IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 10, 2020, accepted April 3, 2020, date of publication April 16, 2020, date of current version May 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988408

A Systematic Mapping Study on Software Quality Control Techniques for Assessing Privacy in Information Systems

DANNY S. GUAMÁN[1,2], JOSE M. DEL ALAMO[1], AND JULIO C. CAIZA[1,2]
[1]Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[2]Departamento de Electrónica, Telecomunicaciones y Redes de Información, Escuela Politécnica Nacional, Quito 170517, Ecuador

Corresponding author: Jose M. Del Alamo (jm.delalamo@upm.es)

The work of Danny S. Guamán and Julio C. Caiza was supported by Escuela Politécnica Nacional. This work was also partially supported by the CLIIP Project funded by the Comunidad de Madrid and Universidad Politécnica de Madrid through the V-PRICIT Research Programme Apoyo a la realización de Proyectos de ICD para jóvenes investigadores UPM-CAM, under Grant APOYO-JOVENES-QINIM8-72-PKGQ0J.

ABSTRACT Software Quality Control (SQC) techniques are widely used throughout the software development process with the objective of assessing and detecting anomalies that affect the quality of an information system. Privacy is one quality attribute of software systems for which several SQC techniques have been proposed in recent years. However, research has been carried out from different perspectives and, consequently, it has led to a growing body of knowledge scattered across different domains. To bridge this gap, we have carried out a systematic mapping study to provide practitioners and researchers with an overview of the state-of-the-art techniques to carry out software quality control of information systems focusing on aspects of privacy. Our results show a steady growth in the research efforts in this field. The European General Data Protection Regulation seems to have a significant influence on this growth, since 37% of techniques that focus on assessing compliance derive their assessment criteria from this legal framework. The maturity of the techniques varies between the type of technique: Formal verification techniques exhibit the lowest level of maturity while the combination of techniques has demonstrated its successful application in real-world scenarios. The latter seems a promising avenue of research as it provides better results in terms of coverage, precision and effectiveness than the application of individual, isolated techniques. In this paper, we describe the existing SQC techniques focusing on privacy and provide a suitable basis for identifying future research directions.

INDEX TERMS Data protection, information systems, mapping, privacy, software quality control, software engineering, systematic study.

I. INTRODUCTION

Software Quality Control (SQC) [1] includes a set of activities that evaluates information systems (IS) throughout the entire development process to detect anomalies that may negatively affect software quality. While software quality involves several attributes, including security, reliability, and usability [2], privacy is another, recently addressed attribute that may affect software quality [3]. The propensity to embed this attribute during the development of an IS has been reflected in the Privacy by Design paradigm [4], which has been incorporated into several privacy regulations worldwide, such as article 25 of the European General Data Protection Regulation (GDPR) [5]. Like other quality attributes, privacy also needs to be evaluated throughout the development lifecycle of an IS in order to detect anomalies that could undermine it. Evaluation activities are even highlighted in some way though legal texts, e.g. GDPR Art. 42 establishes the need for mechanisms to evaluate and demonstrate the compliance of the developed IS.

There are several types of SQC techniques. Static techniques aim at assessing and finding anomalies in different representations or models of the system. Dynamic techniques examine the real behaviour of the system at runtime [1], [6]. All of these techniques require the criteria or conditions that circumscribe when a particular quality attribute is preserved or violated to be defined in order to report an anomaly. However, the definition of these criteria or conditions is not

The associate editor coordinating the review of this manuscript and approving it for publication was Yan Huo.

a simple matter when dealing with privacy due to the plurality [7] and contestability [8] of the concept. For example, some SQC techniques can assume privacy as personal data confidentiality [9] and then report an anomaly when any exposure of such data occurs in the IS being evaluated. Other techniques can assume privacy as contextual integrity [10] and then report an anomaly only when certain flows of personal data specific to a particular context are not met.

In this context, research efforts into SQC techniques focusing on the detection of privacy-related anomalies have been conducted from different perspectives and are scattered across several research communities. This has resulted in a growing body of knowledge and a number of workshops, symposia, conference tracks and publications spread across a variety of domains. Some previous works have provided an overview of the SQC techniques focusing on other quality attributes (e.g. [11], [12]) or have focused on a specific SQC technique (e.g. [13], [14]). However, to the best of our knowledge, the state of the art still lacks an overview of SQC techniques that can support engineers in detecting privacy-related anomalies throughout the development of IS.

This paper presents the first overview of the different SQC techniques used to detect privacy-related anomalies throughout the development process of IS, obtained through a systematic mapping study. This paper also identifies the concrete software artefacts to which the SQC techniques are applied, as well as the targeted privacy properties and conditions used to detect anomalies. Finally, it provides quantitative evidence that support trends, an analysis of the level of maturity of the SQC techniques, and institutions in which research is conducted, and venues where results in the field are published.

The rest of the paper is organised as follows: Section 2 provides a background into SQC techniques and the related concepts that have been considered in this study. Section 3 presents the research method, i.e. the underlying process of the flow and tasks of the systematic mapping study, including the research questions and classification scheme. Section 4 provides the results of the mapping study, responding to the research questions formulated. Section 5 presents a trend analysis, discusses the results and outlines the gaps observed. Section 6 examines the potential threats to the construct, internal and external validity, and how we have dealt with them. Section 7 summarises the existing surveys and highlights the differences with our study. Finally, Section 8 presents the conclusions.

II. BACKGROUND

This section provides a summary of the fundamental concepts for understanding this study. First, it briefly describes the types and sub-types of SQC techniques, as well as other related concepts. Second, it describes a classification to determine the level of maturity of the contributions presented.

A. SQC TECHNIQUES

The Software quality control (SQC) domain, according to the SWEBOK [1], includes the techniques used for evaluating intermediate and final software products for the purpose of detecting anomalies. The term anomaly, which can adversely affect software quality, is a broad term which, in this study, implies that an IS (1) does not comply with its requirements or specifications, which may be derived from users' expectations, policies or regulations; or, (2) contains vulnerabilities or deficiencies due to design issues, incorrect steps, process or data definition in the source code, or an improper system configuration [1], [15]. This domain encompasses techniques for detecting anomalies rather than mitigating them, and the target of the evaluation is a software product rather than the development process.

TABLE 1. Subtypes of software quality control techniques.

Type of technique	Subtype	Description
Static	Formal verification	If it involves rigorous mathematical proofs of correctness to verify a property or formal specification.
	Inspection/ review	If it involves a manual or automated analysis and checking of software system requirements, design models or code artifacts without executing them.
Dynamic	Testing	If it captures the real behavior or operation of the software system in a controlled test environment and then compares it to the (un)expected behavior.
	Monitoring	If it captures the real behavior or operation of the software system in a production environment and then compares it to the (un) expected behavior.
Combined	Compilation	If static and dynamic techniques are combined to be applied with a common goal but in isolation, i.e. without the output of one technique being needed for the second one.
	Integration	If static and dynamic techniques are combined to be applied with a common goal, and the output of one technique is used as an input to the second one.

An SQC technique can fall into one of three types, namely *static*, *dynamic* or *combined*. A *static* technique does not need to execute an IS but analyses any readable representation of it. A *dynamic* technique needs to execute the IS or component in order to observe its real behavior [1]. A *combined* technique uses both *static* and *dynamic* techniques [13]. Each type of technique can be broken down into the *subtypes* presented in Table 1. While the subtype *monitoring* was identified during the refinement process when building the classification scheme (Section III-D), *formal verification, inspection/review* and *testing* are defined in the IEEE standard for System, Software, and Hardware Verification and Validation [6], and, finally, *compilation* and *integration* are defined in [13].

There is a set of concepts related to the application of an SQC technique for assessing privacy, namely the *target*

TABLE 2. Targets of evaluation of static techniques.

Attribute	Description
Requirements artifact	If the evaluation is conducted on the requirements specification that captures a condition or capability to be met or possessed by a software system or component to satisfy user needs and expectations, standard, or other formally imposed document.
Design artifact	If the evaluation is conducted on any of the following artifacts: • *Software architecture*: if it is related to the overall structure of the software system, the main components, and their relationships (e.g. data flow between its components). • *Interface:* if it is related to the interfaces between components or user interfaces. • *Component:* if it is related to a particular software component (e.g. an algorithm to provide a particular functionality). • *Database:* if it is related to any type of data structure and its representation in a database.
Code artifact	If the evaluation is conducted on any kind of implementation including the coding of newly built software elements, existing software elements, third party software packages, or any combination of them. It then encompasses source code, object code or any intermediate representation, and configuration files.

TABLE 3. Sources of behavior of dynamic techniques.

Attribute	Description
Program	If the program behavior is captured by monitoring the traces of method invocations or data flows within the program under evaluation itself. It also includes programs instrumented with further log statements to capture behavior.
Execution environment	If the program behavior is captured by monitoring the calls the program makes to the execution environment. In this study, execution environment represents a particular execution platform, such as a virtual machine or an operating system.
Network	If the program behavior is captured by monitoring the network traffic relative to the program under evaluation.
Other communication channels	If the program behavior is captured by monitoring other sources than those mentioned above. For example, monitoring other communication channels such as Bluetooth.

of evaluation, the *evaluation objective* and the *privacy property assessed*. The *target of evaluation* determines the specific software artefact to which an SQC technique is applied. The targeted software artefacts vary according to the type of SQC technique. On the one hand, for static techniques, the *target of evaluation* can be any model of a software-based IS as described in Table 2 [16]. On the other hand, for dynamic techniques, the *target of evaluation* is the executable software-based IS or component (program), which is executed to capture its behaviour; Table 3 describes the possible *sources of behaviour*.

The *evaluation objective* determines the ultimate goal pursued by an SQC technique during the evaluation [1], [17], as described in Table 4.

Finally, the *privacy property* determines which aspect of privacy is addressed by an SQC technique. While there is still no single, widely accepted definition of privacy due to plurality [7], and the contestability [8] of the concept, we have used LINDDUN [18] and Hansen's [19] privacy properties which are widely recognised in the privacy engineering research community. Furthermore, they were selected because they provide a design-centered framework focused on the software system rather than on users or organizational aspects. Table 5 describes them.

TABLE 4. Objectives pursued by studied techniques.

Attribute	Description
Defect detection	If the evaluation aims to find vulnerabilities or deficiencies due to design issues, incorrect steps, processes or data definition in the source code, or improper system configuration that may potentially threaten a privacy aspect. .
Specification verification	If the evaluation aims to verify that a software system or component fulfils a specification, which can stem, e.g., from a privacy policy or regulation.
User validation	If the evaluation aims to validate that a software system meets the user's preferences and expectations. Note that user validation implies that "users or customers provide input and advice on system testing" [17], it therefore requires the execution of a software system in an (almost) real environment and, of course, user participation.

TABLE 5. Privacy properties [6], [7].

Privacy Prop.	Description
Identifiability (Anonymity /pseudonymity)	It "*means that the attacker cannot sufficiently identify the subject within a set of subjects, the anonymity set*." [6]
Undetectability /unobservability	It "*refers to hiding the user's activities*". E.g. it is impossible to know whether an entry in a database corresponds to a real user, or to distinguish whether someone or no one is in a given location. [6]
Unlinkability	"*Unlinkability of two or more items of interest (IOIs, e.g., subjects, messages, actions, etc.) from an attacker's perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not*." [9]
Plausible deniability	It refers to the ability of a system "*to repudiate having performed an action*". [6]
Non-disclosure of personal data	Exposure of personal data to individuals or providers who are supposed to have access to it. [6]
Awareness/Transparency	It refers to the measures that a system itself can take to guide and educate the user concerning his or her personal data processing (e.g. collection and disclosure) and nudge the user into a more privacy-aware use of the system. [6]
Intervenability	"*Intervenability is defined as the property that intervention is possible concerning all ongoing or planned [personal] data processing*" [7]. Intervenability enables, e.g., to exercise the individuals' rights to rectification and erasure of personal data, providing and withdrawing consent, and so on. Going beyond data protection rights, it enables users to control/decide what kind or information is processed about them.
Compliance	It refers to verifying and demonstrating that personal data processing meets data protection regulations and internal business policies. [6]
Privacy and data protection	If refers to privacy or data protection as a whole. We will use this value when the SQC technique does not mention a particular privacy property.

TABLE 6. Attributes of the dimension "maturity level" [20].

Type of research	Description
Evaluation Research	*"Techniques are implemented in practice and an evaluation of the technique is conducted. That means, it is shown how the technique is implemented in practice (solution implementation) and what are the consequences of the implementation in terms of benefits and drawbacks (implementation evaluation). This also includes to identify problems in industry."*
Validation Research	*"Techniques investigated are novel or a significant update and have not yet been implemented in practice. Techniques used are for example experiments, i.e., work done in the laboratory."*
Solution Proposal	*"A solution for a problem is proposed, the solution can be either novel or a significant extension of an existing technique. The potential benefits and the applicability of the solution is shown by a small example or a good line of argumentation."*
Philosophical Papers	*"These papers sketch a new way of looking at existing things by structuring the field in form of a taxonomy or conceptual framework."*
Opinion Papers	*"These papers express the personal opinion of somebody whether a certain technique is good or bad, or how things should be done. They do not rely on related work and research methodologies."*
Experience Papers	*"Experience papers explain on what and how something has been done in practice. It has to be the personal experience of the author."*

B. TYPES OF RESEARCH

A type of research describes the maturity of a study based on the efforts its authors took to validate it. Specifically, the attributes of this dimension correspond to the types of research set by Wieringa *et al.* [20]. As shown in Table 6, it enables empirically-proven SQC techniques (i.e., *evaluations* and *validations*) to be differentiated from those which are non-empirically proven (i.e., *solutions*, *philosophical*, *opinions* or *experiences*). During the refinement process, certain criteria were set to distinguish between *evaluations* and *validations,* as both are based on empirical proofs. Thus, a study was classified as *evaluation* only if three criteria were met: (i) the SQC technique has been applied in a real-world IS; (ii) the SQC technique has been used in a real-world context; and, (iii) the study explicitly shows the results of evaluating an SQC technique and suggests a mature rather than an evolving technique. If any of these criteria are not met, we classified it as *validation.*

III. METHOD: MAPPING STUDY

Our research followed the general guidelines for conducting systematic mapping study (SMS) studies in software engineering proposed by Petersen *et al.* [21]. An SMS is a systematic approach that aims to provide an overview of a research area of interest by showing quantitative evidence to identify trends. Adhering to the Petersen guidelines, we organised our tasks into a process flow consisting of three main stages: planning, conducting and reporting. Fig. 1 details the SMS process flow and the tasks carried out throughout this research.

The planning stage began with the formulation of the study scope, i.e. the main goal and research questions (RQs). On the basis of the RQs, we then defined the SMS protocol that includes the selection of the search strategy and search string, the exclusion and inclusion criteria for the candidate papers, the classification scheme and the extraction process, as well as the means for visualising the results, and the threats to validity. The protocol took into account lessons learned from Petersen [22] in order to ensure the quality of the study, such as peer-review validation, and the use of standards, well-known vocabularies and taxonomies of the field of research.

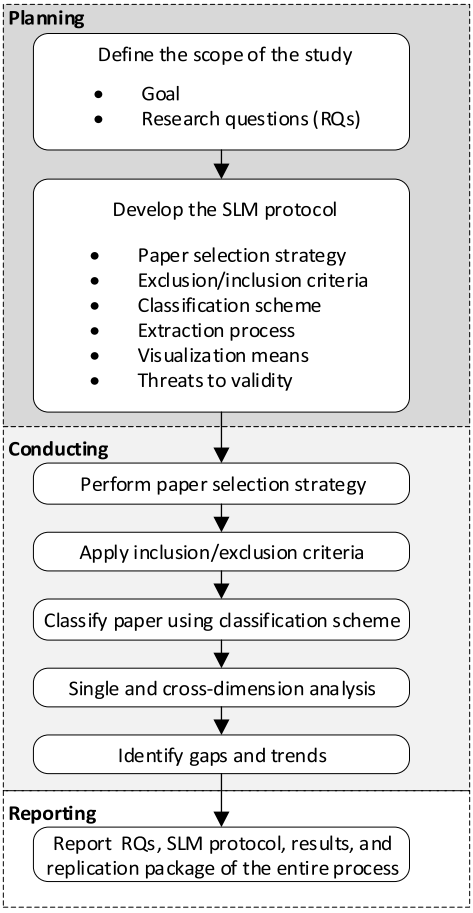

FIGURE 1. Process flow and tasks of the systematic mapping study.

