IoTPrivComp: A Measurement Study of Privacy Compliance in IoT Apps

Javaria Ahmad, Fengjun Li, and Bo Luo(✉)

Department of Electrical Engineering and Computer Science, Center for High Assurance and Secure Systems (HASS), Institute of Information Sciences (I2S), The University of Kansas, Lawrence, KS, USA
{javaria.ahmad,fli,bluo}@ku.edu

Abstract. The growth of IoT apps poses increasing concerns about sensitive data leaks. While privacy policies are required to describe how IoT apps use private user data (i.e., data practice), problems such as missing, inaccurate, and inconsistent policies have been repeatedly reported. Therefore, it is important to assess the actual data practice in IoT apps and identify the potential gaps between the actual and declared data usage. In this work, we conducted a measurement study using our framework called IoTPrivComp, which applies an automated analysis of IoT apps' code and privacy policies to identify compliance gaps. We collected 1,489 IoT apps with English privacy policies from the Play Store. IoTPrivComp found 532 apps with sensitive external data flows, among which 408 (76.7%) apps had undisclosed data leaks. Moreover, 63.4% of the data flows that involved health and wellness data was inconsistent with the practices disclosed in the apps' privacy policies.

Keywords: IoT · Compliance · Security · Privacy

1 Introduction

Regulations such as EU General Data Protection Regulation (GDPR) [43] and the California Online Privacy Protection Act require a service provider (e.g., websites or mobile apps) who collects personally identifiable data from users to disclose its actions with the collected data in the privacy policy. Therefore, privacy policies nowadays become a standard practice to notify users about the necessary data collection, management, and/or sharing operations that a mobile or IoT app requests to perform. However, the state-of-the-art (SOTA) implementations of privacy policies face two main challenges: (i) the privacy policies are often difficult to comprehend [17,29], while the users are unwilling to spend the time and effort necessary to understand the policies; and (ii) while the vendors should disclose user-data-related practices in privacy policies, recent studies [4,37,46,49,52] uncovered various issues showing the policies were incomplete or inconsistent with the actual practices.

V. Atluri et al. (Eds.): ESORICS 2022, LNCS 13555, pp. 589–609, 2022.
https://doi.org/10.1007/978-3-031-17146-8_29

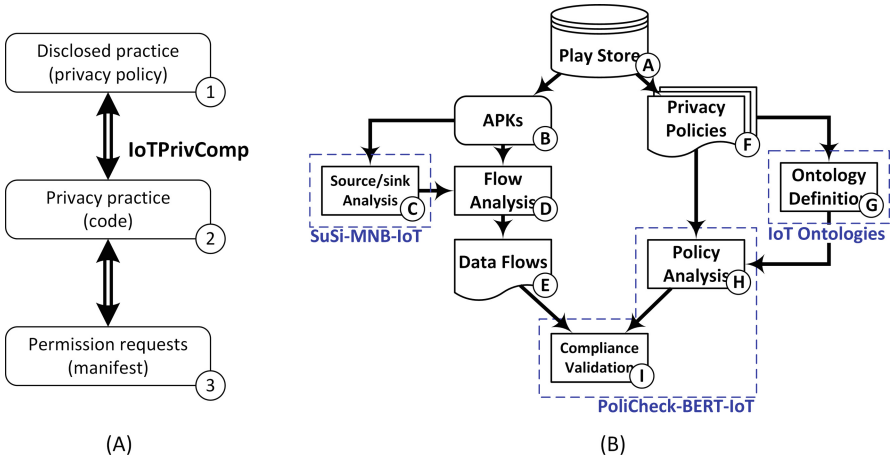


Fig. 1. (A) Privacy compliance measurement for mobile/IoT apps.; (B) The flowchart and key contributions of IoTPrivComp.

Such challenges motivate researchers to study privacy policies and practices. Based on their objectives and methodologies, SOTA research efforts can be categorized into three directions: (1) *privacy policy comprehension* (Fig. 1-A ①) that focuses on facilitating the (automated) understanding of privacy policies [22,42]; (2) *privacy threat detection* (Fig. 1-A ② and ③) that aims to examine app code and behaviors to identify potential privacy threats [23,35]; and (3) *privacy compliance gap detection* (Fig. 1-A ① and ②), which studies the gaps between privacy policies and data practices [4,13,21,27,31,51,52].

In this paper, we present a measurement study to investigate the privacy compliance gaps in *IoT apps*, which are mobile applications interacting with or managing IoT devices. With the rapid deployment of IoT technologies in our daily life, the IoT devices collect personal data such as heart rate, pulse, voice, biometrics, and location, which raise increasing privacy concerns [25]. Therefore, it is important to understand the privacy practices of IoT apps, as they often access more sensitive data than general mobile apps. However, **the compliance gap between privacy policies and privacy practices in IoT apps is yet to be investigated**. Our initial exploration shows that the off-the-shelf (OTS) tools for IoT code and privacy policy analysis provide insufficient performance. Therefore, we first developed IoTPrivComp, a framework for IoT code and policy consistency analysis. IoTPrivComp consists of five main components as shown in Fig. 1-B: (1) a new ontology to represent entity and data object relationships in privacy policies; (2) the SuSi-MNB-IoT sink analysis module that uses a Multinomial Naive Bayesian (MNB) classifier to analyze data flows and identify sinks; (3) a static code analysis module to identify leaks of sensitive information through external data flows; (4) a model based on Bidirectional Encoder Representations from Transformers (BERT) to identify entity and data objects from IoT privacy policies; and (5) the PoliCheck-BERT-IoT compliance analysis module that finally identifies the inconsistent privacy disclosures.

