al. (2015)

(Notario et al. 2015) identify two complementary approaches to privacy analysis: goal-based approaches, such as (Kalloniatis et al. 2008), and risk-based approaches, such as (Deng et al. 2011). In their method PRIPARE, the authors aim to support a wide range of stakeholders along the entire software development lifecycle. To acknowledge different possible backgrounds of these stakeholders, the method provides multiple variants for conducting each of the following method steps:

For the analysis of privacy requirements (step 1), the method handbook refers to the criteria in ("ISO/IEC 27000:2018" 2018), which are a subset of the requirements we identified in the previous section. The prioritized requirements are then incorporated into a privacy aware system design (step 2), e.g. using privacy design strategies (Hoepman 2014). However, there is no explicit link to specific techniques in the implementation (step 3).

A notable feature of the method is that it accounts for later phases of the software development lifecycle. The verification step (4) advises checking for documentation or formal verification of the design. The release step (5) accounts for new releases or modifications that affect the privacy properties of a system. It includes activities like privacy impact assessments or incident response plans. In the maintenance step (6), or rather productive phase of a system, possible incidents must be handled. Decomissioning (step 7) includes secure deletion of personal data. As stated in (Martin and Kung 2018), the group plans to integrate data protection principles in general-purpose software engineering tools.

Discussion

Our view is that the conceptualization of privacy in the GDPR contributes to the confusion about which measures have to be taken. An example is the missing structure in the privacy terminology: Our list includes high-level goals, such as confidentiality, as well as implementation techniques, in particular encryption, without acknowledging the relationship between the two. The work we analyzed provides clear conceptual models that also help in understanding the GDPR structure.

Overall we observed an increasing level of detail in the privacy engineering approaches as time progresses. Privacy requirements, measured against GDPR requirements, and development support are covered more in-depth in more recent approaches. It is, however, not our intention to rate approaches based on their conceptual clarity. As shown in (Jensen et al. 2005), it is extremely difficult to create meaningful evidence and draw conclusions about the effectiveness of privacy engineering approaches. Thus, our aim is to analyze the features of the approaches and put them into context.

[4] Related, but not identical to the notion of tactics in software architecture

[5] A pattern catalog that builds on privacy design strategies and contains the patterns described in (Hoepman 2014) is available at www.privacypatterns.org

In terms of "GDPR completeness", we present Table 2. In (Bellotti and Sellen 1993), the design guidelines were developed based on the concerns that emerged in a confined experiment. In this setting, the principles of *feedback* and *control* are important, while other privacy properties and properties from the intersection of privacy and security are not covered. (Jensen et al. 2005) make use of requirements engineering concepts and base their framework on the FIPs (Landesberg et al. 1998). Later approaches draw from comprehensive privacy concepts, which explains their larger coverage of the GDPR requirements (Deng et al. 2011; Hoepman 2014; Kalloniatis et al. 2008; Spiekermann and Cranor 2009). The work we found to be most extensive in this context are the privacy design strategies.

	Pseudonymity	**Unlinkability**	**Access control / Authorization**	**Integrity**	**Confidentiality**	**Availability / Access**	**Data minimization**	**Information / transparency**	**Storage limitation**	**Purpose limitation**	**Accountability**	**Encryption**
(Bellotti and Sellen 1993)	○	○	○	○	○	○	○	●	○	●	●	○
(Hong et al. 2004)	●	○	●	○	●	○	○	●	●	●	○	○
(Jensen et al. 2005)	○	○	○	●	●	●	●	●	●	○	●	●
(Kalloniatis et al. 2008)	●	●	●	●	●	●	○	○	○	○	○	●
(Spiekermann and Cranor 2009)	●	●	●	○	○	●	●	●	○	●	●	●
(Deng et al. 2011)	●	●	○	○	●	○	●	●	◐	◐	●	●
(Hoepman 2014)	●	●	●	●	●	●	●	●	●	●	●	●
(Notario et al. 2015)	○	○	○	●	●	●	●	●	●	●	●	◐

Table 2: Coverage of GDPR privacy properties

An important distinction we found in the approaches is between *privacy by architecture* and *privacy by policy* (Spiekermann and Cranor 2009) and *hard privacy* and *soft privacy* (Deng et al. 2011). We see this as a rough association to the technical and organizational measures in the regulation.