The conducting stage followed the SMS protocol defined in the planning stage in order to answer the RQs. The first two tasks, i.e. the paper selection and the application of the inclusion/exclusion criteria, were carried out iteratively with the aim of refining the search string and the inclusion/exclusion criteria. Once stabilized, i.e. they passed a validation with a series of tests of the relevant papers and a peer review, we obtained a pool of candidate papers that were filtered by applying the inclusion/exclusion criteria. Next, the resulting subset of papers was coded using the classification scheme already defined and the results were analysed to answer the RQs.

Finally, in the reporting stage we analysed the results of the study by answering the RQs and attempting to identify the gaps and trends and consequently the directions for possible research. Further details on the different resources, data and

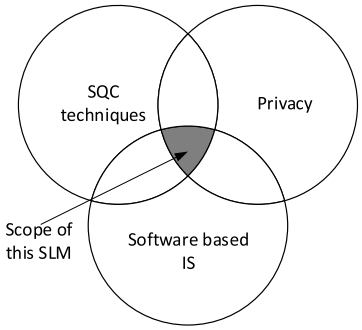

FIGURE 2. Inclusion and exclusion procedure.

procedures used can be found in the replication package at http://dx.doi.org/10.17632/zvp3986f5b.1

In this section we focus on detailing the scope of the study (Section A) and the SMS protocol planned and conducted, specifically the paper selection strategy (Section B), the inclusion/exclusion criteria and procedure (Section C), the classification scheme (Section D), and the classification procedure (Section E).

A. SCOPE OF THE STUDY

As illustrated in Figure 2, the scope of this study is centered on the intersection of three domains, namely (1) software quality control, (2) software-based information systems, and (3) privacy.

- The **Software quality control** domain includes the techniques used for evaluating intermediate and final software products for the purpose of detecting anomalies.
- The **Software-based information system** domain specifically covers application layer software, i.e., as Fielding describes, software that ''represent the business-aware functionality of a system'' 23], thus excluding software that mostly ignores business functionality, such as operating systems, networking software or cryptographic protocols.
- The **Privacy** domain represents the quality attribute on which this study focuses.

Therefore, the objective of this study is twofold: a) to identify and organize the state-of-the-art techniques systematically that can assist engineers in carrying out the software quality control of software-based IS focusing on privacy aspects; and, b) to provide suitable bases to outline the gaps and trends in this field of research and, subsequently, the directions for possible research to help fill them in. In order to achieve this objective, we have formulated the following research questions (RQ):

RQ1. What SQC techniques have been proposed for finding privacy-related anomalies throughout the lifecycle of software-based IS? The cornerstone of this SMS is to gain knowledge into the state-of-the-art SQC techniques for finding privacy-related anomalies and thus identify the gaps and trends in this field of research. The state-of-the-art SQC techniques can be helpful for practitioners such as developers or auditors. Gaps, trends and possible research directions can be leveraged by researchers in this area. In order to acquire more details into SQC techniques, this RQ has been broken down as follows:

RQ1.1. To which software artefacts are SQC techniques applied? As software-based IS can be evaluated throughout its entire lifecycle, our goal is to determine the specific software work products needed to apply the SQC techniques reported.

RQ1.2. Which particular objectives are pursued by the reported SQC techniques? This information allows us to determine the objectives pursued by the identified SQC techniques: defect detection, compliance verification and user validation.

RQ1.3. Which privacy properties are evaluated by the SQC techniques identified? There are multiple definitions of privacy, each described as a set of different properties. As each technique evaluates a given set of properties, our goal is to gain knowledge into the privacy properties that are evaluated by the reported SQC techniques.

RQ2. What are the types of research conducted by the studies identified? This information is valuable for rating the maturity of the reported SQC techniques based on the efforts their authors took to evaluate them. For this purpose, we have used the types of research proposed by Wieringa *et al.* [20] that assign a level of maturity to each paper ranging from a minimum level, when a contribution is based only on opinions, to a maximum level, when a contribution has been evaluated empirically in real scenarios.

RQ3. Which institutions contribute most in this area and what are the venues for publishing? This information is useful for researchers and practitioners to know places where to start looking for information into this research area.

B. PAPER SELECTION STRATEGY

We used a database-based strategy and the Scopus database to find high-quality refereed research literature, including journal and conference papers. Scopus indexes high-quality peer-reviewed papers from the main digital libraries used in the area of research we are interested in, including IEEE Xplore, Springer Link, Science Direct, and ACM. This database has also been used by other relevant papers reporting related systematic studies (e.g. [24], [25]). Furthermore, between Scopus and Web of Science (WoS), Cavacini states in his study *''What is the best database for computer science journal articles?''* that Scopus is better than WoS for identifying computer science publications [26].

Based on the SMS scope presented in Section A, we built the search string as a conjunction of the three aforementioned research domains (SQC techniques, software-based IS and privacy). We used well-known standards, vocabularies and taxonomies in the field of research, to build a search string capable of retrieving all related papers, including:

- top level terms (privacy and data protection) to have as many papers as possible,

TABLE 7. Fiinal search string.

```
("privacy" OR "data protection")
AND
("software" OR "system" OR "computing" OR
"computation" OR "computer" OR "informatics" OR
"application" OR "information")
AND
("quality assurance" OR "quality control" OR
"quality assessment" OR "validation" OR
"verification" OR "testing" OR "compliance" OR
"compliant" OR ("static" W/3 "analysis" ) OR (
"dynamic" W/3 "analysis") OR ( "static" W/3
"technique") OR ("dynamic" W/3 "technique") OR
("analysis" W/3 "privacy") OR ("evaluation" W/3
"privacy") OR ("assessment" W/3 "privacy"))
```

- synonyms of software-based IS obtained from taxonomies and vocabularies, including the 2017 IEEE Thesaurus Version 1.0 [27], ACM Computing Classification System 2012 [28], and Systems and Software Engineering – Vocabulary ISO/IEC/IEEE 24765 [15]), and,
- terms related to software quality control defined in the SWEBOK [1] and the IEEE Standard for System and Software Verification and Validation [6].

Our search string was the conjunction (AND) of the three research domains and each of them was, in turn, represented as a disjunction (OR) of all domain-related terms. Table 7 presents the final search string with the domains and related terms. This search string was coined after a four-iteration process, in which the results of each iteration were validated with a test-set of 15 relevant papers provided by an experienced privacy engineering researcher. The initial search string was refined through the retrieval of all related papers. In some cases, the search string was expanded to include more terms (e.g. ''compliance'', ''application'', ''app''), while in others, they were removed as they added so much noise (e.g. ''protocol''). Furthermore, in order to eliminate irrelevant papers when using isolated terms, we used compound terms (e.g. ''quality assurance'') and compound terms joined by the w/3 Scopus operation (e.g. ''static'' w/3 ''analysis'', which means that ''static'' can appear before or after ''analysis'' by no more than 3 words). Further details on the search string definition are presented in the replication package.

Once the search string was fully refined, a query was made to search within the title, abstract and keywords of papers obtaining 13,180 papers. This was carried out on July 24, 2019.

C. INCLUSION AND EXCLUSION PROCEDURE

We conducted an inclusion and exclusion procedure to further filter the candidate papers, as not all papers retrieved from the search query fall within the scope of our study.

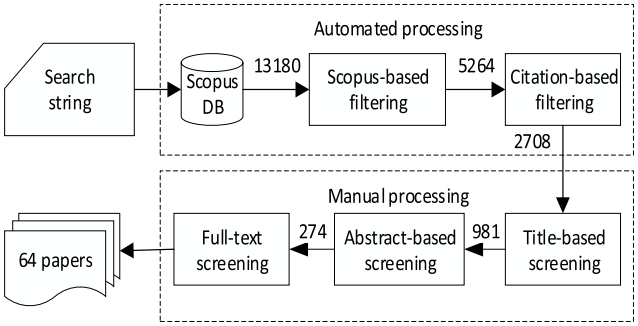

FIGURE 3. Scope of this study.

TABLE 8. Inclusion criteria applied by Scopus-based filtering.

Filter	Values
Research field	Computer Science Engineering Decision Sciences
Publication date	< = 2019
Document type	Conference paper [Journal/Magazine] Article
Language	English
Keywords	"Privacy" or "Data Protection" or "Privacy Preserving" or "Security And Privacy" or "Privacy Protection" or "Computer Privacy" or "Privacy Preservation" or "Privacy And Security" or "Location Privacy" or "Privacy Requirements" or "Privacy risks" or "Privacy analysis" or "Data Privacy"

TABLE 9. Minimum number of citations required–50th percentile in computer science.

Publishing year	Minimum citations
Before 2009	6
2010 or 2013	5
2014	4
2015 or 2016	3
2017	2
2018	1
2019	0

Figure 3 details the procedure that consisted of an automated processing followed by a manual processing. The automated processing involved two main tasks: Scopus-based filtering according to the inclusion criteria shown in Table 8, and citation-based filtering according to the year of publishing and the minimum number of citations listed in Table 9. After applying the Scopus-based filtering we obtained a pool of 5,264 candidate papers was obtained. We then filtered by the minimum number of citations a paper should have to fall above the 50th percentile of papers in the Computer Science category, according to the Thomson Reuters indicators [29]. Note that for papers published before 2009 and 2019, the minimum citations were kept at 6 and 0, respectively. At the end of this stage, a set of 2,708 papers passed to manual processing.

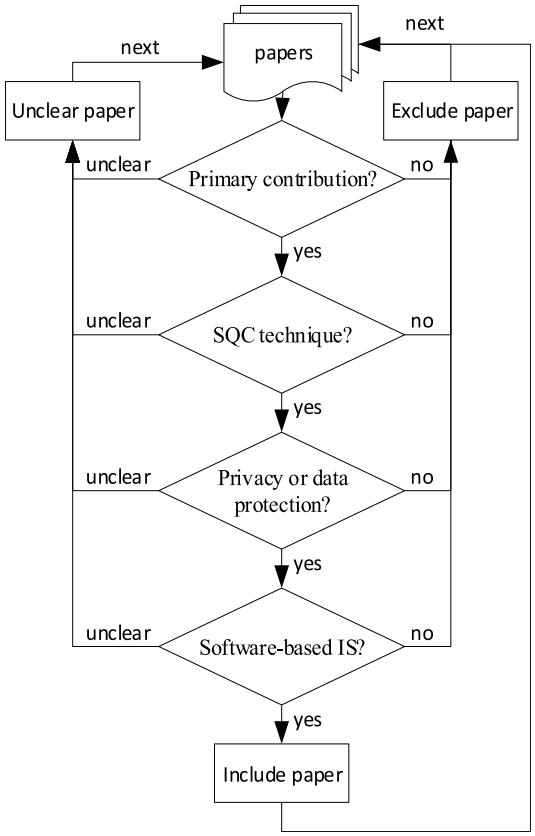

FIGURE 4. Criteria for inclusion.

The manual processing consisted of a screening based on titles, abstracts and full texts, with the goal of including papers reporting primary contributions that fall at the intersection of the three domains presented in Section A (*SQC techniques, software-based IS, and privacy and data protection*). With the support of the CADIMA tool [30], we have applied the inclusion criteria as a decision tree, as shown in Figure 4. Thus, we excluded papers that:

- [-] exclusively focused on presenting the results of evaluating one or more software-based IS by using state-of-the-art SQC techniques or tools, but the contribution is not an SQC technique itself (e.g. [31]–[34]).
- [-] focused only on anomaly mitigation techniques rather than anomaly detection (e.g. [35]–[37])
- [-] focused solely on the criteria through which a software system is evaluated, without describing an SQC technique (e.g. [38]–[40]).
- [-] the object under evaluation is not a software-based IS, but datasets in isolation (e.g. [41]–[44]), privacy policies (e.g. [45]–[47]), business processes (e.g. [48]–[50]), network protocols (e.g. [51]–[53]), or cryptography- and obfuscation-based protocols (e.g. [54]–[57]).
- [-] exclusively reports an SQC technique that is only applicable to one specific software-based IS and which can hardly be reused for another software based IS (e.g. [58], [59])
- [-] exclusively focused on privacy risk assessment to elicit the requirements of a privacy-friendly software-based IS (e.g. [60], [61]) or to quantify a privacy risk (e.g. [62]–[64]), but not to detect privacy-related anomalies.
- focused solely on SQC techniques for evaluating security-related anomalies or malware, but privacy or data protection are barely mentioned in the abstract or introduction sections (e.g. [65]–[68]). We did include papers reporting on general SQC techniques that, at least, provide an experiment or discussion related to privacy or data protection.

These exclusion criteria were added as ''clues'' using the CADIMA options, so that they were visible to the screeners.

Three possible values were used to label each paper: *included*, when all inclusion criteria and no exclusion criteria were met; *excluded*, when the paper met some exclusion criterion or did not meet an inclusion criterion, and; *unclear*, when the screener had some doubts. Prior to the main screening, a team of three researchers (screeners) conducted a four-iteration pilot in 92 papers (23 in each iteration), aimed at normalizing their criteria for including and excluding papers. After each iteration, divergences in papers labeled were discussed and agreed upon. After obtaining a 91% success rate and a Krippendorff's alpha inter-coder reliability coefficient of 0.748, we then moved to the main screening stage.

The main screening consisted of three stages, i.e. screening based on titles, abstracts and full texts. In each stage, the screeners worked individually on different sets of papers that CADIMA provides automatically. However, in order to ensure the inter-coder reliability, in the first two stages (screening based on title and abstract), 10% of the papers were reviewed by at least two screeners, while in the full-text screening all the papers were reviewed by two screeners. Divergences in labeled papers were discussed and agreed upon by the team.

In the title-based screening, after screening 2,708 papers, a pool of 981 candidate papers was labeled as *included* or *unclear* and passed to the next stage. Similarly, in the abstract-based screening, 274 candidate papers were labeled as *included* or *unclear* and moved to the final stage. In the full-text screening, screeners carried out a mandatory and an adaptive depth reading, i.e. the title, abstract, introduction and conclusions were mandatory, but if this information was not enough, the screeners read the section explaining the SQC technique. As already mentioned, at this stage all 274 papers were reviewed by two screeners, discussing and agreeing when divergences appeared. This process led to the selection of 64 papers for this study, whose bibliographic data are shown in the Appendix.

D. CLASSIFICATION SCHEME

The classification scheme was built using existing recognized classifications and then refining them iteratively.

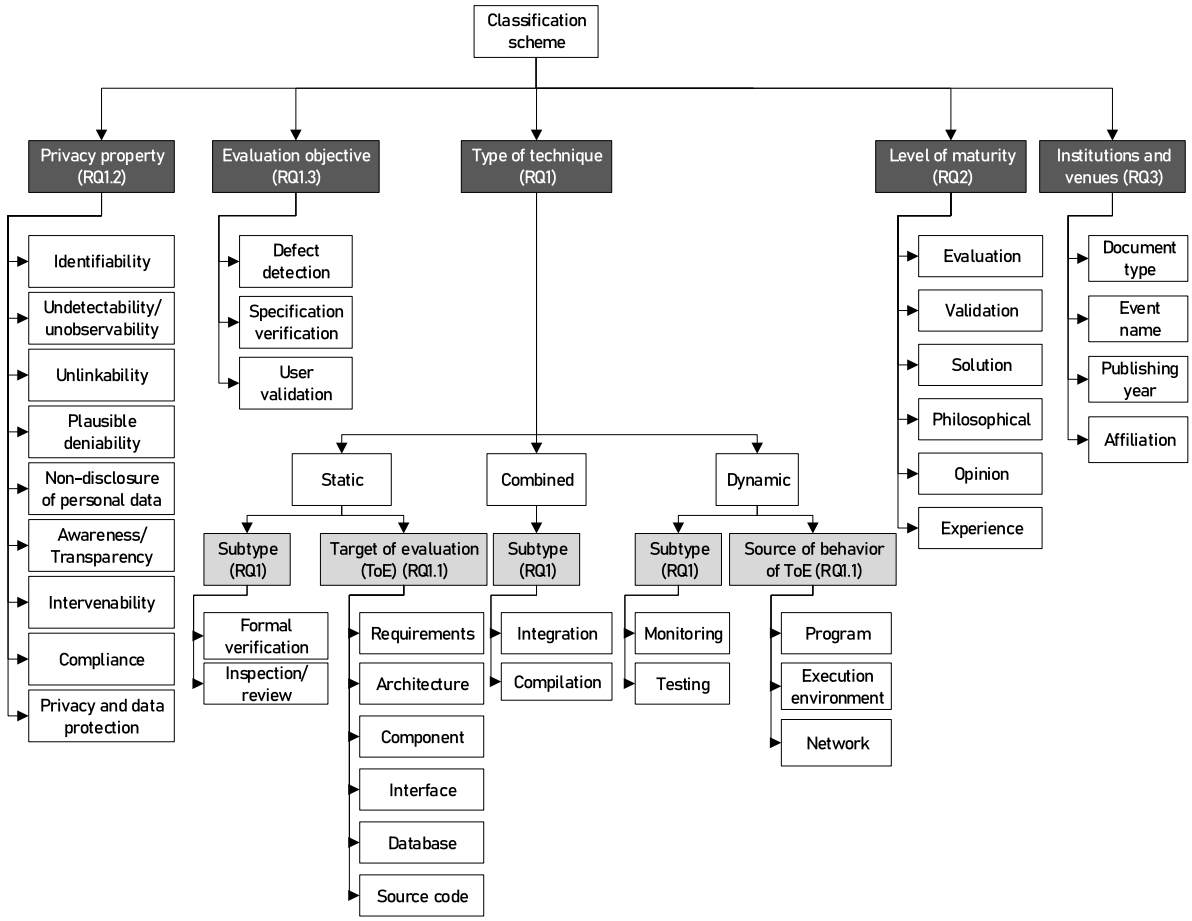

FIGURE 5. Classification scheme.