With IoTPrivComp, we further present a large-scale measurement study of the inconsistencies between the practices disclosed in privacy policies (Fig. 1-A ①) and the privacy practices implemented in app code (Fig. 1-A ②). In particular, we aim to answer the following questions: *(1) what does the landscape of IoT app privacy compliance look like?* While we are interested in the current practices that IoT apps take to be compliant with privacy regulations, there lacks such an overview in the literature. *(2) Which privacy compliance gaps exist in IoT apps?* Compliance issues exist when an app's actual practices are not consistent with the practices disclosed in its privacy policy. For example, does the privacy policy of an app fully disclose all types of private data transmitted to the first and third parties? And *(3) does there exist any patterns in privacy compliance gaps?* For example, are certain types of IoT data more commonly associated with compliance issues? Our primary contributions are three-fold:

1. We conduct a measurement study to identify the privacy gaps between the privacy practices and disclosures in 1,951 IoT apps. IoTPrivComp is the first attempt to autonomously validate privacy compliance of IoT apps at this scale.
2. We show that simply assembling OTS tools only provides limited performance for IoT compliance validation and non-trivial modifications/enhancements are necessary. We developed an automated privacy compliance analysis tool for IoT apps, called IoTPrivComp, with a new SuSi-MNB-IoT mechanism for sink identification and a new PoliCheck-BERT-IoT mechanism for privacy policy analysis, and open-sourced it[1]. With all the novel improvements, IoTPrivComp achieves significantly better performance (94% accuracy) than the OTS baseline.
3. We examined 1,951 IoT apps from Google Play Store and analyzed the privacy disclosure gaps. For instance, out of 532 apps with sensitive data flows, we identified compliance violations in 408 (76.7%) apps. We further provided a comprehensive analysis of the inconsistent disclosures and the leaked data.

Ethics: This study did not involve any human subjects. All the data analyzed in this work was collected from the publicly available data in the Play Store.

2 The Problem and Baseline Solution

2.1 Problem Statement

Our objectives are two-fold: (1) we conducted a large-scale measurement study to examine the (in)consistencies between the privacy practices implemented in IoT apps and the privacy disclosures released in privacy policies. In particular, we focus on answering key questions about the current state of IoT privacy policy usage, compliance gaps, and potential compliance patterns, e.g., *how many apps provide available privacy policies to the users, what type of private data is transmitted to first/third-party entities, which of the practices are disclosed in*

[1] https://github.com/IoTPrivComp.

privacy policies? (2) Since existing OTS tools are unable to provide satisfactory performance for IoT apps, we propose the IoTPrivComp framework to perform app code analysis, privacy policy analysis, and compliance gap analysis.

In privacy practices and disclosures, we focus on information *collection* and *sharing*. "Collection" means that certain (private) data is accessed by the app and transmitted out of the app's memory space to a first-party entity, e.g., the app's cloud server, while "sharing" takes place when the data is transmitted to a third party, e.g., the app sends payment information to PayPal. The collection and sharing practices take place as a result of certain API executions [5,51]. In both cases, the data accessed is transferred externally, either outside the app or out of the device. We refer to these leaks as *external data flows*. Some of the external data flow scenarios that we have identified include sending the data to an external server, sending the text messages and emails, sending the log data outside the app, and sharing data with another app using exported components.

The problem of IoT privacy compliance analysis is challenging. Data leaks are usually found by analyzing the APIs, permissions, and protected resources [16, 19,52], but identifying data sources and sinks is not straightforward as many permission-protected methods are not source nor sink [34]. Moreover, like other static flow analyzers for mobile apps, we assume the code is not obfuscated. Privacy policy analysis approaches usually rely on a hand-annotated corpus [42, 51,52] but they are limited due to the manual effort. It is also difficult to identify contradictory statements in the policies [3]. NLP still has limited capabilities in analyzing statements that span multiple sentences and that use confusing language [4].

Finally, we would like to note that our focus in this work is not on detecting unknown privacy threats or providing security evaluations. This aspect of IoT app security has been extensively studied in the literature. Instead, we aim to understand the landscape of IoT app privacy compliance. While our findings can be used to improve future privacy policies, we consider privacy policy comprehension or enforcement out of the scope of this work.

2.2 The Baseline Solution Using Off-the-Shelf Tools

Privacy policy compliance in mobile applications has been extensively studied. As shown in Fig. 1-B, it consists of three main tasks, i.e., *app flow analysis* (Ⓒ and Ⓓ), *privacy policy analysis* (Ⓖ and Ⓗ), and *flow-to-policy consistency analysis* (Ⓘ). To avoid re-inventing the wheel, we examined several existing tools and tested their effectiveness in identifying privacy gaps in IoT apps.

Tools and Implementation. First, we studied PoliCheck [4], which implements an automated, entity-sensitive privacy policy consistency analysis for mobile apps. It employed AppCensus [16] for data flow analysis and PolicyLint [3] for privacy policy analysis. Unfortunately, AppCensus was commercialized and unavailable to the research community. Hence, we replaced the data flow analysis module of PoliCheck with two other open-source tools, i.e., SuSi [34] and AndroShield [2]. SuSi was used to identify Android source and sink methods,

while AndroShield was used to extract the paths between the identified sources and sinks. In this baseline approach, we modified the interface to feed flow data from AndroShield to PoliCheck and also implemented the necessary interfaces to assemble all the off-the-shelf tools together.

Table 1. Sample IoT devices from four popular IoT platforms.

IFTTT	Ai-Sync, Iotics, Lexi, LIFX, AirTouch, Arlo, Neato, Neo Smart
SmartThings	Ring, Belkin, Leviton, Yeelight, Blaze, Awair, Danalock, Connected
OpenHAB	Netatmo, BenQ, Nest, Nanaleaf, Ecobee, Nuki, Onkyo, OpenGarage
Zapier	Phillips Hue, Luxafor, Flic, Kisi, bttn, Amazon Alexa, Tap NFC

Evaluation and Results. We extracted external data flows from 68 IoT apps and manually verified them. Then, we randomly selected 100 external data flows for evaluation. The baseline approach discovered only 64 external data flows and correctly reported only 29 privacy disclosures including 2 clear disclosures and 27 omitted disclosures (see Sect. 3.4 for privacy disclosure definitions). Therefore, it achieved an overall accuracy of 29%. In comparison, IoTPrivComp identified all the external data flows and correctly reported 94 (consistent and inconsistent) privacy disclosures, achieving a 94% overall accuracy. IoTPrivComp failed in 6 cases because the corresponding privacy policies did not include clear statements about their privacy practices.