	Concept	**Analysis**	**Design**	**Impl.**	**Test**	**Eval.**
(Bellotti and Sellen 1993)	○	●	●	○	○	○
(Hong et al. 2004)	●	●	●	○	○	○
(Jensen et al. 2005)	●	●	●	◐	○	○
(Kalloniatis et al. 2008)	●	●	●	●	○	○
(Spiekermann and Cranor 2009)	●	●	○	○	○	○
(Deng et al. 2011)	○	●	●	●	○	○
(Hoepman 2014)	●	●	●	●	○	○
(Notario et al. 2015)	●	●	●	◐	●	●

Table 3: Support of development process

For developer support (cf. Table 3), we found that earlier approaches focused on creating awareness for privacy concerns and eliciting corresponding requirements, while later approaches also provide links to solutions. A particular blind spot we observed is the lack of support for the operations phase that follows the system development phase. The PRIPARE method suggests formal verification of the desired properties, and refers to dedicated languages to express privacy properties, such as *SIMPL* or *S4P*. (Colesky et al. 2016) define the privacy design strategy *demonstrate,* but the authors state that there are currently no privacy patterns that implement this strategy.

Conclusion & Future Research

In this paper, we conducted a text analysis of the GDPR and identified twelve privacy requirements that must be addressed by "appropriate technical and organizational measures". We then identified eight privacy engineering approaches and evaluated them with respect to coverage of the GDPR requirements and support during the software development phases. We conclude that comprehensive theoretical foundations for designing privacy-aware systems are already in place and identify two areas for further research:

Our next step is to investigate how privacy compliance can be achieved in development processes. To this end, we developed a prototype that assigns roles and necessary privacy properties within a project. The role *developer* can then use enhanced navigation functionalities to identify suitable privacy patterns. Preliminary industry feedback indicates that this approach could support documentation and traceability of privacy properties, and offer a lightweight alternative to extensive developer guidelines. We will refine and evaluate the underlying process and the prototype in an industry study.

Secondly, as we can see in Table 1, the statement "technical and organizational measures" sometimes refers to the data subject rights, marked as DSR. Even though the approaches incorporate the principles of notice and choice, they do not provide detailed support of these data subject rights (access, rectification, deletion, restriction/objection, data portability). Since the data subject rights have gained importance with the GDPR, we advocate research towards this topic. This could e.g. be specified in a similar way to the privacy process patterns by (Kalloniatis et al. 2008).

Acknowledgment

This work has been sponsored by the German Federal Ministry of Education and Research (BMBF) grant 01IS17049 / UMEDA. The responsibility for the content of this publication lies with the author.

REFERENCES

Acquisti, A., Brandimarte, L., and Loewenstein, G. 2015. "Privacy and Human Behavior in the Age of Information," *Science* (347:6221), pp. 509–515. (https://doi.org/10.2139/ssrn.2580411).

Beckers, K. 2012. "Comparing Privacy Requirements Engineering Approaches," *Proceedings - 2012 7th International Conference on Availability, Reliability and Security, ARES 2012*, pp. 574–581. (https://doi.org/10.1109/ARES.2012.29).

Bellotti, V., and Sellen, A. 1993. "Design for Privacy in Ubiquitous Computing Environments," in *Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13–17 September 1993, Milan, Italy ECSCW '93*, pp. 77–92. (https://doi.org/10.1007/978-94-011-2094-4_6).

Breaux, T. D., Vail, M. W., and Antón, A. I. 2006. "Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations," *Proceedings of the IEEE International Conference on Requirements Engineering*, pp. 46–55. (https://doi.org/10.1109/RE.2006.68).

Colesky, M., Hoepman, J.-H., and Hillen, C. 2016. "A Critical Analysis of Privacy Design Strategies," in *Proceedings - 2016 IEEE Symposium on Security and Privacy Workshops, SPW 2016*, pp. 33–40. (https://doi.org/10.1109/SPW.2016.23).

Deng, M., Wuyts, K., Scandariato, R., and Wouter, B. P. 2011. "A Privacy Threat Analysis Framework: Supporting the Elicitation and Fulfillment of Privacy Requirements," *Requirements Engineering* (16:1), pp. 3–32. (https://doi.org/10.1007/s00766-010-0115-7).