As suggested in the Petersen guidelines [22], we defined an initial scheme based on existing taxonomies and classifications widely recognized in the research community in order to support comparability. This scheme then evolved iteratively either by merging or adding new categories (e.g. combined SQC techniques) or by breaking down categories into subcategories (e.g. dynamic techniques into testing and monitoring). For this process, we took advantage of the full-text screening and a three-iteration pilot to classify 21 papers (7 papers per iteration) using an online form in Google Docs. Each paper was coded by the three researchers and, at the end of each iteration, divergences were discussed in detail, refining the classification scheme if needed.

Figure 5 shows the final and stable scheme used to classify the 64 papers consistently. Each paper was coded by assigning one or more attributes (white boxes) to each of the five dimensions (dark grey boxes) and other further dimensions depending on the type of SQC technique (light grey boxes). The dimensions and attributes shown in Figure 5 allowed us to answer RQ1, RQ2, and RQ3 as marked in brackets. The attributes relative to RQ1 and RQ2 were further elaborated in Section II. As for RQ3's attributes, they were obtained from the papers' bibliographic data to analyze the active countries or regions, active researchers' institutions (affiliations), and main venues at which papers are targeted in this field of research (type of document, i.e. journal or conference, and event name).

E. CLASSIFICATION PROCEDURE

A team of three researchers (coders) carried out the classification of the 64 papers included, using the scheme presented in Section D. For each paper, the attributes of dimension *Institutions and venues (RQ3)* were automatically extracted from the structured bibliographic data provided by Scopus, while the attributes of the other dimensions were manually coded by at least two researchers using an online form in Google Docs. The coders carried out a mandatory and an adaptive depth reading approach, i.e. the title, abstract, introduction and conclusions were mandatory before coding, but if this information was not enough, the coders reviewed the section explaining the SQC technique. The codification was carried out in subsets of five papers (the last of four papers), discussing and agreeing when divergences appeared and then moving on to the next subset of papers. Full details of the classification procedure are described in

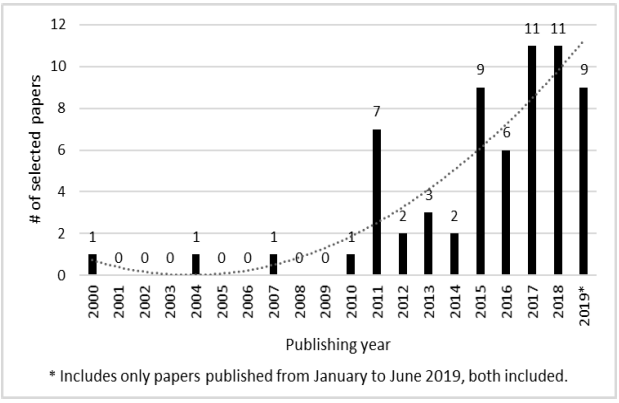

FIGURE 6. Distribution of selected papers by publishing year.

the codebook, which can be found in the replication package at http://dx.doi.org/10.17632/zvp3986f5b.1

IV. RESULTS

This section presents the results of the SMS to answer the three research questions. The findings were derived from explicit observations and trends in the collective results of the research team rather than individual researcher interpretations. Figure 6 shows the distribution of the papers, which have been published throughout the last 20 years. The number of publications has increased continuously since 2010. This has increased even more in the last five years (from 2015 to 2019), which concentrates more than two thirds of the total number of published articles (46 out of 64) despite papers from the second half of 2019 not being considered, as the study started in July 2019.

In the following subsections we respond to the research questions: Section A presents the results regarding the SQC techniques for detecting privacy-related anomalies, reported in the literature (RQ1), findings on the level of maturity of reported SQC techniques (RQ2) are presented in Section B, and finally Section C shows the institutions and venues for publishing in the domain (RQ3). Throughout this section we refer to the papers studied by the number ID used in the replication package in order make traceability easier.

A. RQ1. WHAT SQC TECHNIQUES HAVE BEEN PROPOSED FOR FINDING PRIVACY-RELATED ANOMALIES THROUGHOUT THE SOFTWARE DEVELOPMENT PROCESS?

Figure 7 provides a summary of the *types* and *subtypes* of SQC techniques that have been reported in our pool of 64 papers, according to the categories discussed in Section III-D. We can see in Figure 7 that the research efforts for finding privacy-related anomalies in software-based IS are led by *static techniques* (53%), followed by *dynamic techniques* (30%), and *combined techniques* (17%). Note that when a paper reports a *combined technique*, it is necessarily made up of two individual SQC techniques working together: one *static technique* and one *dynamic technique*. As a result, the papers studied report a total of 75 different instances of *static* and *dynamic* techniques for assessing software-based IS in order to detect privacy-related anomalies: 34 falling into the *static technique* category and 19 into the *dynamic technique* (see Figure 7a), plus 11 *static techniques* and 11 *dynamic techniques* which were combined to be applied with a common goal (see Figure 7b). In the following sub-sections we elaborate on these results in more detail.

- Static techniques

As shown in Figure 7 a, the majority of research into *static techniques* fall into the *inspection and review* subtype (19 out of 34 instances), whereas the remaining 15 instances strive to detect anomalies through *formal verification* techniques. *Inspection and review* techniques follow two main approaches: *model-based analysis* on design artifacts (ID34, ID131, ID195, ID97, ID519, ID1067) and program static analysis on code artifacts (see the remaining *inspection/review* techniques in Figure 7a). The former rely on structural and behavioral design models that have been annotated with privacy-specific information (e.g. personal data category, purpose, recipient); they are then checked for compliance with customer preferences expressed as Privacy Level Agreements. The latter is performed on the object code, source code or configuration files of IS in order to detect potential disclosures of personal data.

As for *formal verification*, these techniques use two main approaches in order to prove that formal representations of privacy-friendly IS are compliant with privacy requirements: *theorem proving* and *model checking* [69]. *Formal verification techniques* that rely on *theorem proving* (ID3, ID17, ID21, ID25, ID31, ID1494, ID1684, ID1951, ID1992) are not necessarily an automated approach, but may require the involvement of an expert analyst to derive a conclusion iteratively (i.e., theorem) with the support of reasoning tools. Conversely, *formal verification* techniques that rely on *model checking* automatically prove a proposition (i.e. privacy requirement specification) through the brute-force exploration of all relevant states of the information system (see the remaining *formal verification* techniques in Figure 7a).

- Dynamic techniques

Research efforts on *dynamic techniques* are also significant (30%) and have been almost equally divided between *testing techniques* (9 out of 19) and *monitoring techniques* (10 out of 19). *Testing techniques* capture and analyze traces of three different sources of behavior, namely *execution environment*, *network traffic*, and the *instrumented programs*. Almost all of them focus on detecting disclosures of personal data.

Monitoring techniques are aimed mainly at demonstrating compliance with privacy policies by collecting well-structured logs during the operation of the software-based IS, although they are also used to detect privacy policy

violations and then take remedial actions at the development level (ID2100, ID2228, ID1195).

- Combined techniques

The *combined techniques* have also received attention from researchers, although still at a low rate (17%). Figure 7b shows that static and dynamic techniques were either

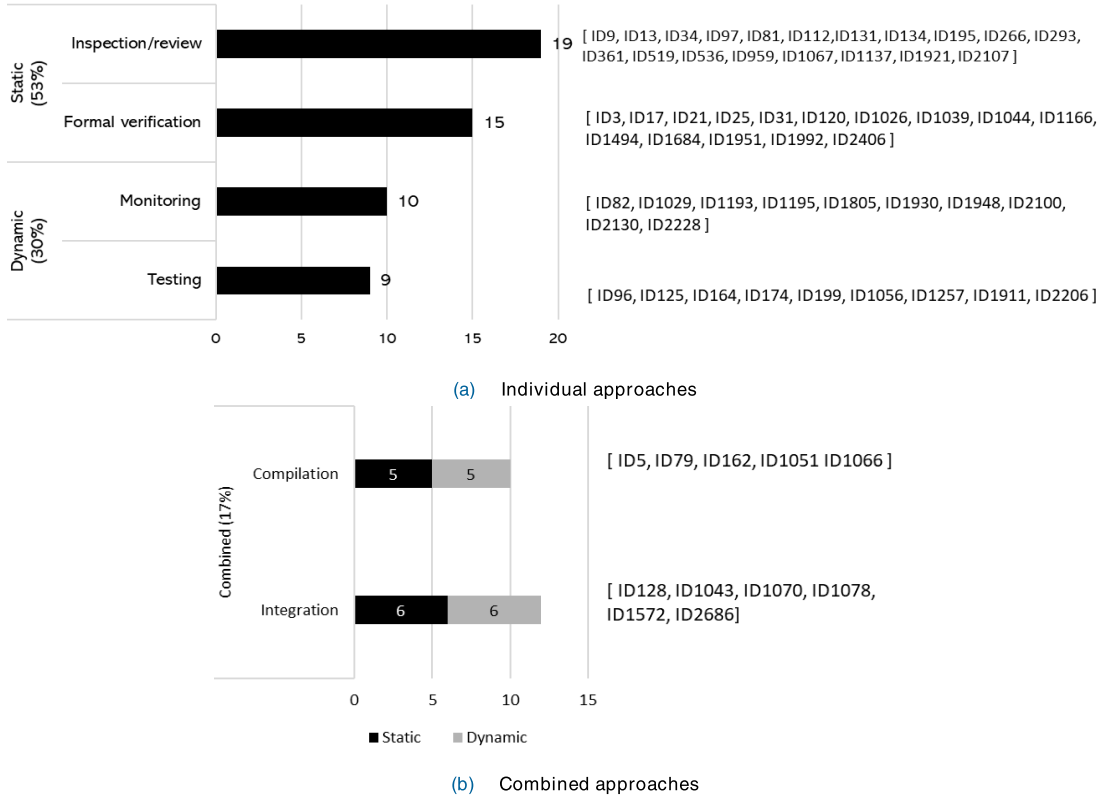

FIGURE 7. Software quality control techniques proposed in the studied papers.

compiled (i.e., each operates separately) or *integrated* (i.e., both operate complementarily by building an evaluation pipeline) in order to improve *coverage, precision or efficiency* of the detection of privacy-related anomalies.

Almost half of the efforts in *combined techniques* (5 out of 11) fall into the *compilation* subtype and pursue the *improvement of coverage* either by (1) covering a larger portion of the software-based IS being evaluated (ID79, ID1066) or (2) covering a broader set of privacy-related anomalies (ID5, ID162, ID1051, ID1066).

On the other hand, an equally significant effort in the *combined techniques* (6 out of 11) is framed within the *integration* subtype and pursues either (1) the *improvement in precision* by using a d*ynamic technique* to assert or eliminate potential privacy-related anomalies previously detected by a *static technique* (ID1070, ID1078, ID1572, ID2686), or (2) the *improvement in efficiency* in order to detect privacy-related anomalies in a shorter period of time (ID128, ID1043).

In the following subsections, the reported SQC techniques are set out in more detail by presenting their surrounding elements. Specifically, we present the software artefacts that are the specific target of evaluation of the techniques studied (Section 1), the particular objective which is pursued by the techniques (Section 2) and the privacy property and the reference criteria through which a software system is evaluated (Section 3). It is worth emphasizing that we do not report on these individual elements unless described in connection with an SQC technique.

1) RQ1.1. IN WHICH SOFTWARE ARTIFACTS ARE SQC TECHNIQUES APPLIED?

As explained in Section II, we have broken down the classification of the software artefacts depending on whether they are inputs for static or dynamic techniques. For the static techniques we have identified the readable representations (i.e., *software artefacts*) of the software-based IS from which the models to be analysed are extracted (see *Figure 8*). For dynamic techniques, we have identified three different checkpoints (i.e. *source of behaviour*) used for monitoring and logging the relevant execution traces, which are then analysed to detect privacy-related anomalies (see *Figure* 9). For both we distinguish the subtypes.

We can see in *Figure* 8 that almost two thirds (65%) of *static techniques* (24 *inspection/review* and 4 *formal verification*) take a type of *code artifact* as input in order to extract privacy relevant models. Three different kinds of *code artefacts* were used: *configuration files, source code*, and *object code*.

- Object code

A large majority of static techniques (4 *formal verifications* and 16 *inspections/reviews*) take the *object code* of Android

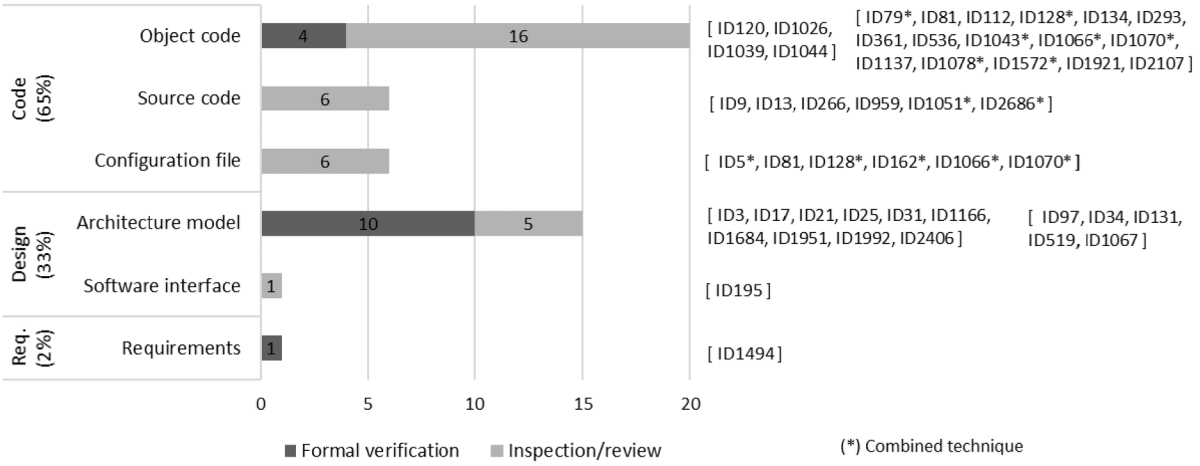

FIGURE 8. Target of evaluation of the static techniques.

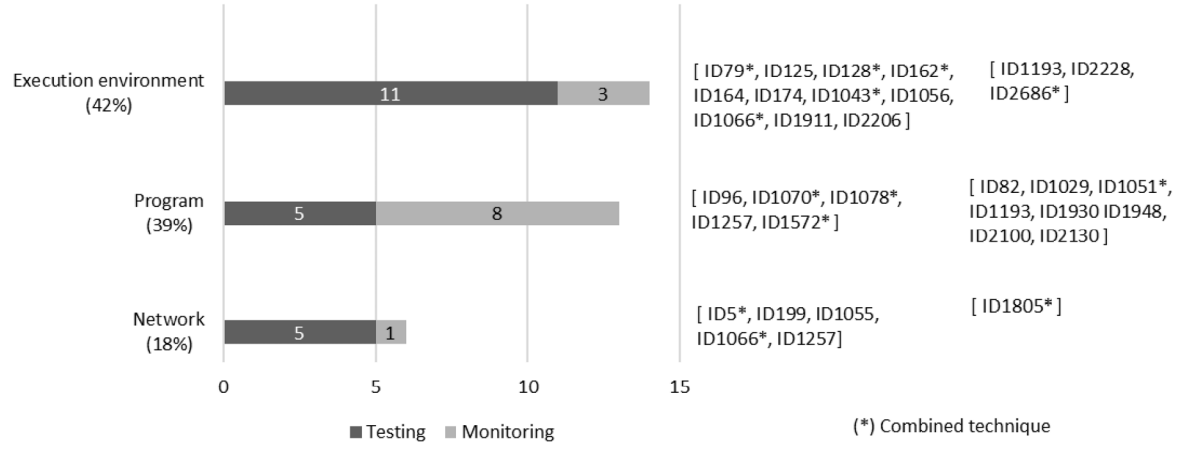

FIGURE 9. Source of behavior of dynamic techniques.

applications (i.e. Dalvik bytecode) as input for the analysis. Almost all of them decompile the *object code* into other intermediate representations before analysis in order to leverage generic frameworks already implemented for flow analysis, for instance, into *Smali* (e.g., ID79, ID134, ID1043, 1066), *Jimple* (e.g., ID536, ID1070), *Shimple* (e.g. ID81) or *Java bytecode* (e.g. ID120). These generic frameworks enable program models to be extracted from the object code by generating well-known data structures such as call graphs (CG) or control flow graphs (CFG). Privacy analysis is then carried out, for example, by propagating taint information through these data structures and carrying out a backward or forward tracing, *inter alia*, to determine the reachability between sensitive sources and sinks and hence detect potential disclosures of personal data.