The baseline's low performance may be caused by three issues. First, SuSi and AndroShield were implemented over Android 4.2, which cannot correctly handle new Android APIs (e.g., Android 29) and IoT-specific data flows. Second, SuSi and PoliCheck adopted conventional machine learning models with limited performance for classification and NLP tasks. Finally, PolicyLint used spaCy's NER engine (`en_core_web_lg model`) for entity and data object identification. Its outdated ontology cannot correctly handle the IoT-specific terminologies in the privacy policies. In recognizing the root causes of the low performance, we propose to revamp the baseline approach by tackling these three issues.

3 IoTPrivComp: Privacy Compliance for IoT Apps

In this section, we present IoTPrivComp, an automated privacy compliance verification framework for IoT apps. As shown in Fig. 1-B (our primary contributions are highlighted), the IoTPrivComp framework consists of four key components: data collection (Sect. 3.1), ontology definition (Sect. 3.2), data flow analysis (Sect. 3.3), privacy policy analysis, and compliance validation (Sect. 3.4).

3.1 Data Collection

There is no clear or authoritative definition for IoT apps. In this work, we consider all the mobile applications that control, manage, and/or interact with IoT devices as *IoT apps*. According to [25], IoT devices are low-cost devices with

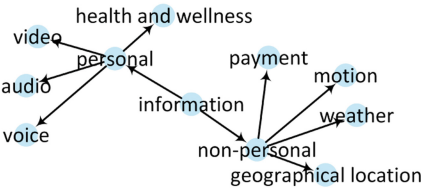
sensors and/or actuators that generate sensing data and offer various services to their users. Therefore, smartphones, laptops, and PCs are controller devices that interact with IoT devices but they are not considered IoT devices themselves.

IoT Apps. To recognize the loosely-defined IoT apps from the Play Store, we considered the popular IoT platforms studied in the literature [8,20,24,33,36,48], e.g., IFTTT (If-This-Then-That), SmartThings, openHAB, and Zapier, and identified IoT devices from each platform, as shown in Table 1. We also added wearable devices that directly connect to smartphones using WiFi/Bluetooth to this seed set. Next, we wrote a Scrapy script to collect the most relevant Play Store apps for the seed devices and identified 543 unique app manufacturers from the apps, for which we further scraped all their free IoT apps. Finally, we employed a pattern matching approach to identify and remove any non-IoT apps based on their names and descriptions. Our final dataset has 1,951 IoT apps. We downloaded the original APK files of 1,915 IoT apps from the Google Play Store, where 36 APKs failed to download.

```
Entity ontologies:
('payment-providers', 'PayPal')
('social-media', 'Twitter')
('social-media', 'Facebook')

Data ontologies:
('personal', 'voice')
('personal', 'health and wellness')
('non-personal', 'geographical location')
('non-personal', 'payment')
```

(a)



(b)

Fig. 2. (a) Examples of entity and data ontologies; (b) A part of data ontology graph.

Privacy Policies. For each crawled app, we followed the embedded link to retrieve its full policy page including the dynamically loaded contents, and then converted it into a text file. Among 1,951 IoT apps, 234 apps did not have an available privacy policy due to missing or broken links and 228 apps had non-English policies (discussed in Sect. 4). Finally, we obtained the privacy policies of 1,489 apps and used them in our privacy compliance analysis.

3.2 Ontology Definition for IoT App Privacy Policies

Ontologies are used to represent "is-a" relationships among the terms in privacy policies, which enable a semantic analysis of privacy policies. Since there is no ontology for IoT privacy policies, we created our own in this work by manually annotating 134 policies with *entity objects* and *data objects* and their subsumptive ("is-a") relationships and splitting the annotated data for training, validation, and testing. We trained a Tok2Vec relation classifier with 81% precision. Next, we extracted entity/data objects with a BERT model that is fine-tuned on a privacy policy corpus. BERT was introduced in [14] to fine-tune

the pre-trained models for accomplishing various tasks without changing the architecture significantly for each task. Then, we applied the Tok2Vec classifier to the extracted objects and retained only the predictions with a confidence score of 90% or higher. For example, for the below sentence, our model generated the relation as `('information', 'location')`:

```
We collect information about you, including location.
```

We further extracted two types of ontology graphs from the relationships, *entity ontologies*, and *data ontologies*, which represent the relationships between entities and data objects, respectively. Figure 2(a) shows a few examples of the extracted data and entity ontologies, while Fig. 2(b) shows a subset of the data ontology graph. A few more ontology examples can be found in Appendix A. Using the relationships found from ontology extraction, we also identified the synonyms, i.e., functionally equivalent/similar entities and data objects. For example, `payment transaction` and `payment processing` are identified as synonyms of `payment`. Finally, the ontologies were fed into PoliCheck-BERT-IoT.

3.3 SuSi-MNB-IoT: Analyzing Sinks and Data Flows in IoT Apps

To extract the app code, we reverse-engineered the downloaded APK files. In particular, we obtained the DEX (Dalvik Executable) files from the APKs using Apkanalyzer [39] and converted them to Jimple files using the Soot program analysis framework [38]. Then, we extracted the manifest files in binary format and converted them into XML. The extracted Android code is analyzed to identify *sources* that are associated with the collection of sensitive data, *sinks* that transmit that data external to app/device, and *data flows* from the sources to the sinks. Manually classifying the sources and sinks is a costly task due to a large number of supported methods. To tackle the challenge, automated code analyzers such as SuSi [34] and AndroShield [2] proposed machine-learning-based approaches to conduct flow analysis in three steps, i.e., source and sink identification, data flow tracing, and sink categorization. However, our baseline study showed that 71% of flows were missed or incorrectly reported when directly applying Susi for IoT app analysis. Inspired by Susi, we developed a code analysis module, called SuSi-MNB-IoT, which introduced non-trivial modifications as follows to improve sink and data flow identification for Android code analysis.

Customization for IoT Devices/Apps. IoT apps collect new types of data that are rarely accessed by general mobile apps. Therefore, existing code analyzers fall short in identifying these sensitive data. In SuSi-MNB-IoT, we defined several new sink categories pertaining to sensitive data types, including `Geolocation`, `Health_wellness`, `Motion`, `Socialmedia_activity`, `Music`, `Payment`, and `Video`. A detailed list of sink categories is shown in Table 2.