European Union. 2016. "Regulation 2016/679 of the European Parliament and the Council of the European Union," *Official Journal of the European Union*, pp. 1–88. (https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02016R0679-20160504).

Gürses, S., and Del Alamo, J. M. 2016. "Privacy Engineering: Shaping an Emerging Field of Research and Practice," *IEEE Security and Privacy* (14:2), pp. 40–46. (https://doi.org/10.1109/MSP.2016.37).

Hadar, I., Hasson, T., Ayalon, O., Toch, E., Birnhack, M., Sherman, S., and Balissa, A. 2018. "Privacy by Designers: Software Developers' Privacy Mindset," *Empirical Software Engineering* (23:1), Empirical Software Engineering, pp. 259–289. (https://doi.org/10.1007/s10664-017-9517-1).

Hoepman, J.-H. 2014. "Privacy Design Strategies," in *IFIP International Information Security Conference*, Berlin, Heidelberg: Springer, pp. 446–459. (https://doi.org/10.1007/978-3-642-55415-5).

Hong, J. I., Ng, J. D., Lederer, S., and Landay, J. A. 2004. "Privacy Risk Models for Designing Privacy-Sensitive Ubiquitous Computing Systems," *Proceedings of the 2004 Conference on Designing Interactive Systems Processes, Practices, Methods, and Techniques - DIS '04*, p. 91. (https://doi.org/10.1145/1013115.1013129).

"ISO/IEC 27000:2018." 2018. (https://www.iso.org/obp/ui/#iso:std:iso-iec:27000:ed-5:v1:en, accessed February 6, 2019).

Jensen, C., Tullio, J., Potts, C., and Mynatt, E. D. 2005. *STRAP: A Structured Analysis Framework for Privacy*. (http://smartech.gatech.edu/handle/1853/4450).

Kalloniatis, C., Kavakli, E., and Gritzalis, S. 2008. "Addressing Privacy Requirements in System Design: The PriS Method," *Requirements Engineering* (13:3), pp. 241–255. (https://doi.org/10.1007/s00766-008-0067-3).

Kalloniatis, C., Kavakli, E., and Gritzalis, S. 2009. "Methods for Designing Privacy Aware Information Systems: A Review," in *PCI 2009 - 13th Panhellenic Conference on Informatics*, IEEE, pp. 185–194. (https://doi.org/10.1109/PCI.2009.45).

Landesberg, M. K., Levin, T. M., Curtin, C. G., and Lev, O. 1998. "Privacy Online : A Report To Congress." (https://www.ftc.gov/sites/default/files/documents/reports/privacy-online-report-congress/priv-23a.pdf).

Martin, Y. S., and Kung, A. 2018. "Methods and Tools for GDPR Compliance Through Privacy and Data Protection Engineering," in *Proceedings - 3rd IEEE European Symposium on Security and Privacy Workshops, EURO S and PW 2018*, IEEE, pp. 108–111. (https://doi.org/10.1109/EuroSPW.2018.00021).

Notario, N., Crespo, A., Martin, Y. S., Del Alamo, J. M., Metayer, D. Le, Antignac, T., Kung, A., Kroener, I., and Wright, D. 2015. "PRIPARE: Integrating Privacy Best Practices into a Privacy Engineering Methodology," in *Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015*, pp. 151–158. (https://doi.org/10.1109/SPW.2015.22).

Pfitzmann, A., and Hansen, M. 2010. "A Terminology for Talking about Privacy by Data Minimization : Pseudonymity , and Identity Management." (https://doi.org/10.1.1.154.635).

Solove, D. J. 2006. "A Taxonomy of Privacy," *University of Pennsylvania Law Review* (154:3), p. 477. (https://doi.org/10.2307/40041279).

Spiekermann, S., and Cranor, L. F. 2009. "Engineering Privacy," *IEEE Transactions on Software Engineering* (35:1), pp. 67–82. (https://doi.org/10.1109/TSE.2008.88).

Tikkinen-Piri, C., Rohunen, A., and Markkula, J. 2017. "EU General Data Protection Regulation: Changes and Implications for Personal Data Collecting Companies," *Computer Law and Security Review* (1:2017), Elsevier Ltd. (https://doi.org/10.1016/j.clsr.2017.05.015).