- Source code

Six techniques require the use of *source code* as input. While several of them use the source code in a similar way to the aforementioned approach (ID9, ID13, ID1051), we distinguish here profile-based techniques that detect the potential disclosure of personal data after the statistical learning of a benign profile (1) by using static code metrics as predictors, including lines of code, number of functions, functional complexity, and another 18 metrics (ID266), and (2) by using the paths of typical flows of personal data between sensitive sources and sinks (ID2686). The other approach (ID959) carries out a syntactic analysis looking for dark code patterns (i.e. code smells) in the source code, for example, code statements accessing unique hardware identifiers or the use of dynamic permission with missing revocation statements, which may indicate the presence of some privacy anomalies.

- Configuration files

As for the *configuration files*, the *Manifest* and *Layout* files of Android applications were the target of evaluation of 6 techniques studied. The *Manifest* file encodes a declarative permission model that is significant for privacy assessment, particularly to detect whether an application is over-privileged when accessing sensitive personal data (i.e., request more permissions than necessary). Some techniques merely rely on the Android protection levels

to analyse the declaration of ''dangerous permissions'' in the *Manifest* (ID5), other techniques additionally manually check whether these dangerous permissions were declared in the applications' privacy policies (ID162), while others, once generated CG and CFG of the application, search for permission-protected API methods and then detect whether the permissions declared in the *Manifest* are actually being used (ID81, ID1066, ID1070). The *Manifest* also includes high-level architectural information which is leveraged by some techniques studied to ascertain all built-in components and hence improve the detection of the disclosure of personal data (ID81, ID1066).

Similarly, some techniques (ID128) leverage the *Layout* file, which defines the structure of the applications' user interfaces (UI), in order to retrieve clickable UI elements and therefore focus only on those tasks that users can interact with and that may trigger a personal data disclosure.

- Architecture model

In terms of *design artifacts, architecture models* have been used by a third of the SQC techniques studied (15 *formal verification* and 5 *inspection/review*). On the one hand, *formal verification* techniques take Unified Modeling Language (UML) models as input [70] which are then translated into formal models (e.g., ID1166) or directly take formal models that have been specified through available declarative frameworks (e.g., ID3, ID17, ID31). Most of the resulting formal models provide a description of the software-based IS architecture, focusing especially on the actors, the system components and their relationships, represented as personal data flows between them. Then, using one of the aforementioned formal verification approaches (i.e. model checking or theorem proving), one or more of the formal properties that an architecture should satisfy (e.g., purpose limitation or data minimization) are proven.

On the other hand, the *inspection/review* techniques take different structural and behavioral system models as input, including UML class, activity, component and deployment diagrams (ID34, ID97, ID131, ID219) and data flows diagrams (ID1067), which are manually annotated with convenient privacy elements (i.e. *purpose, visibility, granularity* and *retention period*). Privacy verification then relies on these annotated models to check whether or not customer privacy level agreements, specified in terms of the aforementioned elements, are met.

- User interfaces

Privacy-friendly *user interfaces* are another design artifact that has been minimally used as input for an SQC technique in order to assess transparency aspects (only the ID195 technique uses it). Similarly, only one of the techniques studied (ID1494) takes the *requirements specification* as input from different but dependent IS in order to detect potential data repurposing and over-collection.

- Execution environment

As shown in Figure 9, the *execution environment* is the primary checkpoint used by *dynamic techniques* to capture the relevant execution traces of the software-based IS being evaluated (11 *testing techniques* and 3 *monitoring techniques*). In some cases, the *execution environment* is instrumented in order to perform, for example, dynamic taint analysis (ID164, ID1043, ID1056, ID1911, ID2206, ID2228). In other cases, the *execution environment* is merely extended with available frameworks that allow certain sensitive methods invoked by the software system to be hooked (ID79, ID125, ID128, ID174). Each invocation, argument(s), return values and other relevant information are then logged for further analysis. Once execution traces are logged, they are analysed in order to detect violations of the least privilege principle (ID128), measure the user's awareness after a personal data disclosure (ID1056), while the remaining ones focused on detecting the disclosure of personal data. These approaches require the privileges necessary to modify the *execution environment*, whereas the software-based IS being evaluated remains unchanged.

- Program

The program is also largely used by *dynamic techniques* as a checkpoint to capture the execution traces. Most *monitoring techniques* (8 techniques) rely on pieces of code (i.e. monitor code) conveniently embedded by a software-based IS, with the aim of writing structured logs during its execution. These structured logs are then used by the eight techniques to check whether the software-based IS being evaluated complies with privacy policies or regulations. Only one *monitoring technique* (ID1029) relies on manual checking of stored logs, but the remaining seven techniques identified in *Figure* 9 focus primarily on automatic compliance checking through the use of reasoning logic approaches.

On the other hand, most of the *testing techniques* instrumentalize the software-based IS prior to its execution by adding monitor code around sensitive methods to log their invocation, while keeping the *execution environment* unchanged. Afterwards, logged information is analyzed in order to detect violations of the *least privilege* principle (ID1070) or personal data disclosures (ID1078, ID1257, ID1572). There is a key difference between *monitoring* and *testing techniques* regarding the use of the *program* as a source of behavior. In the former, the monitor code is embedded by the developers from the design of the software system, thus containing sufficient contextual information to check compliance. On the other hand, testing techniques instrument third-party programs, requiring its decompiling and placing the monitor code accordingly.

- Network traffic

A smaller number of *dynamic techniques* use *network traffic* as a source of behavior (5 *testing techniques* and 1 *monitoring technique*) in order to detect the disclosure of personal data of the software-based IS being evaluated. Three of them deploy a proxy between the software-based IS and the remote counterpart servers to carry out a man-in-the-middle attack and then intercept network packets (ID5, ID199, ID1257, ID1805), while the remaining one (ID1066) takes advantage

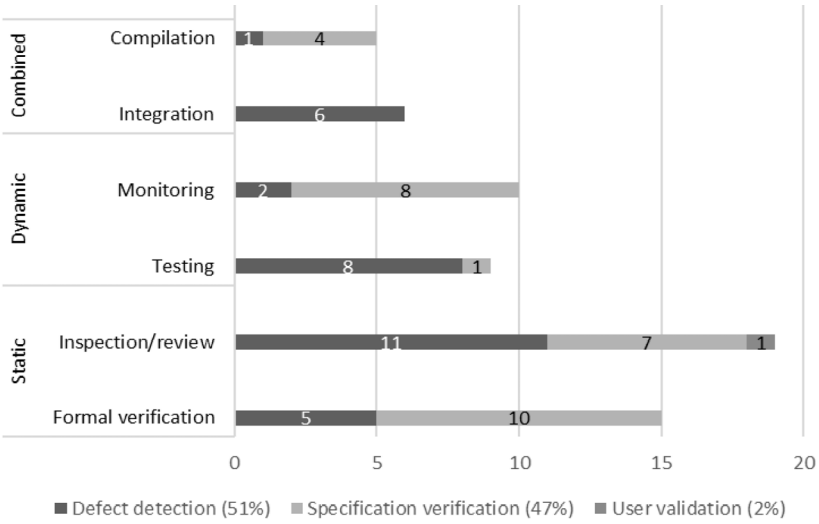

FIGURE 10. Distribution of the objectives pursued by the reported SQC techniques.

TABLE 10. Goals pursued by the reported SQC techniques.

Technique type	Subtype	Defect detection	Specification verification	User validation
Static	Formal verification	ID31, ID120, ID1026, ID1039, ID1044 **(5)**	ID3, ID17, ID21, ID25, ID1166, ID1494, ID1684, ID1951, ID1992, ID2406 **(10)**	**(0)**
	Inspection/review	ID112, ID134, ID266, ID293, ID361, ID536, ID959, ID1067, ID1137, ID1921, ID2107 **(11)**	ID9, ID13, ID34, ID81, ID97, ID131, ID519 **(7)**	ID195 **(1)**
Dynamic	Testing	ID96, ID125, ID164, ID174, ID1056, ID1257, ID1911, ID2206 **(8)**	ID199 **(1)**	**(0)**
	Monitoring	ID1195 , ID2228 **(2)**	ID82, ID1029, ID1193, ID1805, ID1930, ID1948, ID2100, ID2130 **(8)**	**(0)**
Combined	Integration	ID128, ID1043, ID1070, ID1078, ID1572, ID2686 **(6)**	**(0)**	**(0)**
	Compilation	ID1066 **(1)**	ID5, ID79, ID162, ID1051 **(4)**	**(0)**

of a built-in feature of the emulator used to capture network traffic. Once network traffic is captured, it is analyzed by searching for keywords, phrases or regular expressions relative to personal data (ID5, ID1066), calculating similarity rather than precise string matching (ID1257), or building a catalogue of key-value pairs related to personal data, which is then used to detect personal data disclosure looking for either key or value (ID199). Moreover, a monitoring technique (ID1805) aims at analyzing outbound network traffic searching for fingerprints of the personal data rather than personal data itself, thus preserving the confidentiality of the owner's personal data in the production environment.

2) RQ1.2. WHICH PARTICULAR OBJECTIVE IS PURSUED BY REPORTED SQC TECHNIQUES?

We classified the SQC techniques according to the ultimate objective they pursued, i.e. *defect detection*, *specification verification*, and *user validation*. Figure 10 and Table 10 shows a summary of these results.

- Defect detection

Almost all of the research efforts are equally divided between techniques that pursue *defect detection* (51%) and *specification verification* (47%), while only one technique (2%) pursues user validation. The techniques aimed at defect detection are primary carried out at the code level. Accordingly, as shown in *Figure 10, the integration, testing*, and *program static analysis (a form of inspection/review* as for [1])) are mainly used for this purpose. As explained further in Section III-A-3, almost all these SQC techniques focus on detecting the potential disclosure of personal data on the basis of an ad-hoc set of predefined rules at the code level, i.e., connections between sensitive method-level sources and sinks. Only two techniques pursue *defect detection* at the architectural level: one of them (ID1067) matches the flows

TABLE 11. Privacy property addressed by the reported SQC techniques.

Privacy property	Static		Dynamic		Combined	%
	Formal verification	Inspection/ Review	Testing	Monitoring		
Identifiability (0)	**(0)**	**(0)**	**(0)**	**(0)**	**(0)**	0%
Undetectability /unobservability (0)	**(0)**	**(0)**	**(0)**	**(0)**	**(0)**	0%
Unlinkability (0)	**(0)**	**(0)**	**(0)**	**(0)**	**(0)**	0%
Plausible deniability (0)	**(0)**	**(0)**	**(0)**	**(0)**	**(0)**	0%
Non-disclosure of personal data (31)	ID31, ID120, ID1026, ID1039, ID1044 **(5)**	ID112, ID134, ID266, ID293, ID361, ID536, ID959, ID1137, ID1921, ID2107 **(10)**	ID125, ID164, ID174, ID1257, ID1911, ID2206 **(6)**	ID1195, ID1805, ID2228 **(3)**	ID128, ID1043, ID1066, ID1070, ID1078, ID1572, ID2686 **(7)**	48%
Awareness/Transparency (3)	**(0)**	ID195 **(1)**	ID96, ID1056 **(2)**	**(0)**	**(0)**	5%
Intervenability (0)	**(0)**	**(0)**	**(0)**	**(0)**	**(0)**	0%
Compliance (29)	ID3, ID17, ID21, ID25, ID1166, ID1494, ID1684, ID1951, ID1992, ID2406 **(10)**	ID9, ID13, ID34, ID81, ID97, ID131, ID519 **(7)**	ID199 **(1)**	ID82, ID1029, ID1193, ID1930, ID1948, ID2100 ID2130 **(7)**	ID5, ID79, ID162, ID1051 **(4)**	45%
Privacy and data protection (1)	**(0)**	ID1067 **(1)**	**(0)**	**(0)**	**(0)**	2%

of personal data, modelled as DFD (Data Flow Diagrams), against a set of threat patterns, while the other (ID31) models attack scenarios and prove whether the data minimization properties of a software architecture have been met.

- Specification verification

The *specification verification* is also pursued by a significant number of the techniques studied (47%). They have been applied primarily at the design stage through *formal specification and model-based analysis (a form of inspection/review* as for [1])) and at the maintenance stage through *monitoring techniques*. In contrast to techniques pursuing *defect detection*, these techniques check whether the IS being evaluated complies (or not) with a predefined set of (non-) permitted flows of personal data that have been specified in either the privacy policies, privacy level agreements or requirements specifications. Thirty-seven percent of the SQC techniques (11 out of 30) aimed at specification verification derive their compliance criteria from the GDPR (ID3, ID5, ID9, ID13, ID21, ID25, ID34, ID131, ID162, ID199, ID519). The other three derive from EU-DPD, the pre-GDPR European regulatory framework (ID17, ID97, ID1029), others from HIPAA (ID1948, ID1193, ID2130), while the remaining are of a generic nature. Finally, some papers used the term *privacy policy* to refer to an ad-hoc code-level set of (non-) allowed flows of personal data between sensitive sources and sinks (c.f. ID1921); they were categorised as *defect detection*.

- User validation

User validation is only pursued by a user-centered testing technique (ID195), which provides users with different design alternatives in order to evaluate aspects related to transparency/awareness and then receiving feedback from them. Note that we did study some techniques that assess whether a software-based IS meets the customer preferences specified through PLAs (ID34, ID97, ID131, ID1805, ID2100); however, these have not been categorized as *user validation*, as there is not really any user/customer involvement or feedback from them.

3) RQ1.3. WHICH PRIVACY PROPERTIES ARE EVALUATED BY THE SQC TECHNIQUES IDENTIFIED?

The 64 papers studied have primarily focused on two privacy properties: *(non-)disclosure of personal information* and *compliance*. These privacy properties are targeted by 93% of the SQC techniques, 48% and 45% respectively, while a small 5% targeted *awareness/ transparency*. On the other hand, there were no appearances of SQC techniques assessing several privacy properties: *identifiability, undetectability, unlinkability, plausible deniability*, and *intervenability*. Table 11 shows the distribution of the privacy properties, arranged according to the SQC technique that addresses them.

- Non-disclosure of personal data

Anomalies relative to the *(non-)disclosure of personal data* are evaluated using different techniques, the *inspection/review (10), testing (6)* and *combined (7)* being the techniques mostly used for this purpose. As explained in Section II-D, the *disclosure of personal data* implies that personal data are exposed to individuals or providers who are not supposed to have access to it. Different criteria or

TABLE 12. Classification of the SQC techniques according to the maturity level.