Advanced Machine Learning Approach. The Support Vector Machine (SVM) model performs poorly when the training data is small [45]. To improve the performance in text snippet classification, we built a Multinomial Naive

Bayesian (MNB) classifier in SuSi-MNB-IoT. MNB is a probabilistic classifier based on Bayes' theorem that predicts using previous and current knowledge [15]. Compared with SVM, our MNB model is two to six times faster [28].

Adding Support for Android 29. As most of the IoT apps in our dataset use Android 29, we added a mechanism to handle the new API methods and identified the new sinks from Android 29 API methods.

Adding Support for Exported Components. Exported components in Android facilitate permissioned data sharing *between apps*. In SuSi-MNB-IoT, we included exported components along with other data flows.

With the sources and sinks identified by SuSi-MNB-IoT, we traced data flows and identified data leaks through Android API sinks. In particular, we followed the AndroShield approach [2], which is based on FlowDroid [5], to construct call graphs. The APIs and methods involved in sensitive data flows are identified as nodes of the call graph. Then, we extracted tainted paths from the sources to the sinks by traversing the call graph with a Depth-First Search (DFS) algorithm. Sensitive data travels through these paths and is finally sent out through the sinks. Next, we manually annotated 2,450 data flows (1,960 for training, 490 for testing) across various sink categories. Using this dataset, we finally trained an MNB classifier for automated app analysis, which categorizes data flows based on their classes and sink methods.

Table 2. Sink categories and SuSi-MNB-IoT's classification performance. R: Recall (%); P: Precision (%).

Category	R	P	Category	R	P	Category	R	P	Category	R	P
AAID	100	98	Audio	100	88	Calendar	100	98	Camera	81	94
Email	100	95	Gallery	96	96	Geolocation	97	92	Health_Wellness	92	97
Motion	100	98	Music	100	97	NFC	97	97	Payment	100	95
Phone	92	98	Router	100	95	SIMID	100	100	Socialmedia_Activity	96	94
SMS	100	100	Sound	100	97	Video	69	98	Weather	100	95
Voice	98	92	Weighted Average: Recall: 96%; Precision: 96%								

3.4 PoliCheck-BERT-IoT: Policy and Consistency Analysis

To detect the inconsistencies between apps' data flows and the disclosed privacy practices, we developed a new policy analyzer, called PoliCheck-BERT-IoT, which followed the PoliCheck approach [4] originally developed for mobile privacy policy analysis. Compared with PoliCheck, PoliCheck-BERT-IoT introduced two improvements to capture IoT-specific data practice statements.

IoT-Specific Ontology. PoliCheck uses PolicyLint [3] to identify entities and objects in privacy policies. To process IoT policies, PoliCheck-BERT-IoT extended PolicyLint to capture the IoT-specific ontologies developed in Sect. 3.2. It can recognize the synonyms for entities and data objects and the IoT-specific relationship mappings. With domain adaptation, the improved PolicyLint module

achieved an 89.6% precision and a 73.3% recall in identifying data objects, and an 88.5% precision and a 69.5% recall in identifying entities, respectively.

Adapting State-of-the-Art NLP Model. PoliCheck/PolicyLint uses spaCy's NER engine with the `en_core_web_lg` model, which is based on Convolutional Neural Networks (CNN). To improve NER performance, we replaced the CNN-based model with a transformer-based BERT model. Transformers enable downstream tasks to fine-tune a pre-trained model to a specific domain without incurring the resource-intensive training process of complex models [47].

Based on the policy analysis results and the sensitive data flows obtained in Sect. 3.3, PoliCheck-BERT-IoT performs a consistency analysis. It extracts sentences about the app's data practice from its privacy policy. Each statement is then matched with the identified sensitive data flows to determine the type of privacy disclosures. Data flows of the same data type and the same root domain are combined to output unique data flows. For example, two flows `<com.samsung.auth, music.activity.SoundPickerActivity>` and `<com.samsung.report, music.activity.SoundPlayerActivity>` are considered the same flow in consistency analysis since they have the same data type (i.e., music) and the same root domain.

Table 3. Privacy policies of IoT apps under study: (left) policy availability; (right) External Data Flows (EDF).

App category	# of apps	% of apps	App category	# of apps	% of apps
All crawled apps	1,951	100	Apps w. sensitive EDFs	623	100
Missing policy	234	12.0	Missing policy	33	5.3
Non-English policy	228	11.7	Non-English policy	58	9.3
Available policy	1,489	76.3	Available policy	532	85.4

PoliCheck-BERT-IoT identifies five types of privacy disclosures: **(i) clear disclosures**, in which the privacy policy precisely states that the data is being disclosed to the entity involved in the flow. **(ii) vague disclosures**, in which the privacy policies use vague or broad terms to describe data types and/or entities, e.g., stating that the app "collects your data" instead of "collects your fingerprint and voice data", or the app "shares data with social networks", instead of "shares data with Facebook and Twitter". **(iii) omitted disclosures**, where the privacy policy fails to disclose the data flow, e.g., sharing data with Facebook without mentioning it in the privacy policy. **(iv) incorrect disclosures**, in which the privacy policy statement incorrectly states that the practice will not take place, e.g., collecting camera information while the privacy policy states not collecting such data. And **(v) contradictory disclosures**, where the flow matches more than one privacy statement and the statements contradict each other. We consider the privacy practice and disclosure as *consistent* in case of clear and vague disclosures, whereas, *inconsistent* disclosures refer to omitted, incorrect and contradictory cases. Moreover, when the entity names match with

the app package names or a part of the privacy policy link, the flow is considered *first-party*. Otherwise, the flow is considered *third-party*.

4 Evaluation and Analysis

In this section, we first evaluate the performance of IoTPrivComp in identifying the inconsistencies between privacy disclosures and privacy practices and then employ IoTPrivComp to measure the privacy compliance status of the IoT apps and answer the research questions presented in Sect. 1.