Maturity level	Static		Dynamic		Combined	%
	Formal verification	Inspection/ review	Testing	Monitoring		
Evaluation research (10)	**(0)**	ID34 **(1)**	ID125, ID174, ID2206 **(3)**	ID2228 **(1)**	ID79, ID128, ID1051, ID1078, ID1043 **(5)**	15.6%
Validation research (30)	ID120, ID1494, ID1039 **(3)**	ID13, ID81, ID112, ID131, ID134, ID266, ID293, ID361, ID519, ID536, ID959, ID1137 **(12)**	ID96, ID164, ID199, ID1056, ID1257 **(5)**	ID82, ID1029, ID1193, ID1195, ID1948 **(5)**	ID5, ID162, ID1066, ID1070, ID2686 **(5)**	46.8%
Solution proposal (22)	ID3, ID17, ID21, ID25, ID31, ID1026, ID1044, ID1166, ID1684, ID1992, ID2406 **(11)**	ID9, ID97, ID195, ID1067, ID1921, ID2107 **(6)**	**(0)**	ID1805,ID2100, ID1930, ID2130 **(4)**	ID1572 **(1)**	34.4%
Philosophical paper (1)	ID1951 **(1)**	**(0)**	**(0)**	**(0)**	**(0)**	1.6%
Opinion paper (1)	**(0)**	**(0)**	ID1911 **(1)**	**(0)**	**(0)**	1.6%

conditions have been used to determine anomalies relative to this property. Thus, the majority of techniques warn of the *disclosure of personal data* when they detect that personal data merely leaves the IS being evaluated, assuming that anyone could access them (ID120, ID134, ID164, ID266, ID293, ID361, ID536, ID1026, ID1039, ID1043, ID1044, ID1078, ID1137, ID1195, ID1257, ID1572, ID1911, ID1921, ID2107, ID2206, ID2228, ID2686). Two of them (ID127, ID174) focus on detecting *who* is accessing the personal data (i.e., third-party libraries or the IS itself), and then alert to *disclosures of personal data* only when third-parties do so.

The other technique (ID128) focuses on discriminating whether access to certain personal data is required for a core functionality of the IS or for another secondary (third-party) task; when the latter occurs, a potential *disclosure of personal data* is alerted. Finally, other techniques warn of potential future disclosures by detecting whether an IS has been granted unnecessary authorization/permissions to access personal data, i.e. it is *over-privileged* (ID112, ID128, ID959, ID1066, ID1070).

- Compliance

Different conditions have been used in order to detect and then warn of anomalies relative to the property *compliance*, which is targeted by a significant number of SQC techniques (30 out of 64). Some of them leverage the detection of *disclosure of personal data*, in a similar way as that explained in the previous paragraph, but they also check that the access or share of certain *categories of personal data* have been declared in their textual privacy policies (ID79, ID81, ID162, ID199). Others emphasize the convenient aggregation of the results of detecting *disclosures of personal data* to represent them at different levels of abstraction, i.e. from developer level to data protection officer level, and then check that the *categories of personal data* have been declared in their formal privacy policies (ID9, ID13). Yet other techniques go beyond just considering the *category of personal data* to warn of a potential anomaly relative to *compliance*, but the criteria or conditions also consider the *entity accessing or sending* a certain category of personal data, the *entity receiving* the personal data, or the *purpose or role* of these entities (ID3, ID17, ID25, ID82, ID1029, ID1051, ID1193, ID1494, ID1684) and, in some cases, whether they have the *consent* to do so (ID21, ID1948, ID2130, ID5).

In the same vein, some techniques warn of anomalies relative to *compliance*, when IS architecture models deviate from four privacy checks (i.e., *purpose, visibility, granularity*, and *retention*) specified by customers through a PLA (ID34, ID97, ID131, ID519, ID2100). The remaining techniques also pointed out that they address the property *compliance*, although they have not defined concrete criteria or conditions but general (ad-hoc) privacy requirements (e.g. ID1166, ID1930, ID1951, ID2406).

- Awareness/transparency

A small number of techniques studied (3) targeted the property *awareness/transparency*. Two of them (ID96, ID1056) assess whether the IS interfaces provide or display concrete functionality or information (e.g., offer forms of intervention on data collection), while the other (ID1056) checks personal data disclosure and measures the user's awareness as a timing issue (i.e., how long it takes before the user becomes aware of a data disclosure).

B. RQ2. WHAT ARE THE TYPES OF RESEARCH CONDUCTED BY THE STUDIES IDENTIFIED?

Among the 64 papers, five types of research were found: *evaluation research, validation research, solution proposal, philosophical papers and opinion papers*. As shown in Table 12, most of the instances of SQC techniques (62.4%)

have been empirically proven, while the remaining 37.6% have not. On the one hand, among the 40 empirically proven techniques, only 10 instances have been *evaluated* in a real-world context, while most of them (30 techniques) have been merely *validated*. On the other hand, among the non-empirically proven techniques, almost all papers (22 out of 24) are *solution proposals*, and of the two remaining ones, one is a philosophical paper and the other an opinion paper. Note also that 96.8% of contributions (62 out of 64) are at least a *solution proposal* or higher. Accordingly, these results clearly suggest that contributions on SQC techniques for evaluating privacy aspects tend to be specific rather than abstract proposals or only general insights. Thus, the majority of papers tend to assess the validity of their contributions at least by means of an example to illustrate the applicability of the solution.

Considering the diversity of the techniques studied, we analysed the trend of the maturity levels by clustering the contributions by the subtype of SQC technique. See Table 12 and Figure 11. The results showed that, on the one hand, formal verification techniques are the least mature, since almost all contributions (12 out of 15) have not been empirically proven, and only the remaining three instances have been *validated* (but none *evaluated* in a real context), as shown in Figure 11 (a). On the other hand, as shown in Figure 11 (e), the combined techniques emerge as the most mature approaches since the majority of their instances have been empirically proven (10 of the 11) and, in addition, they have the highest number of instances empirically *evaluated* in a real-world context (5 out of 10 at this maturity level), four of them in the last five years.

In the middle of the two aforementioned techniques, and arranged from a higher to a lower maturity level of their instances, we found the *testing* (8 out of 9 empirically proven), *inspection/review* (13 out of 19) and *monitoring techniques* (6 out of 10). Although admittedly only a few instances of these three SQC techniques were evaluated in a real-world context, most of these contributions tend to be proven empirically (27 out of 38).

C. RQ3. WHO ARE THE INSTITUTIONS THAT CONTRIBUTE MOST IN THIS AREA AND WHAT ARE THE VENUES FOR PUBLISHING?

As for the institutions that have contributed to the state of the art of SQC techniques focusing on privacy aspects, a total of 95 institutions from 25 different countries have published 64 studied papers. We analysed these results in terms of the institutions (and their countries) with the largest number of contributions in this domain, also pointing out the most active in the last five-year period. On the one hand, a group of 14 institutions have published two or more contributions as shown in Figure 12. Nearly all of them could be considered the most potentially active in the research domain, as they have published more than one paper and at least one of them in the last five years (from 2015 to 2019). However, we also highlight the appearance of new research institutions, which entered the research domain with one publication in the last five years. For instance, out of a total of 18 contributing institutions that published in 2019 (see Figure 13), 14 of them are newcomers to the domain. Similarly, of a total of 20 institutions in 2018, 13 of them are newcomers. These results suggest a growing and sustained concern about the domain of study by new research institutions mainly in the last 5 years.

On the other hand, from a geographical perspective, the top three countries of the organization researching in the domain belongs to the United States (24 institutions), followed by Germany (16) and China (9). Furthermore, it is worth noting that almost half of all contributing institutions (45%) are located in the European Union (EU), which can be explained by the fact that a significant number of the reported SQC techniques are mainly used for privacy compliance, focusing on the GDPR principles as shown in the results of RQ1.2 (Section IV).

With respect to the venues for publishing contributions on SQC techniques focusing on privacy aspects, there is a great diversity in terms of the channels used and the venue domains. On the one hand, four channels (journals, conferences, symposiums, and workshops) and 44 specific venues (10 journals, 28 conferences, 4 symposiums, and 2 workshops) were identified. 80% of the papers (51 papers) were presented through conferences (41 papers), symposiums (8 papers), and workshops (2 papers), while the remaining 20% (13 papers) were published in journals.

Table 13 presents the eight conferences with more than one publication and all the journals in which the corresponding papers were published (the remaining 39 conferences with only one publication can be found in the replication package). These results show there are a large number of venues and four different channels in which researchers can publish their contributions on SQC techniques focusing on privacy aspects. Furthermore, the *IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Trustcom* stands out as the venue where the largest number of papers has been published, all of them in the last three years.

V. DISCUSSION

The evaluation of software-based IS in order to detect privacy related anomalies seems to be a promising area of research. Having found the first work published in 2000, we have identified a growing interest especially in the last five years in which we found more than two thirds (72%) of papers published. This is also evidenced by the sustained growth in the number of institutions researching and publishing in the field (Figure 13).

Some evidence suggests that this growing interest might have been stimulated by the appearance of the European General Data Protection Regulation (GDPR). First, the initial GDPR proposal was released in 2012, roughly aligned with the stepped increase in publications. Second, evaluation activities are highlighted in many passages of the GDPR, for example Article 42 establishes the need for mechanisms to

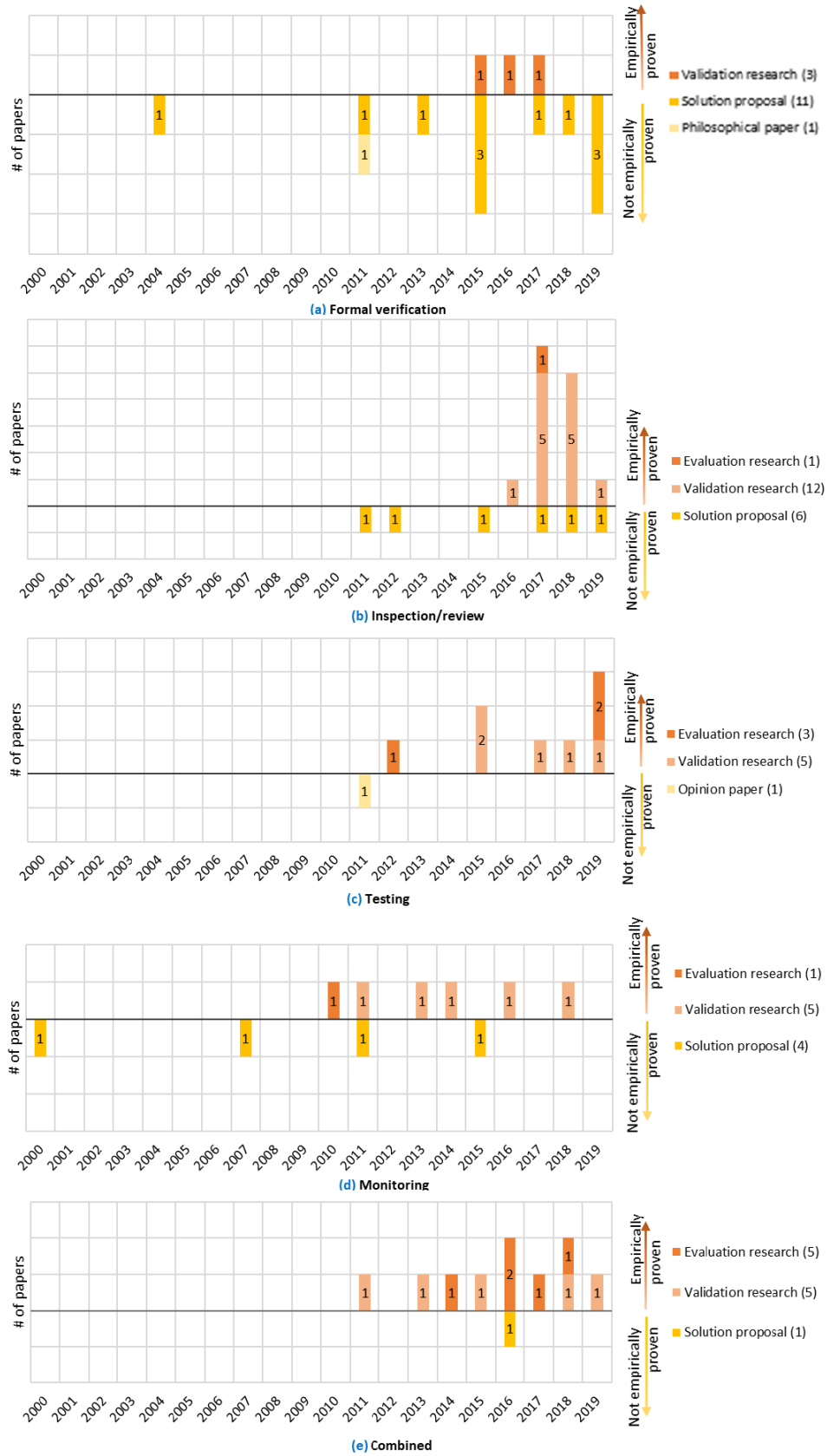

FIGURE 11. Distribution of the number of published papers by the SQC technique and its maturity level.

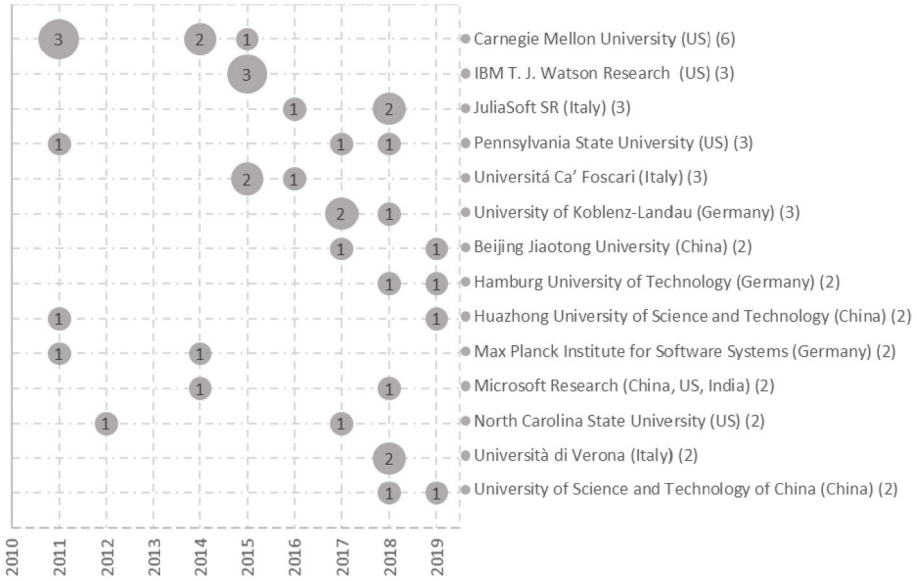

FIGURE 12. Institutions with the highest number of papers published in the domain (two or more).

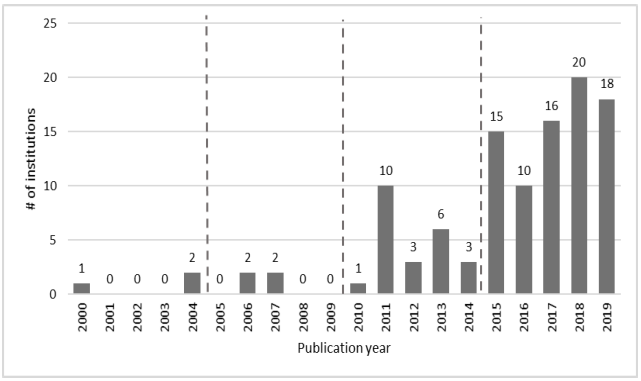

FIGURE 13. Distribution of the number of institutions per publishing year.

TABLE 13. Papers by top conference and journals.

Venue	# Papers	Type
IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Trustcom	4	Conference
Annual ACM Symposium on Applied Computing, SAC	2	Symposium
Annual Privacy Forum, APF	2	Conference
IEEE International Conference on Cloud Computing Technology and Science, CloudCom	2	Conference
ACM Conference on Data and Application Security and Privacy, CODASPY	2	Conference
IEEE International Conference on Computer and Communications, ICCC	2	Conference
Computers and Security	2	Journal
Mobile Information Systems	2	Journal
IEEE Access	1	Journal
Journal of Information Security and Applications	1	Journal
Journal of Web Semantics	1	Journal
Neurocomputing	1	Journal
IEEE Transactions on Information Forensics and Security	1	Journal
Transactions on Data Privacy	1	Journal
IEEE Transactions on Mobile Computing	1	Journal
IEEE Transactions on Software Engineering	1	Journal

evaluate and demonstrate the compliance of the developed IS. Indeed, we found that 37% of SQC techniques targeting specification verification derive their reference criteria from the GDPR legal framework, highlighting its incidence. Finally, almost half of all contributing research institutions (45%) are located in the European Union (EU). All of this evidence is aligned with the findings of a previous work [25] that highlights the greatest impact of GDPR although in terms of the amount of research work into privacy by design activities.

The techniques found are of a fairly high level of maturity, at least when compared with other subfields of privacy engineering [25]. Nearly all techniques (96.8%) fall into concrete contributions, i.e. from solution proposal upward. However, the level of maturity distinguishes between techniques applied in the early (“pre-code”) and later (“post-code”) phases of the software development process. On the one hand, techniques falling into post-code phases, i.e. static program analysis, testing and monitoring are the most mature techniques. This trend can be explained because they mostly rely on frameworks for information flow analysis already available in the security domain. Current research efforts

focus on overcoming difficulties found in their application in real-world contexts due, for example, to the use of obfuscation mechanisms, native code, dynamically loaded code, and reflection [11]. On the other hand, techniques falling into pre-code phases, such as formal verification and model analysis, are less mature. The maturity of formal techniques is aligned with results in other domains [17] where they have "*a limited impact on practical software development [as they involve] an expensive and complex procedure*". Nevertheless, we expect a sustained growth in these pre-code techniques as new methods for addressing privacy issues in the early stages of the development process (Privacy by Design) come about.

A large number of techniques have targeted mobile applications. Apps run on the user (data subject) side and therefore involve higher privacy risks due to their ubiquity, the vast quantity of the personal data they handle and their sensitive nature, as well as the amount of service providers involved in the processing (c.f. [71]). Android is the dominant platform in the market, open in terms of its source code, APIs (Application Programming Interface), and ease of access with maximum privileges to the whole operating system. Accordingly, research efforts have focused primarily on Android applications and research results on other mobile platforms such as iOS are scarce. However, Zang *et al.* revealed that while the average Android app sends potentially personal data to 3.1 third-party domains, the average iOS applications also disclose personal data to 2.6 third-party domains [72], highlighting the need to diversify research efforts. State-of-the-art techniques focusing on the Android OS could serve as a basis for adapting and applying them to the other Android-based IS e.g. Android Auto or Wear OS. However, further efforts are required to port them to evaluate iOS applications as the available techniques are currently highly dependent on Android specificities.

Modern privacy paradigms, such as contextual integrity [10], are starting to be considered, particularly in recipient-side IS. Traditional privacy paradigms such as *privacy as confidentiality* [9] rely on the binary criterion that any exposure of personal data outside the IS leads to a privacy violation. However, this binary criterion is insufficient when certain flows of personal data are, for example, expected and authorised by the data subject. In such cases, the detection of a privacy-related anomaly rather than being absolute requires more contextual information to be considered. *Privacy as contextual integrity* considers five key elements to define the privacy context: (1) the *type of personal data*; (2) the *data subject* to whom the personal data refers; (3) the *sender* and (4) the *receiver* of personal data acting in a particular *capacity* or *role*; and, (5) the transmission principles that constrain a flow (e.g. consent of the data subject). We have found only three works considering the sender of the personal data in user-side IS, and the receiver or the alignment with *transmission principles* is either overlooked or barely mentioned. On the other hand, context-aware techniques are most visible in the recipient-side IS where (un-)allowed flows of personal data are defined in terms of the *sender*, the *receiver*, or the *role* of these entities (ID3, ID17, ID25, ID82, ID1029, ID1051, ID1193, ID1494, ID1684), and the *transmission principle* (ID21, ID1948, ID2130, ID5). These context-aware SQC techniques are based on annotating the source code and logging program traces, which require a privacy-by-design mindset to consider these elements from the onset of the project.

All in all, the majority of these context-aware SQC techniques focus on establishing that personal data flows are aligned with a given privacy policy, and assume that this policy complies with a certain regulation, e.g. GDPR. Unfortunately, this is not always the case, particularly for mobile applications [73]. As a result, these techniques are not assessing any privacy-related anomaly but just compliance with their own privacy policies. We therefore strongly emphasise that these techniques should carry out more rigorous assessments of both privacy policy and regulatory compliance. We argue that an evaluation pipeline should be built involving interdisciplinary knowledge, including evaluation criteria supported by legal and not just technical interpretations. Some research explicitly highlights the need to incorporate interdisciplinary knowledge into the evaluation pipeline (ID17, ID1051), while other research includes a preliminary interdisciplinary analysis (ID21, ID25).

Finally, the combination of different SQC techniques appears to be a promising evolution path to improve precision, efficiency and coverage ([11], [13], [14]). Their adoption in the privacy realm still remains low (17%) when compared with other SQC techniques, although aligned with their adoption in other fields e.g. security (15%, as for [11]). Despite this small rate, almost half of these techniques (5 out of 11) have been used and evaluated in real-world contexts, as they seem to provide better results than the application of individual, isolated techniques. Most of the combinations integrate static program analysis with testing techniques. Static program analysis leads to over-approximations, *inter alia*, for analysing the entire code, which tends to generate occasional false positives. In contrast, testing leads to under-approximations, as it is difficult to cover all the code because of the resources required, which tends to generate false negatives. Therefore, the two approaches are used complementarily to improve results in terms of precision, coverage and effectiveness.

VI. THREATS TO VALIDITY

This section discusses the potential threats to validity of this SMS along with the actions we have taken in order to mitigate or minimize them. Although we carefully followed the SMS process in order to minimize the threats to the validity of the results and conclusions drawn in this paper, there are some threats that we faced at their different stages that deserve further discussion.

A. CONSTRUCT VALIDITY

When defining the scope of the SMS, we faced a lack of consensus when researchers refer to the domain dealing

with techniques for detecting anomalies in software artefacts. These techniques are considered in some works to be part of the *Software Quality Assurance* domain (e.g. [13]), but in others to be part of the *Software Quality Control* domain (e.g. [1]). SWEBOK's definitions, however, state that although the ultimate goal of SQA and SQC is to ensure that software systems meet quality requirements, SQA focuses the evaluation on the software development process, while SQC focuses on the software artefacts themselves [1]. Although this study is clearly circumscribed to the latter, we did not exclude the former terms (and others such as verification and validation), but included them in the search string in order to cover all of the related papers, generating an initial pool of 13,180 papers. It mitigated the risk that the study setting does not reflect the construct under study, at the cost of adding additional manual efforts mainly when applying the inclusion and exclusion criteria.

When formulating the strategy for paper selection, we were faced with two threats regarding the completeness of the study, i.e. whether both (1) the database search strategy and (2) the search string enabled all relevant papers to be retrieved. On the one hand, for dealing with the former, we used the Scopus database since it enables us to find the most suitable and complete high-quality refereed research literature for our research. As already mentioned, Scopus indexes high-quality peer-reviewed papers from the most relevant digital libraries for computer science, including journal and conference papers from IEEE Xplore, Springer Link, Science Direct, and ACM. On the other hand, for dealing with validity threats regarding the search string (i.e. missing keywords leading to the exclusion of relevant papers), we (1) used well-known standards, vocabularies and taxonomies in the research field, and (2) carried out a four-iteration validation against 20 papers provided by a senior privacy researcher. The final search string was the conjunction of the three research domains (*software quality control, privacy, and software-based* IS) and each of them was, in turn, represented as a disjunction of domain-related terms, as further explained in Section III-B.

Finally, despite the actions taken, we are aware that our study has limitations mainly related to coverage. The number of candidate papers might have been affected because (1) the search string might not be complete and might require additional or alternative terms, (2) only one search strategy was used to select the candidate papers, and (3) the candidate papers have been filtered by the number of citations. We recognize that these issues can be improved, for instance, using further thesauri, using other search strategies such as snowballing, or by using more lax criteria for the number of citations. However, considering the significant number of candidate papers (13,180), we consider that our results and findings are valuable for providing researchers and practitioners with an overview of the state of the art of SQC techniques focused on privacy aspects.

B. INTERNAL VALIDITY

Individual researcher's bias in (1) deciding whether to include or exclude a candidate paper, (2) classifying it according to the built scheme, and (3) analyzing the results is an internal threat inherent in the study that could lead to biased or erroneous conclusions. We took two main actions to minimize this threat: we standardized the criteria across the research team to ensure a similar understanding, and we cross-checked the papers so that each was reviewed by at least two researchers. In the following subsections we further elaborate on the specific actions carried out during the aforementioned tasks.

Regarding the application of inclusion and exclusion criteria during the screening procedure, the research team (one senior privacy researcher and two PhD privacy researchers) carried out an iterative pilot, aimed at validating the criteria as well as normalizing their understanding, and only passed to the main screening after obtaining a 91% success rate and a high rate of the Krippendorff's alpha coefficient (0.748). Furthermore, as explained in detail in Section III-C, in the first two stages of the main screening (screening based on title and abstract), 10% of the papers were reviewed by at least two researchers, while in the full-text screening, all the papers were reviewed by two researchers in order to ensure the inter-coder reliability. In fact, the full-text screening consisted of a mandatory and an adaptive approach (i.e., the title, abstract, introduction and conclusions were mandatory, but the section explaining the SQC technique was also read, if necessary), thus striving to ensure adequate depth reading and preventing relevant papers from being mistakenly omitted and, inversely, some irrelevant ones being mistakenly included.

As for the classification procedure, much effort was made to build a collectively exhaustive classification scheme in order to consistently classify the 64 papers. Thus, it was built using existing recognized classifications and refining it iteratively through a three-iteration pilot conducted by all members of the research team, making sure that each was reviewed by at least two researchers, discussing in detail and agreeing when divergences appeared. In addition, a mandatory and an adaptive depth reading approach, similar to the full-text screening, was carried out in order to ensure a sufficient reading level before classifying a paper and to avoid misclassification.

Finally, to analyze the results and draw the conclusions, we rely on the collective results of the research team rather than individual researcher interpretations. Therefore, the graphs were generated directly from the team classification results, and the findings and conclusions were drawn from the explicit observations and trends. In this sense, on the basis of the public online replication package, which include both individual and team results, our findings can be traced directly to the classification results and can be both reproduced and validated by other researchers.

C. EXTERNAL VALIDITY

A lack of consensus when researchers refer to the domain addressed in this study (e.g., software quality control vs. software quality assurance) might lead to an erroneous generalization in our findings. The results and conclusions of this SMS are only valid for the techniques that fall within the scope defined in Section III-A, i.e. techniques to detect privacy-related anomalies in software-based information systems. We have made great efforts to set up the SMS protocol systematically —through a detailed definition of the research questions, inclusion and exclusion criteria and classification scheme— and applying it to ensure that general conclusions are valid irrespective of the lack of consensus highlighted. Accordingly, our conclusions do not apply to techniques focusing only on anomaly mitigation rather than anomaly detection; or techniques whose object under evaluation is not a software-based IS, but privacy policies, business processes, network and cryptographic protocols, or datasets in isolation (see Section III-C for more details). Indeed, the exclusive focus on software-based IS explains to some extent the results on the privacy property targeted by the SQC techniques.

The papers found focus almost exclusively on assessing personal data disclosure and compliance, while the other privacy properties are underrepresented or do not appear at all. For instance, we found few contributions targeting transparency/awareness. This can be explained because privacy policies in isolation are out of the scope of our study (we focus on software-based IS), yet privacy policies are widely used to inform users on privacy practices [74]. Something similar occurs with the properties of identifiability, undetectability and unlinkability. The first two properties are related to anonymization or obfuscation algorithms (c.f. [41], [42]), while unlinkability is related to cryptographic algorithms and protocols (c.f. [51], [52]). Both algorithms and protocols are out of the scope of this study, and thus have therefore also been excluded.

VII. RELATED WORK

To the best of our knowledge, there are currently no systematic mapping studies, surveys or reviews that fall into the intersection of the three domains specified in the scope of this study (i.e. *software quality control, software-based IS,* and *privacy*). We found secondary studies related to the intersection of the first two domains but either 1) bounded to a specific SQC technique and not focusing on privacy (c.f. [13], [14]) or 2) bounded to a specific, different quality attribute (c.f. [11], [12]). Thus, they are complementary to ours as they do not address *privacy* and do not cover techniques that can be applied throughout the entire software development process. The following paragraphs describe these related works in more detail.

Garousi *et al.* carried out a systematic mapping study in the field of SQC techniques but focused specifically on *testing techniques* applied in web applications to evaluate security aspects [12]. It is worth mentioning that we actually found several works similar to Garousi's survey in the literature (c.f. [75], [76]); however, we will only refer to this one to explain the similarities and differences with our SMS, which can be generalized to the others. Garousi *et al.*, similar to ours, surveyed techniques aimed at detecting anomalies in software-based IS, although they only focus on the *testing techniques*, which is only one of the six subtypes of techniques we have addressed in our systematic mapping. Most importantly, they studied techniques aimed at detecting *security-related anomalies*, and is narrowed down to web applications only. Our SMS, on the other hand, focuses on privacy and data protection aspects and it is domain-agnostic, hence it includes web domain software systems and mobile applications, among others.

Another closer mapping study has been carried out by Elberzhager *et al.* [13], covering domain-agnostic software-based IS and is not bounded to a particular quality attribute. Elberzhager *et al.* carried out their study in a group of 51 papers that reported on *combined techniques*. None of the 11 combined techniques we studied were included in Elberzhager's pool of papers, as that survey was conducted in 2012, and 10 out of the 11 *combined techniques* of our pool were published from 2013 onwards. Our pool also did not include any of these papers, as they do not address the privacy attribute. Li *et al.* [14] have also carried out a systematic review on static analysis of Android applications without focusing on a particular quality attribute. Based on a pool of 124 papers, they answered, *inter alia*, which kinds of anomalies are detected by these techniques in mobile applications. Thus, there are four papers (ID1044, ID1070, ID1921, ID2107) also included in our pool of papers. Both stand out because static program analysis is only a subtype of the SQC technique used to detect privacy-related anomalies, so we have also addressed model-based privacy analysis and dynamic techniques.

Finally, Sadegui *et al.* have published the most recent related research [11]. They systematically studied static, dynamic and hybrid techniques for the security assessment of Android applications. Although they focused on the security aspects, there are 6 papers that have also been analyzed in our study (ID1026, ID1066, ID1070, ID1911, ID1921, ID2107), which have been categorized by the authors as *grayware* because they can exfiltrate any personal data (we call them *defect detection*). However, as can be seen in Section II-A-2, *defect detection* is only one of the objectives of SQC techniques that address privacy. In addition, there are a significant number of techniques pursuing *specification verification* that have not been studied by Sadeghi's survey. Furthermore, in contrast to that survey, our SMS is not limited to mobile applications.

All in all, our SMS differs from all the aforementioned surveys in the way that we address SQC techniques applied throughout the entire software development lifecycle and, most importantly, it focuses exclusively on privacy aspects. We believe that the aforementioned surveys (as well as other similar ones) and our SMS are complementary and all of them

TABLE 14. List of studied papers.