4.1 Performance Evaluation of IoTPrivComp

We first evaluated the performance of the key components of IoTPrivComp. In particular, SuSi-MNB-IoT achieved an average precision and recall of 96% for sensitive sink identification, as shown in Table 2. For PoliCheck-BERT-IoT, we manually annotated 50 privacy policies of IoT apps. We extracted the dictionary of annotations and applied them to the large corpus of 2,050 policies (1,640 policies for fine-tuning and 410 policies for validation). The training process took 2 h on an NVIDIA Tesla P100 GPU (PCI-E 16GB). The final model achieved an 87.94% precision and an 88.09% recall for identifying data objects from privacy policies, and a 90.89% precision and a 91.05% recall for identifying entities. The performance is significantly improved over the CNN-based model.

Table 4. Number of flows and apps associated with different privacy disclosure types.

Privacy disclosures		IoT 1st-party		IoT 3rd-party		Wearable 1st-party		Wearable 3rd-party	
		Flows	Apps	Flows	Apps	Flows	Apps	Flows	Apps
Consistent	Clear	12	12	0	0	9	8	0	0
	Vague	92	75	72	63	22	19	28	21
Inconsistent	Omitted	171	136	253	203	52	46	98	61
	Incorrect	1	1	1	1	0	0	0	0
	Contradictory	1	1	4	3	0	0	0	0
Total #		277	225	330	270	83	73	126	82
Total Inconsistent		173	138	258	207	52	46	98	61
Inconsistent rate (%)		62.5	61.3	78.2	76.7	62.7	63.0	77.8	74.4

Next, we validated the overall performance of IoTPrivComp by sampling 68 IoT apps and manually analyzing their data flows. We read the corresponding privacy policies to verify the disclosure types and consistency results reported by IoTPrivComp. If IoTPrivComp extracts a data flow and classifies the corresponding policy statement correctly, the result is marked as true positive. In particular, IoTPrivComp reported 100 sensitive data flows and labeled 18, 38, and 44 flows as "clear disclosures", "vague disclosures", and "omitted disclosures", respectively.

There were no incorrect and contradictory disclosures, as they are very rare. We found that 94 out of the 100 reported flows were true positives, indicating an overall accuracy of 94%. Moreover, all the "clear disclosures" were correctly reported, while 5 "vague disclosures" and one "omitted disclosure" were incorrect. The discrepancies occur because of the confusing language of privacy policies that do not state clearly the collection and sharing practices.

4.2 Policy and Data Flow Analysis

In this study, we identified a total of 1,951 unique IoT apps and retrieved 1,489 privacy policies written in English and 1,825 APKs, where 36 APKs failed to download and 90 apps failed during static analysis.

Missing Privacy Policies. As shown in Table 3, 234 (12%) apps did not have available privacy policies. Among them, 188 apps did not provide any policy URL, while 46 apps provided invalid URLs. The number of apps with missing policies reported in this study was non-trivial, as these apps may have potentially undisclosed data leakages. The result highlights the need for strict and continuous enforcement of regulations. Meanwhile, there were 228 (11.7%) apps with non-English privacy policies. Among them, 160 had app descriptions in English, which indicates that they were intended for English-speaking users but their privacy policies fell short in disclosing the app practices to the users.

Data Flows and Sinks. IoTPrivComp extracted a total of 23,959 *external data flows* from 1,825 APKs, from which information flows out of the device or the app's memory space through first- or third-party code. Among them, 1,782 external data flows disclosed *sensitive information* of 21 categories defined in Table 2. These sensitive data flows involved 623 IoT apps and 1,075 distinct Android APIs. `com.facebook`, `com.samsung`, and `com.amazon` are the most frequently used APIs, which appeared in multiple flow categories, e.g., payment, social media activity, voice, and video. Finally, we found 33 apps with missing policies and 58 apps with non-English policies, which is 14.6% of all apps that disclose sensitive data.

Table 5. Privacy disclosures of IoT-specific practices.

Privacy disclosures	Clear	Vague	Omitted	Total
# of flows	10	37 (26.4%)	93 (66.4%)	140
# of apps	10	37 (27.2%)	89 (65.4%)	136

4.3 IoT Privacy Compliance Analysis

Next, we conducted a privacy compliance analysis at the flow level and app level of 532 apps with available privacy policies. As wearable devices collect

more sensitive data such as biometrics and physical activities than general IoT devices, we reported the results for wearable apps separately.

Flow-Level Compliance Analysis. IoTPrivComp extracted 6,823 sentences with the positive or negative sentiment about the apps' data practices and associated them with 816 unique data flows. Each flow has a unique data type and disclosure type. The results of the flow-level compliance analysis are summarized in Table 4. For *IoT apps*, 173 (62.5%) first-party data flows and 258 (78.2%) third-party flows were reported with inconsistent privacy disclosures, where most of them had "omitted disclosures", indicating a direct compliance violation in apps' data practices. Similarly, for wearable apps, 52 (62.7%) and 98 (77.8%) inconsistent disclosures were detected in the first-party and third-party flows, respectively. Overall, a total of 581 (i.e., 173+258+52+98) inconsistent disclosures were reported, among which 574 (i.e., 171+253+52+98) were omitted disclosures.

IoT apps often collect personal data that are rarely accessed by conventional mobile apps. For example, among the 21 data categories defined in Table 2, `health and wellness`, `motion`, and `voice` data are often collected by IoT apps. Therefore, we further analyzed the flow-level compliance gaps specifically in the IoT data practices. We extracted data flows to these three sink categories and measured the number of flows with clear, vague, and omitted disclosures. As shown in Table 5, 92.8% of IoT-specific data flows had vague or inconsistent disclosures (26.4% vague and 66.4% omitted disclosures).

App-Level Compliance Analysis. 816 unique flows were associated with 411 IoT apps and 121 wearable apps. Table 4 summarizes apps with different types of privacy disclosures. It is worth noting that an app may be counted more than once if it has multiple flows with different types of disclosures. To understand the disclosure behavior of individual apps, we further calculate the number of apps with at least one disclosure of each type. As shown in Table 6, only 12 (2.9%) of the 411 IoT apps clearly disclosed their data collection practices, while 123 (30%) IoT apps disclosed the data practices in vague terms. The majority of the apps (74.5%) failed to disclose their privacy practices (omitted disclosures). The situation is worse for wearable apps. Only 8 (6.6%) apps clearly disclosed the data practices, while 36 (29.8%) apps disclosed the practices in vague terms and 96 (79.3%) apps did not disclose the practice at all.