ID3	K. Bavendiek *et al.*, "Automatically Proving Purpose Limitation in Software Architectures Kai," in *Proc. 34th IFIP TC 11 International Conference on Information Security and Privacy Protection*, Dubai, UAE, 2019, pp. 345–358
ID5	A. Papageorgiou, M. Strigkos, E. Politou, E. Alepis, A. Solanas, and C. Patsakis, "Security and Privacy Analysis of Mobile Health Applications: The Alarming State of Practice," *IEEE Access*, pp. 9390–9403, Jan. 2018.
ID9	P. Ferrara and F. Spoto, "Static analysis for GDPR compliance," in *Proc. CEUR Workshop*, Plejsy, Slovakia, 2018, pp. 1–10.
ID13	P. Ferrara, L. Olivieri, and F. Spoto, "Tailoring taint analysis to GDPR," in *Proc. Annual Privacy Forum*, Barcelona, Spain, 2018, pp. 63–76.
ID17	P. Guarda, S. Ranise, and H. Siswantoro, "Security analysis and legal compliance checking for the design of privacy-friendly information systems," in *Proc. 22nd ACM Symposium on Access Control Models and Technologies*, Indianapolis, USA, 2017, pp. 247–254.
ID21	E. Vanezi, G. M. Kapitsaki, D. Kouzapas, and A. Philippou, "A formal modeling scheme for analyzing a software system design against the GDPR," in *Proc. 14th International Conference on Evaluation of Novel Approaches to Software Engineering*, Crete, Greece, 2019, no. Enase, pp. 68–79
ID25	F. Kammueller, "Formal Modeling and Analysis of Data Protection for GDPR Compliance of IoT Healthcare Systems," in *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, Miyazaki, Japan, 2019, pp. 3319–3324.
ID31	K. Bavendiek, R. Adams, and S. Schupp, "Privacy-Preserving Architectures with Probabilistic Guaranties," in *Proc. 16th Annual Conference on Privacy, Security and Trust*, 2018, pp. 1–10.
ID34	A. S. Ahmadian, D. Struber, V. Riediger, and J. Jurjens, "Model-Based Privacy Analysis in Industrial Ecosystems," in *Proc. European Conference on Modelling Foundations and Applications*, Marburg, Germany, 2017, pp. 126–141,
ID79	X. Liu, S. Zhu, W. W. B, and J. Liu, "Alde: Privacy risk analysis of analytics libraries in the android ecosystem," in *Proc. 12th EAI International Conference on Security and Privacy in Communication Networks*, Guangzhou, China, 2017, pp. 655–672.
ID81	L. Yu, X. Luo, C. Qian, S. Wang, and H. K. N. Leung, "Enhancing the Description-to-Behavior Fidelity in Android Apps with Privacy Policy," *IEEE Trans. Softw. Eng.*, pp. 834–854, Sep. 2018.
ID82	R. Samavi and M. P. Consens, "Publishing privacy logs to facilitate transparency and accountability," *J. Web Semant.* pp. 1–20, Feb. 2018.
ID96	P. Barreto, L. Salgado, and J. Viterbo, "Assessing the communicability of human-data interaction mechanisms in transparency enhancing tools," in *Proc. Federated Conference on Computer Science and Information Systems*, Poznań, Poland, 2018, pp. 897–906.
ID97	A. S. Ahmadian and J. Jurjens, "Supporting model-based privacy analysis by exploiting privacy level agreements," in *Proc. 8th IEEE International Conference on Cloud Computing Technology and Science*, Luxembourg, 2017, pp. 360–365.
ID112	M. Nobakht, Y. Sui, A. Seneviratne, and W. Hu, "Permission Analysis of Health and Fitness Apps in IoT Programming Frameworks," in *Proc. 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, NY, USA, 2018, pp. 533–538.
ID120	S. Bhandari, F. Herbreteau, V. Laxmi, A. Zemmari, P. S. Roop, and M. S. Gaur, "SneakLeak: Detecting multipartite leakage paths in android apps," in *Proc. 16th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2017, pp. 285–292
ID125	Y. He, X. Yang, B. Hu, and W. Wang, "Dynamic privacy leakage analysis of Android third-party libraries," *J. Inf. Secur. Appl.* pp. 259–270, Mar. 2019.
ID128	L. L. Zhang, C.-J. M. Liang, Z. L. Li, Y. Liu, and E.-H. Chen, "Characterizing Privacy Risks of Mobile Apps with Sensitivity Analysis," *IEEE Trans. Mob. Comput.*, pp. 1–14, 2018
ID131	A. S. Ahmadian, J. Jürjens, and D. Strüber, "Extending model-based privacy analysis for the industrial data space by exploiting privacy level agreements," in *Proc. 33rd Annual ACM Symposium on Applied Computing*, Pau, France, 2018, pp. 1142–1149
ID134	V. Jain, S. Bhandari, V. Laxmi, M. S. Gaur, and M. Mosbah, "SniffDroid: Detection of inter-app privacy leaks in android," in *Proc. 16th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2017, pp. 331–338.
ID162	M. Hatamian, N. Momen, L. Fritsch, and K. Rannenberg, "A Multilateral Privacy Impact Analysis Method for Android Apps," in *Proc. Annual Privacy Forum*, Rome, Italy, 2019, pp. 87–106.
ID164	D. Sun, C. Guo, D. Zhu, and W. Feng, "Secure HybridApp: A detection method on the risk of privacy leakage in HTML5 hybrid applications based on dynamic taint tracking," in *Proc. 2nd IEEE International Conference on Computer and Communications*, Krakow, Poland, 2017, pp. 2771–2775
ID174	M. Diamantaris, E. P. Papadopoulos, and E. P. Markatos, "REAPER: Real-time App Analysis for Augmenting the Android Permission System," in *Proc. 9th ACM Conference on Data and Application Security and Privacy*, TX, USA, 2019, pp. 37–48.
ID195	O. Ayalon and E. Toch, "A/P(rivacy) Testing: Assessing applications for social and institutional privacy," in *Proc. Conference on Human Factors in Computing Systems*, Scotland, UK, 2019, pp. 1–6.
ID199	Q. Jia, L. Zhou, H. Li, R. Yang, S. Du, and H. Zhu, "Who Leaks My Privacy: Towards Automatic and Association Detection with GDPR Compliance," in *Proc. 14th International Conference on Wireless Algorithms, Systems, and Applications*, Honolulu, USA, 2019, pp. 137–148.
ID266	A. Rahman, P. Pradhan, A. Partho, and L. Williams, "Predicting Android Application Security and Privacy Risk with Static Code Metrics," in *Proc. 4th IEEE/ACM International Conference on Mobile Software Engineering and Systems*, Buenos Aires, Argentina, 2017, pp. 149–153
ID293	Z. Meng, Y. Xiong, W. Huang, L. Qin, X. Jin, and H. Yan, "AppScalpel: Combining static analysis and outlier detection to identify and prune undesirable usage of sensitive data in Android applications," *Neurocomputing*, pp. 10–25, Feb. 2019
ID361	P. Feng, J. Ma, and C. Sun, "Selecting Critical Data Flows in Android Applications for Abnormal Behavior Detection," *Mob. Inf. Syst.*, Apr. 2017.

TABLE 14. *(continued.)* **List of studied papers.**

ID519	A. S. Ahmadian, S. Peldszus, Q. Ramadan, and J. Jürjens, "Model-based privacy and security analysis with CARiSMA," in *Proc. 11th Joint Meeting of the European Software Engineering Conference*, Paderborn, Germany, 2017, pp. 989–993.
ID536	S. Kelkar, T. Kraus, D. Morgan, J. Zhang, and R. Dai, "Analyzing HTTP-Based Information Exfiltration of Malicious Android Applications," in *Proc. 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2018, pp. 1642–1645.
ID959	M. Ghafari, P. Gadient, and O. Nierstrasz, "Security Smells in Android," in *Proc. 17th IEEE International Working Conference on Source Code Analysis and Manipulation*, 2017, pp. 121–130.
ID1026	G. Barbon, A. Cortesi, P. Ferrara, M. Pistoia, and O. Tripp, "Privacy analysis of android apps: implicit flows and quantitative analysis," in *Proc. IFIP International Conference on Computer Information Systems and Industrial Management*, Warsaw, Poland, 2015, pp. 3–23.
ID1029	A. S. De Oliveira, J. Sendor, A. Garaga, and K. Jenatton, "Monitoring personal data transfers in the cloud," in *Proc. 5th IEEE International Conference on Cloud Computing Technology and Science*, Bristol, UK, 2013, pp. 347–354
ID1039	G. Barbon, A. Cortesi, P. Ferrara, and E. Steffinlongo, "DAPA: Degradation-aware privacy analysis of Android apps," in *Proc. 12th International Workshop on Security and Trust Management*, Crete, Greece, 2016, pp. 32–46.
ID1043	J. C. J. Keng, L. Jiang, T. K. Wee, and R. K. Balan, "Graph-aided directed testing of android applications for checking runtime privacy behaviours," in *Proc. 11th International Workshop on Automation of Software Test*, Austin, USA, 2016, pp. 57–63.
ID1044	A. Cortesi, P. Ferrara, M. Pistoia, and O. Tripp, "Datacentric semantics for verification of privacy policy compliance by mobile applications," in *Proc. 16th International Conference on Verification, Model Checking, and Abstract Interpretation*, 2015, pp. 61–79.
ID1051	S. Sen, S. Guha, A. Datta, S. K. Rajamani, J. Tsai, and J. M. Wing, "Bootstrapping privacy compliance in big data systems," in *Proc. 35th IEEE Symposium on Security and Privacy*, San Jose, USA, 2014, pp. 327–342
ID1056	Y. Kim, T. Oh, and J. Kim, "Analyzing User Awareness of Privacy Data Leak in Mobile Applications," *Mob. Inf. Syst.*, Nov. 2015
ID1066	M. Spreitzenbarth, F. Freiling, F. Echtler, T. Schreck, and J. Hoffmann, "Mobile-sandbox: Having a deeper look into Android applications," in *Proc. 28th Annual ACM Symposium on Applied Computing*, Coimbra, Portugal, 2013, pp. 1808–1815.
ID1067	F. Knirsch, D. Engel, C. Neureiter, M. Frincu, and V. Prasanna, "Model-driven privacy assessment in the smart grid," in *Proc. 1st International Conference on Information Systems Security and Privacy*, Loire Valley, France, 2015, pp. 173–181.
ID1070	D. Geneiatakis, I. N. Fovino, I. Kounelis, and P. Stirparo, "A Permission verification approach for android mobile applications," *Comput. Secur.*, pp. 192–205, Nov. 2015.
ID1078	A. Ali-Gombe, I. Ahmed, G. G. Richard, and V. Roussev, "AspectDroid: Android app analysis system," in *Proc. 6th ACM Conference on Data and Application Security and Privacy*, New Orleans, USA, 2016, pp. 145–147
ID1137	M. Junaid, D. Liu, and D. Kung, "Dexteroid: Detecting malicious behaviors in Android apps using reverse-engineered life cycle models," *Comput. Secur.*, pp. 92–117, Mar. 2016.
ID1166	F. Amato and F. Moscato, "A model driven approach to data privacy verification in e-health systems," *Trans. Data Priv.*, pp. 273–296, Aug. 2015.
ID1193	O. Chowdhury, L. Jia, D. Garg, and A. Datta, "Temporal mode-checking for runtime monitoring of privacy policies," in *Proc. 26th International Conference on Computer Aided Verification*, Vienna, Austria, 2014, pp. 131–149
ID1195	H. Xu, Y. Zhou, C. Gao, Y. Kang, and M. R. Lyu, "SpyAware: Investigating the privacy leakage signatures in app execution traces," in *Proc. 26th IEEE International Symposium on Software Reliability Engineering*, Washington, USA, 2016, pp. 348–358.
ID1257	M. Pistoia, O. Tripp, P. Centonze, and J. W. Ligman, "Labyrinth: Visually Configurable Data-Leakage Detection in Mobile Applications," in *Proc. 16th IEEE International Conference on Mobile Data Management*, 2015, pp. 279–286.
ID1494	T. D. Breaux, D. Smullen, and H. Hibshi, "Detecting repurposing and over-collection in multi-party privacy requirements specifications," in *Proc. 23rd IEEE International Requirements Engineering Conference*, Ottawa, Canada, 2015, pp. 166–175.
ID1572	G. Ascia *et al.*, "Making android apps data-leak-safe by data flow analysis and code injection," in *Proc. 25th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Paris, France, 2016, pp. 205–210.
ID1684	D. Le Métayer, "Privacy by design: Formal framework for the analysis of architectural choices," in *Proc. 3rd ACM Conference on Data and Application Security and Privacy*,Texas, USA, 2013, pp. 95–104
ID1805	X. Shu, D. Yao, and E. Bertino, "Privacy-preserving detection of sensitive data exposure," *IEEE Trans. Inf. Forensics Secur.* pp. 1092–1103, 2015
ID1911	P. Gilbert, B. G. Chun, L. P. Cox, and J. Jung, "Vision: Automated security validation of mobile apps at app markets," in *Proc. 9th International Conference on Mobile Systems, Applications, and Services*, Maryland, USA, 2011, pp. 21–25.
ID1921	C. Mann and A. Starostin, "A framework for static detection of privacy leaks in android applications," in *Proc. 27th Annual ACM Symposium on Applied Computing*, Riva del Garda, Italy, 2012, pp. 1457–1462.
ID1930	P. Godefroid, J. D. Herbsleb, L. J. Jagadeesan, and D. Li, "Ensuring privacy in presence awareness systems: An automated verification approach," in *Proc. ACM Conference on Computer Supported Cooperative Work*, Pennsylvania USA, 2000, pp. 59–68.
ID1948	D. Garg and L. Jia, "Policy Auditing over Incomplete Logs : Theory , Implementation and Applications Categories and Subject Descriptors," in *Proc. 18th ACM Conference on Computer and Communications Security*, 2011, pp. 151–162.
ID1951	M. Kost, J. C. Freytag, F. Kargl, and A. Kung, "Privacy verification using ontologies," in *Proc. 6th International Conference on Availability, Reliability and Security*, Vienna, Austria, 2011, pp. 627–632.
ID1992	M. C. Tschantz, D. Kaynar, and A. Datta, "Formal verification of differential privacy for interactive systems (extended abstract)," in *Proc. Electronic Notes in Theoretical Computer Science*, 2011, pp. 61–79.
ID2100	S. Benbernou, H. Meziane, and M. S. Hacid, "Run-Time Monitoring for Privacy-Agreement," in *Proc. 5th International Conference on Service-Oriented Computing*, Vienna, Austria, 2007, pp. 353–364.

TABLE 14. ***(continued.)*** **List of studied papers.**

ID2107	L. Batyuk, M. Herpich, S. A. Camtepe, K. Raddatz, A. D. Schmidt, and S. Albayrak, "Using static analysis for automatic assessment and mitigation of unwanted and malicious activities within Android applications," in *Proc. 6th International Conference on Malicious and Unwanted Software*, Fajardo, USA, 2011, pp. 66–72
ID2130	A. Datta *et al.*, "Understanding and Protecting Privacy: Formal Semantics and Principled Audit Mechanisms," in *Proc. 7th International Conference on Information Systems Security*, Kolkata, India, 2011, pp. 1–27.
ID2206	M. Tran, X. Dong, Z. Liang, and X. Jiang, "Tracking the Trackers : Fast and Scalable Dynamic," in *Proc. 10th International Conference on Applied Cryptography and Network Security*, Singapore, 2012, pp. 418–435.
ID2228	D. Jang, R. Jhala, S. Lerner, and H. Shacham, "An empirical study of privacy-violating information flows in JavaScript web applications," in *Proc. 17th ACM Conference on Computer and Communications Security*, 2010, pp. 270–283.
ID2406	R. J. Hall and A. Zisman, "Validating personal requirements by assisted symbolic behavior browsing," in *Proc. 19th International Conference on Automated Software Engineering*, Linz, Austria, 2004, pp. 56–66.
ID2686	J. Yu, S. Zhang, P. Liu, and Z. T. Li, "LeakProber: A framework for profiling sensitive data leakage paths," in *Proc. ACM Conference on Data and Application Security and Privacy*, Texas, USA, 2011, pp. 75–84.

provide a better view on the landscape of SQC techniques addressing miscellaneous quality attributes.

VIII. CONCLUSIONS

The major result of this systematic mapping study is the identification and classification of existing SQC techniques that detect privacy-related anomalies during the development process of software-based IS. For researchers, we have provided an overview of research efforts on the different types and subtypes of SQC techniques used throughout the software development process, highlighting the targeted privacy properties, analyzing some research trends, and identifying main research institutions and publication venues. For both researchers and practitioners, we showed the specific targeted software artefacts to which the SQC techniques can be applied, as well as the conditions or criteria used to detect anomalies. The level of maturity of the different techniques is also shown, highlighting those that could be closer to their application in practice.

Our future work points towards improving the coverage and precision in assessing privacy compliance, particularly in data subject-side information systems e.g. mobile apps. For that, we are leveraging upon the combination of different techniques, which has demonstrated to be a promising path according to the findings described, and the application of interdisciplinary knowledge to move beyond policy compliance and towards truly privacy compliance.

APPENDIX

See Table 14.

REFERENCES

[1] P. Bourque and R. E. Fairley, "SWEBOK, version 3.0: Guide to the software engineering body of knowledge," IEEE Comput. Soc., Piscataway, NJ, USA, Tech. Rep., 2014.

[2] *Information Technology—Software Product Quality, Part 1: Quality Model*, Standard ISO/IEC 9126-1, 2000.

[3] N. G. Mohammadi, "An analysis of software quality attributes and their contribution to trustworthiness," in *Proc. 3rd Int. Conf. Cloud Comput. Services Sci.*, Aachen, Germany, 2015, pp. 542–552.

[4] A. Cavoukian, "Privacy by design the 7 foundational principles," Inf. Privacy Commissioner, Toronto, ON, Canada, Tech. Rep., 2009.

[5] European Parliament and the Council of the European Union. (2016). *General Data Protection Regulation*. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN.

[6] *IEEE Standard for System, Software, and Hardware Verification and Validation*, Standard IEEE 1012-2012, 2017.

[7] D. J. Solove, "A taxonomy of privacy," *Univ. Pennsylvania Law Rev.*, vol. 154, pp. 477–560, Jan. 2006.

[8] D. K. Mulligan and C. Koopman, "Theorizing privacy's contestability: A multi-dimensional analytic of privacy," in *Proc. Special Workshop Inf. Privacy*, Fort Worth, TX, USA, Feb. 2013, pp. 1026–1029.

[9] S. Gürses, "Can you engineer privacy?" *Commun. ACM*, vol. 57, no. 8, pp. 20–23, Aug. 2014.

[10] H. Nissenbaum, "Privacy as contextual integrity," *Washington Law Rev.*, vol. 79, pp. 101–139, Jun. 2004.

[11] A. Sadeghi, H. Bagheri, J. Garcia, and S. Malek, "A taxonomy and qualitative comparison of program analysis techniques for security assessment of Android software," *IEEE Trans. Softw. Eng.*, vol. 43, no. 6, pp. 492–530, Jun. 2017.