Table 6. Apps' privacy compliance; TP: Third Party, SMA: Social Media Activity.

App category	# of IoT	# of Wearable	Total #	% of apps
Apps analyzed for privacy disclosures	411	121	532	100
At least one clear disclosure	12	8	20	3.8
At least one vague disclosure	123	36	159	30.0
At least one omitted disclosure	306	96	402	75.6
At least one incorrect disclosure	2	0	2	0.4
At least one contradictory disclosure	4	0	4	0.8
At least one inconsistent disclosure	312	96	408	76.7
At least one omitted TP disclosure	203	61	264	49.6
Omitted disclosure about TP SMA	45	19	64	12.0

In summary, we have the following observations regarding the privacy disclosures and privacy compliance gaps in IoT and wearable apps.

(1) *Very few IoT apps clearly disclosed their data collection practices.* Among all the apps analyzed in this work, only 3.8% clearly stated their practices of first-party data collection, while none of them clearly disclosed third-party data sharing actions. Figure 3(a) shows the breakdown of clear disclosures across different data types. Most of the clear disclosures belong to the `health and wellness` category, but only 8 out of 101 `health and wellness` flows were disclosed.

(2) *30% of IoT apps had vague disclosures in their privacy policies.* They often use vague language or broad terms to describe data types (e.g., "your data" instead of "voice data") or the third-party entities (e.g., "social networks", "platforms" "service providers", "vendors", "contractors", and "sponsors"). Such disclosures are considered consistent from the compliance perspective, however, the practice is concerning because agreeing to the blanket policies puts the users in a very vulnerable situation. Figure 3(b) shows vague disclosures of various data types. `Health and wellness` and `social media information` are the most common data types with vague disclosure in first- and third-party data access. Moreover, 3.7% and 10.3% of vague disclosure flows are associated with voice data and payment information, which are sensitive in IoT applications.

(3) *76.7% of the analyzed apps had at least one inconsistent data collection or sharing practice.* That is, they collected or shared at least one sensitive data item that was incorrectly or contradictorily disclosed, or not disclosed at all.

(4) *75.6% of the apps contained at least one undisclosed sensitive data collection or sharing practices.* The most common type of inconsistent disclosure is *omitted*, where the privacy policies did *not* mention the data collection and data sharing practice at all. As compared to the first-party data collection practices, it is more common for the app privacy policies to not disclose the third-party data sharing practices, as shown in Fig. 3(c).

(5) *35.5% of all the flows with omitted disclosure involved personal data* including `audio`, `photo`, `health and wellness`, `video`, and `voice` data. Meanwhile, *12.0% of the apps shared social media information with third-party platforms without disclosing the practice.* There were only five social media omitted disclosure flows for the first party but 65 omitted disclosures for the third party.

(6) *49.6% of the apps had at least one data sharing practice with third parties that were not disclosed in privacy policies.* The third-party omitted flows made up a surprisingly high (43.0%) percentage of all the flows. Incorrect and contradictory disclosures were less frequent, as shown in Fig. 3(d) and (e).

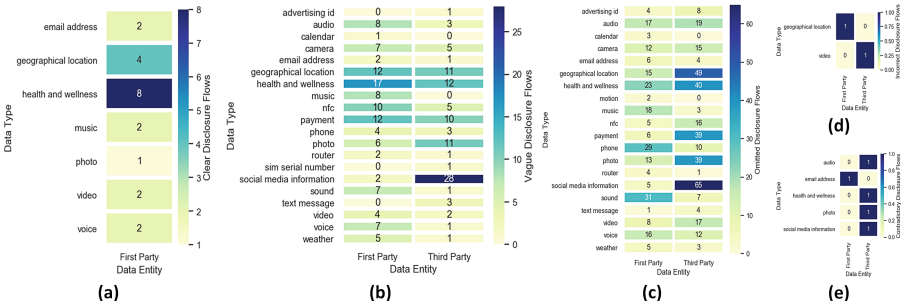


Fig. 3. Statistics of flows for (a) clear disclosures, (b) vague disclosures, (c) omitted disclosures, (d) incorrect disclosures, and (e) contradictory disclosures.

5 Case Studies

We present our case studies of real-world apps and use them as examples to demonstrate the capability of IoTPrivComp in our measurement study. In particular, we examined 2 wearable apps ("Your Fitness" and "Fitness Tracker by Echoronics") and 3 IoT apps ("Ahome Light", "Hager Coviva", and "My Leviton"), Two of them ("Your Fitness" and "Ahome Light") were selected from the manual evaluation set (in Sect. 2.2). In Table 7, we list the privacy policy and the APIs used by each app to collect/share personal data. IoTPrivComp reported different types of vague and/or inconsistent privacy disclosures for each app, which help to raise awareness among users and provide useful information for developers and regulators to identify and address the compliance issues.

Case 1: Your Fitness (`com.yc.yourfitness`) works with smart-bracelet devices to manage daily exercise and sleep activities. IoTPrivComp identified multiple APIs in the code that collected data about the steps taken by the user. However, the privacy policy only vaguely mentioned the collection and disclosure of "personal information", for which IoTPrivComp identified as a *vague disclosure*. It also used the `com.baidu.location` API for the third-party access of the location data but did not disclose this practice in the privacy policy at all. Therefore, it was reported by IoTPrivComp as an *omitted disclosure*.

Case 2: Fitness Tracker by Echoronics (`com.mevofit.fitness.fitness tracker.walkingjogginghrbp.echotrackers`) manages Echoronics wearable devices and tracks fitness-related information. Its privacy policy identifies specific types of personal data collected by the app, however, it does not mention ECG (e.g., heart rate, rhythm, etc.) data and geolocation data. However, IoTPrivComp identified three APIs that were used for third-party sharing of ECG and geolocation data, and therefore reported corresponding omitted disclosures.

Case 3: Hager Coviva (`com.hager.koala.android`) is used with a home controller to monitor and control alarms, lights, and shutters that are deployed in smart homes. Its privacy policy describes multiple types of personal data collected by the Coviva controller, but does not mention any personal data collected

Table 7. Case studies: apps' privacy policies and APIs for data access/sharing.

Case 1	Privacy policy	"Your Fitness will disclose all or part of your personal information in accordance with your personal wishes or legal provisions in the following circumstances: To provide the products and services you require, you must share your personal information with third parties"
	APIs	`com.yc.pedometer.MainActivity, com.yc.pedometer.SplashActivity, com.yc.pedometer.service.MessageAccessibilityService, com.yc.pedometer.wxapi.WXEntryActivity, com.yc.pedometer.service.StatusbarMsgNotificationListener`
Case 2	Privacy policy	"When you use our services, the personal data that is collected includes - your email address, name, gender, age, height, and weight. Depending on your use of application we may collect data like calorie intake, weight loss goal, lifestyle, and body measurements.", "Personal information about you, such as your gender, birthday, zip code, country, height, weight, lifestyle and exercise frequency"
	APIs	`com.ecgview.EcgReportActivity, com.ecgview.EcgReportView, com.gpstracker.GPSTrackerSummeryActivity`
Case 3	Privacy policy	"Data which you make available to Hager: When you register, you provide us with certain data, to be specific your name and email address.", "Data which is automatically collected and saved in the system: Every time you log in, you use and we collect your IP source address to allow the Coviva controller to communicate with your mobile device.", "Data which is collected during installation: Further information on the installer, product-related information (serial number and MAC address), and the status of the installation ('system consigned/not consigned to customer') is recorded"
	APIs	`com.hager.koala.android.activitys.motiondetector.ImageViewerForHistoryMotionDetectorScreen, com.hager.koala.android.activitys.motiondetector.ImageViewerMotionDetectorS, com.hager.koala.android.activitys.motiondetector.LastIntrusionsMotionDetector, com.hager.koala.android.activitys.motiondetector.UpdateInovaMotionDetectorS`
Case 4	Privacy policy	"Specifically, the App and the related Product(s) have collected the following categories of Personal Information ("PI") from its consumers, as defined by the California Privacy Laws: Genetic, physiological, behavioral, and biological characteristics, or activity patterns used to extract a template or other identifier or identifying information, such as fingerprints, faceprints, and voiceprints, iris or retina scans, keystroke, gait, or other physical patterns, and sleep, health, or exercise data."
	APIs	`de.niklasmerz.cordova.biometric.BiometricActivity`
Case 5	Privacy policy	"The information and materials about you collected by this application will be stored on the server of this application and/or its affiliates."
	APIs	`wl.smartled.service.AudioRecorderService`

by its sensors. In fact, IoTPrivComp identified 4 APIs that collected the motion sensor data, which is considered private. IoTPrivComp reported several *omitted disclosures* of the app. Meanwhile, from the app descriptions, we did not notice any functionality associated with user tracking. However, from the data flows, the app appears to trace the users using its own APIs, which could be a serious privacy violation that warrants further investigation.

Case 4: My Leviton (`com.leviton.home`) manages Leviton's Decora smart Wi-Fi devices, such as dimmers, switches, and smart plugs. Its privacy policy mentions the collection of biometric data by the first party, while IoTPrivComp reported that this app also shared the collected biometric data with a third-party `niklasmerz` using the `de.niklasmerz.cordova.biometric.BiometricActivity` API. We researched the third party and found the cordova plugin that works with the biometric sensor data. This case is an example of the *omitted* disclosure of sensitive personal information.

Case 5: Ahome Light (`wl.smartled.rgb.ahomelight`) allows users to control smart LED lights. IoTPrivComp found that the app collects audio data (through

the microphone) using the `wl.smartled.service.AudioRecorderService` API. However, this practice is not disclosed in its privacy policy. Instead, it makes very broad references to the collected data as the "information and materials about you", which vaguely covers the voice data. This was reported by IoTPrivComp as a *vague* disclosure. In fact, collecting or sharing the audio data without properly disclosing it in the privacy policy is quite concerning.

6 Discussions and Future Work

With IoTPrivComp, we conducted a measurement study over 1,489 IoT applications and discovered several types of compliance issues in a significant number of IoT apps, whose privacy practices (such as collecting and sharing of private data) are not properly disclosed in their privacy policies. Our results help to answer the research questions raised in Sect. 1.

1. *What does the landscape of IoT app privacy compliance look like?*
Answer: Our literature review shows that little effort has been devoted to IoT privacy compliance issues in the research community. Our results show that even with the privacy regulations in place, a significant gap still exists between the apps' privacy practices and their disclosures of such practices to the users.
2. *Which privacy compliance gaps exist in IoT apps?*
Answer: The compliance gaps include policies that are unavailable or difficult to comprehend and inconsistent disclosures. For instance, 12% of 1,951 apps do not have any privacy policies, while 8.2% of them have English app descriptions but non-English privacy policies. 75.6% (402/532) of the analyzed apps have omitted disclosures for sensitive data flows, while none of the third-party sensitive data flows is clearly disclosed.
3. *Does there exist any patterns in privacy compliance gaps?*
Answer: Some patterns could be observed from the identified compliance gaps. For instance, the vast majority (574 out of 581) of inconsistent disclosures are omitted disclosures, while incorrect and contradictory disclosures are very rare. Certain types of data are more frequently involved in undisclosed data collection/sharing than others. While one may expect the developers to be more cautious in properly disclosing the practice with more sensitive data, we do not observe such a pattern. In statistics, 237 (40.8%) of all the 581 inconsistent flows are related to personal data such as audio, email address, health and wellness, video, voice, and social media information, while 63.4% of the 101 health and wellness data flows are inconsistent with disclosure.