[12] V. Garousi, A. Mesbah, A. Betin-Can, and S. Mirshokraie, "A systematic mapping study of Web application testing," *Inf. Softw. Technol.*, vol. 55, no. 8, pp. 1374–1396, Aug. 2013.

[13] F. Elberzhager, J. Münch, and V. T. N. Nha, "A systematic mapping study on the combination of static and dynamic quality assurance techniques," *Inf. Softw. Technol.*, vol. 54, no. 1, pp. 1–15, Jan. 2012.

[14] L. Li, T. F. Bissyandé, M. Papadakis, S. Rasthofer, A. Bartel, D. Octeau, J. Klein, and L. Traon, "Static analysis of Android apps: A systematic literature review," *Inf. Softw. Technol.*, vol. 88, pp. 67–95, Aug. 2017.

[15] *Systems and Software Engineering—Vocabulary*, Standard ISO/IEC/IEEE 24765, 2010.

[16] *Systems and Software Engineering—Software Life Cycle Processes*, Standard ISO/IEC/IEEE 12207, 2017.

[17] I. Sommerville, *Software Engineering*, 9th ed. Boston, MA, USA: Pearson, 2010.

[18] K. Wuyts, *Privacy Threats in Software Architectures*. Leuven, Belgium: KU Leuven, 2015.

[19] M. Hansen, M. Jensen, and M. Rost, "Protection goals for privacy engineering," in *Proc. IEEE Secur. Privacy Workshops*, San Jose, CA, USA, May 2015, pp. 159–166.

[20] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, "Requirements engineering paper classification and evaluation criteria: A proposal and a discussion," *Requirements Eng.*, vol. 11, no. 1, pp. 102–107, Mar. 2006.

[21] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proc. 12th Int. Conf. Eval. Assessment Softw. Eng.*, Bari, Italy, Jun. 2008, p. 10.

[22] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf. Softw. Technol.*, vol. 64, pp. 1–18, Aug. 2015.

[23] T. Fielding, "Architectural styles and the design of network-based software architectures," M.S. thesis, Univ. California, Irvine, Irvine, CA, USA, 2000.

[24] B. Kitchenham, "What's up with software metrics?—A preliminary mapping study," *J. Syst. Softw.*, vol. 83, pp. 37–51, Jun. 2010.

[25] J. C. Caiza, Y. S. Martín, D. S. Guamán, J. M. Del Alamo, and J. C. Yelmo, "Reusable elements for the systematic design of privacy-friendly information systems: A mapping study," *IEEE Access*, vol. 7, pp. 66512–66535, May 2019.

[26] A. Cavacini, "What is the best database for computer science journal articles?" *Scientometrics*, vol. 102, no. 3, pp. 2059–2071, Mar. 2015.

[27] The Institute of Electrical and Electronics Engineers (IEEE). (2017). *IEEE Thesaurus, Version 1.0*. [Online]. Available: http://www.ieee.org/publications_standards/publications/journmag/ieee_thesaurus.pdf
[28] ACM Digital Library. *The 2012 ACM Computing Classification System*. Accessed: Jul. 16, 2019. [Online]. Available: http://dl.acm.org/ccs/ccs.cfm?id=10010371&lid=0.10
[29] Thomson Reuters. (2019). *InCites Essential Science Indicators: Field Baselines*. Accessed: Jul. 24, 2019. [Online]. Available: https://esi.incites.thomsonreuters.com/BaselineAction.action
[30] C. Kohl, E. J. McIntosh, S. Unger, N. R. Haddaway, S. Kecke, J. Schiemann, and R. Wilhelm, "Online tools supporting the conduct and reporting of systematic reviews and systematic maps: A case study on CADIMA and review of existing tools," *Environ. Evidence*, vol. 7, no. 1, pp. 1–17, Dec. 2018.
[31] P. B. Brandtzaeg, A. Pultier, and G. M. Moen, "Losing control to data-hungry apps: A mixed-methods approach to mobile app privacy," *Social Sci. Comput. Rev.*, vol. 37, no. 4, pp. 466–488, Aug. 2019.
[32] H. Liu, C. Li, X. Jin, J. Li, Y. Zhang, and D. Gu, "Smart solution, poor protection: An empirical study of security and privacy issues in developing and deploying smart home devices," in *Proc. Workshop IoT Secur. Privacy*, Dallas, TX, USA, 2017, pp. 13–18.
[33] A. Tekeoglu and A. S. Tosun, "A testbed for security and privacy analysis of IoT devices," in *Proc. IEEE 13th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Brasilia, Brazil, Oct. 2016, pp. 2–7.
[34] M. Ikram, N. Vallina-Rodriguez, S. Seneviratne, M. A. Kaafar, and V. Paxson, "An analysis of the privacy and security risks of Android VPN permission-enabled apps," in *Proc. ACM Internet Meas. Conf. (IMC)*, Santa Monica, CA, USA, 2016, pp. 349–364.
[35] Z. Li, T. J. Oechtering, and D. Gunduz, "Smart meter privacy based on adversarial hypothesis testing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 774–778.
[36] T. Antignac, R. Scandariato, and G. Schneider, "Privacy compliance via model transformations," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, London, U.K., Apr. 2018, pp. 120–126.
[37] X. Wu, W. Dou, and Q. Ni, "Game theory based privacy preserving analysis in correlated data publication," in *Proc. Australas. Comput. Sci. Week Multiconf. (ACSW)*, Geelong, VIC, Australia, 2017, pp. 1–10.
[38] N. A. Shozi and J. Mtsweni, "A socio-technical systems analysis of privacy issues in social media sites," in *Proc. 14th Int. Conf. Cyber Warfare Secur.*, Stellenbosch, South Africa, 2019, pp. 369–377.
[39] N. Tsalis, A. Mylonas, and D. Gritzalis, "An intensive analysis of security and privacy browser add-ons," in *Proc. 11th Int. Conf. Risks Secur. Internet Syst.*, Roscoff, France, 2016, pp. 258–273.
[40] L. H. Iwaya, S. Fischer-Hubner, R.-M. Ahlfeldt, and L. A. Martucci, "MHealth: A privacy threat analysis for public health surveillance systems," in *Proc. IEEE 31st Int. Symp. Comput.-Based Med. Syst. (CBMS)*, Karlstad, Sweden, Jun. 2018, pp. 42–47.
[41] A. M. T. Ali-Eldin, A. Zuiderwijk, and M. Janssen, "Opening more data: A new privacy risk scoring model for open data," in *Proc. 7th Int. Symp. Bus. Modeling Softw. Design*, Barcelona, Spain, 2017, pp. 146–154.
[42] R. Pellungrini, L. Pappalardo, F. Pratesi, and A. Monreale, "Fast estimation of privacy risk in human mobility data," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.*, Trento, Italy, 2017, pp. 415–426.
[43] F. Prasser, R. Bild, and K. A. Kuhn, "A generic method for assessing the quality of de-identified health data," in *Proc. Med. Informat. Eur. Conf.*, Munich, Germany, 2016, pp. 312–316.
[44] J. Henriksen-Bulmer, S. Faily, and S. Jeary, "Privacy risk assessment in context: A meta-model based on contextual integrity," *Comput. Secur.*, vol. 82, pp. 270–283, May 2019.
[45] R. L. Rutledge, A. K. Massey, and A. I. Anton, "Privacy impacts of IoT devices: A SmartTV case study," in *Proc. IEEE 24th Int. Requirements Eng. Conf. Workshops (REW)*, Beigin, China, Sep. 2016, pp. 261–270.
[46] R. N. Zaeem, R. L. German, and K. S. Barber, "PrivacyCheck: Automatic summarization of privacy policies using data mining," *ACM Trans. Internet Technol.*, vol. 18, no. 4, pp. 1–18, Nov. 2018.
[47] J. Caramujo and A. M. R. D. Silva, "Analyzing privacy policies based on a privacy-aware profile: The facebook and LinkedIn case studies," in *Proc. IEEE 17th Conf. Bus. Informat.*, Thessaloniki, Greece, Jul. 2015, pp. 77–84.
[48] P. Pullonen, J. Tom, R. Matulevičius, and A. Toots, "Privacy-enhanced BPMN: Enabling data privacy analysis in business processes models," *Softw. Syst. Model.*, vol. 18, no. 6, pp. 3235–3264, Dec. 2019.
[49] A. Toots, "Business process privacy analysis in PLEAK," in *Proc. Fundam. Approaches Softw. Eng.*, Prague, Czech Republic, 2019, pp. 306–312.
[50] S. Ghanavati, D. Amyot, and L. Peyton, "A requirements management framework for privacy compliance," in *Proc. Workshop Requirements Eng.*, Toronto, ON, Canada, 2007, pp. 149–159.
[51] N. Fotiou, S. Arianfar, M. Särelä, and G. C. Polyzos, "A framework for privacy analysis of ICN architectures," in *Proc. Annu. Privacy Forum*, Lisbon, Portugal, 2014, pp. 117–132.
[52] R. Kusters, T. Truderung, and A. Vogt, "Formal analysis of Chaumian mix nets with randomized partial checking," in *Proc. IEEE Symp. Secur. Privacy*, San Jose, CA, USA, May 2014, pp. 343–358.
[53] H. M. N. A. Hamadi, A. Gawanmeh, and M. A. Al-Qutayri, "Extended abstract: Theorem proving verification of privacy in WBSN for healthcare systems," in *Proc. IEEE 20th Int. Conf. Electron., Circuits, Syst. (ICECS)*, Abu Dhabi, UAE, Dec. 2013, pp. 100–101.
[54] C. Rosenberger, "Evaluation of biometric template protection schemes based on a transformation," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, Madeira, Portugal, 2018, pp. 216–224.
[55] K. Wang, G. Bai, N. Dong, and J. S. Dong, "A framework for formal analysis of privacy on SSO protocols," in *Proc. 14th EAI Int. Conf. Secur. Privacy Commun. Netw.*, Singapore, 2018, pp. 763–777.
[56] B. Smyth, M. D. Ryan, and L. Chen, "Formal analysis of privacy in direct anonymous attestation schemes," *Sci. Comput. Program.*, vol. 111, pp. 300–317, Nov. 2015.
[57] B. Chakraborty, D. Sadhya, S. Verma, and K. P. Singh, "Information theoretic analysis of privacy in a multiple query-response based differentially private framework," in *Proc. Int. Conf. Commun., Netw. Comput.*, Singapore, 2019, pp. 262–272.
[58] J. Dreier, A. Kassem, and P. Lafourcade, "Formal analysis of e-cash protocols," in *Proc. 12th Int. Joint Conf. e-Bus. Telecommun.*, Colmar, France, 2015, pp. 65–75.
[59] N. Ansari, P. Sakarindr, E. Haghani, C. Zhang, A. K. Jain, and Y. Q. Shi, "Evaluating electronic voting systems equipped with voter-verified paper records," *IEEE Security Privacy*, vol. 6, no. 3, pp. 30–39, May 2008.
[60] A. S. Ahmadian, D. Strüber, V. Riediger, and J. Jürjens, "Supporting privacy impact assessment by model-based privacy analysis," in *Proc. 33rd ACM/SIGAPP Symp. Appl. Comput. (SAC)*, Pau, France, 2018, pp. 1467–1474.
[61] S. J. De and D. Le Métayer, "PRIAM: A privacy risk analysis methodology," in *Proc. 11th Int. Workshop Data Privacy Manage.*, Crete, Greece, 2016, pp. 221–229.
[62] V. L. Shivraj, M. A. Rajan, and P. Balamuralidhar, "A graph theory based generic risk assessment framework for Internet of Things (IoT)," in *Proc. 11th IEEE Int. Conf. Adv. Netw. Telecommun. Syst.*, Odisha, India, Dec. 2017, pp. 1–6.
[63] A. Mylonas, M. Theoharidou, and D. Gritzalis, "Assessing privacy risks in Android: A user-centric approach," in *Proc. 2nd Int. Workshop Risk Assessment Risk-Driven Test.*, Naples, Italy, 2014, pp. 21–37.
[64] A. Ukil, S. Bandyopadhyay, and A. Pal, "IoT-privacy: To be private or not to be private," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Toronto, ON, Canada, Apr./May 2014, pp. 123–124.
[65] F. Loi, A. Sivanathan, H. H. Gharakheili, A. Radford, and V. Sivaraman, "Systematically evaluating security and privacy for consumer IoT devices," in *Proc. Workshop Internet Things Secur. Privacy*, Dallas, TX, USA, 2017, pp. 1–6.
[66] A. Alarifi, M. Alsaleh, and N. Alomar, "A model for evaluating the security and usability of e-banking platforms," *Computing*, vol. 99, no. 5, pp. 519–535, May 2017.
[67] P. Wang, K.-M. Chao, C.-C. Lo, W.-H. Lin, H.-C. Lin, and W.-J. Chao, "Using malware for software-defined networking-based smart home security management through a taint checking approach," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 8, pp. 1–23, Aug. 2016.
[68] X. Wei, I. Neamtiu, and M. Faloutsos, "Whom does your Android app talk to?" in *Proc. IEEE Global Commun. Conf.*, San Diego, CA, USA, Dec. 2015, pp. 1–6.
[69] S. Gabmeyer, P. Kaufmann, M. Seidl, M. Gogolla, and G. Kappel, "A feature-based classification of formal verification techniques for software models," *Softw. Syst. Model.*, vol. 18, no. 1, pp. 473–498, Feb. 2019.
[70] Object Management Group. (2011). *Unified Modeling Language, Superstructure (V2.4.1)*. Accessed: Jan. 4, 2020. [Online]. Available: https://www.omg.org/spec/UML/2.4.1
[71] Union Agency for Network and Information Security (ENISA). (2017). *A Study on the App Development Ecosystem and the Technical Implementation of GDPR*. Accessed: Nov. 23, 2019. [Online]. Available: https://www.enisa.europa.eu/publications/privacy-and-data-protection-in-mobile-applications

[72] J. Zang, K. Dummit, J. Graves, P. Lisker, and L. Sweeney, "Who knows what about me? A survey of behind the scenes personal data sharing to third parties by mobile apps," *Technol. Sci.*, vol. 30, pp. 1–53, Oct. 2015.
[73] S. Kununka, N. Mehandjiev, and P. Sampaio, "A comparative study of Android and iOS mobile applications' data handling practices versus compliance to privacy policy," in *Proc. IFIP Int. Summer School Privacy Identity Manage.*, Vienna, Austria, 2018, pp. 301–313.
[74] F. Schaub, R. Balebako, and L. F. Cranor, "Designing effective privacy notices and controls," *IEEE Internet Comput.*, early access, Jun. 16, 2017, doi: 10.1109/MIC.2017.265102930.
[75] A. Tahir, D. Tosi, and S. Morasca, "A systematic review on the functional testing of semantic Web services," *J. Syst. Softw.*, vol. 86, no. 11, pp. 2877–2889, Nov. 2013.
[76] M. Palacios, J. García-Fanjul, and J. Tuya, "Testing in service oriented architectures with dynamic binding: A mapping study," *Inf. Softw. Technol.*, vol. 53, pp. 383–384, Mar. 2012.

DANNY S. GUAMÁN received the Engineering degree in electronics and networking from Escuela Politécnica Nacional, Quito, Ecuador, in 2010, and the M.Sc. degree in networking and telematics engineering from the Universidad Politécnica de Madrid, Madrid, Spain, in 2013, where he is currently pursuing the Ph.D. degree.

He is currently an Assistant Professor with Escuela Politécnica Nacional. His main research interests include the analysis of data disclosure and the assessment of privacy compliance in information systems.

JOSE M. DEL ALAMO is currently an Associate Professor (with tenure) with the Universidad Politécnica de Madrid (DIT-UPM). His research interests include issues related to personal data management, including personal data disclosure, identity, privacy and trust management, and considering these aspects to advance software and systems engineering methodologies applying approaches such as privacy-by-design and privacy-by-default.

Dr. Del Alamo has been the Co-Chair of the IEEE International Workshop on Privacy Engineering, co-located to the IEEE Symposium on Security and Privacy, since 2015.

JULIO C. CAIZA received the Engineering degree in electronics and networking from Escuela Politécnica Nacional, Quito, Ecuador, in 2010, and the M.Sc. degree in networking and telematics engineering from the Universidad Politécnica de Madrid, Madrid, Spain, in 2013, where he is currently pursuing the Ph.D. degree in telematics systems engineering.

He is currently an Assistant Professor with Escuela Politécnica Nacional. His main research interest includes the design of privacy-friendly information systems.