Our findings call for stricter control from regulations regarding the violations of sensitive data leaks. In particular, there should be regulations enforcing controls to address the leak of Protected Health Information (PHI) and Personally Identifiable Information (PII) data from the IoT apps. Therefore, IoTPrivComp can be used as a policy compliance verification tool to automatically check if an IoT application's data practice follows its privacy policy. It can

help app users, app markets, and regulators efficiently detect privacy violations. Meanwhile, app developers could leverage it to identify unintended data use or inappropriate privacy policies.

For future work, we recognize that different types of privacy information pose different levels of risk. For instance, sharing weather data is significantly less risky than sharing heath-related sensor data. Therefore, we will consider the risk levels and generate comprehensive compliance and risk profiles for IoT apps.

7 Related Work

Our work is related to three research directions in the literature, i.e., IoT app security, app code analysis, and privacy policy analysis.

IoT App Security and Privacy. Most of the existing work focuses on identifying security vulnerabilities in IoT applications. For example, Celik et al. discovered privacy leaks in IoT apps [10] and proposed mechanisms to verify or enforce security policies [11,12]. Another research direction is to discover side-channel privacy leaks [41,44]. [9] identified privacy leakage in SmartThings apps and [6] proposed to alert the user when the privacy preferences are violated. Finally, some recent work proposed to capture IoT traffic to validate compliance of data disclosure to the privacy policy [40] or check IoT app descriptions against the data practices described in privacy policies [26]. While they are related to our work, they either took manual analysis approaches or focused on one aspect of private information (e.g., 11 apps in [40] and [26] studied voice assistants only).

App Code Analysis. Code analysis has been widely used to study Android app permissions, such as mapping API calls to permissions to analyze the access control models [7,50] and identify the overprivileged apps [1,18], locating potential data leaks by analyzing the APIs, permissions, and protected resources [19,52], etc. SuSi [34] proposed a machine learning approach to identify the sources of sensitive data and sinks. Code analysis tools (e.g., AndroShield [2]) constructed tainted paths from the identified sources to sinks. Recently, AppCensus [16] and Han [21] proposed to identify sensitive data flows based on sensitive resources protected by permissions. These app code analysis approaches adopt static [32,51,52], dynamic [30], and hybrid analysis [16].

Privacy Policy Analysis. Existing works on privacy policy analysis such as MAPS [51], PolicyLint [3] and PoliCheck [4] focused mainly on mobile applications. IoTPrivComp is among the first to study privacy compliance gaps in IoT applications. As reported in [3], prior approaches using bigrams [51] or regular expressions [31] for policy analysis struggled with the accurate detection of negative statements. Similar to PolicyLint [3], IoTPrivComp uses sentence-level NLP and ontologies to detect negations in complex sentences. Finally, previous works mostly rely on hand-annotated corpora datasets and rules [22,42,51,52], which have limited coverage and scalability. IoTPrivComp leverages state-of-the-art machine learning methods to automatically annotate data and entity objects in a large corpus of 2,050 privacy policies.

8 Conclusion

In this paper, we present a large-scale measurement study on the privacy compliance of IoT apps. To conduct this measurement study, we first develop IoTPrivComp, which analyzes the code and privacy policies of IoT apps to find compliance gaps between the actual and declared data practices. The IoTPrivComp framework consists of a new ontology for IoT app privacy policies, a new sink identification module SuSi-MNB-IoT, a data flow analysis module, and a new consistency analysis module PoliCheck-BERT-IoT for detecting inconsistent privacy disclosures.

Using IoTPrivComp, we found that a vast majority of the analyzed apps had data practices that were inconsistent with their privacy disclosures. The most common inconsistencies are the omitted disclosures where the privacy policy does not mention the privacy practice. Despite the privacy regulations in place, we still found significant compliance gaps. Our results show that there is a strong need for strict regulations that are thoroughly enforced in the app stores.

Acknowledgements. The authors were sponsored in part by NSF IIS-2014552, DGE-1565570, DGE-1922649, and the Ripple University Blockchain Research Initiative. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

A Ontologies

We have 121 entity ontology pairs, 52 data ontology pairs, and 7,592 synonyms in the IoT-specific ontology. Table 8 shows parts of the data and entity ontologies.

Table 8. Examples from the Data and Entity Ontologies.

Data ontology
(‘information’, ‘personal’), (‘information’, ‘non-personal’), (‘personal’, ‘email address’), (‘personal’, ‘account user info’), (‘personal’, ‘phone’), (‘personal’, ‘address’) (‘personal’, ‘voice’), (‘personal’, ‘photo’), (‘personal’, ‘social media information’), (‘personal’, ‘audio’), (‘personal’, ‘video’), (‘personal’, ‘account details’), (‘personal’, ‘health and wellness’), (‘non-personal’, ‘music’), (‘non-personal’, ‘router’), (‘non-personal’, ‘sound’), (‘non-personal’, ‘payment’), (‘non-personal’, ‘motion’), (‘non-personal’, ‘geographical location’), (‘non-personal’, ‘user patterns and usage’), (‘non-personal’, ‘weather’), (‘non-personal’, ‘calendar’), (‘non-personal’, ‘camera’), (‘non-personal’, ‘organization info’), (‘non-personal’, ‘sim serial number’) (‘non-personal’, ‘device info’), (‘non-personal’, ‘nfc’), (‘non-personal’, ‘text message’)
Entity ontology
(‘entity’, ‘third-party’), (‘entity’, ‘we’), (‘third-party’, ‘social-media’) (‘third-party’, ‘service-providers’), (‘third-party’, ‘payment-providers’), (‘social-media’, ‘LinkedIn’), (‘social-media’, ‘Twitter’), (‘social-media’, ‘Facebook’), (‘third-party’, ‘analytic-service’), (‘service-providers’, ‘Microsoft’), (‘payment-providers’, ‘PayPal’), (‘analytic-service’, ‘google’)

References

1. Aafer, Y., Tao, G., Huang, J., Zhang, X., Li, N.: Precise android API protection mapping derivation and reasoning. In: ACM CCS, pp. 1151–1164 (2018)
2. Amin, A., Eldessouki, A., Magdy, M.T., Abdeen, N., Hindy, H., Hegazy, I.: Androshield: automated android applications vulnerability detection, a hybrid static and dynamic analysis approach. Information **10**(10), 326 (2019)
3. Andow, B., et al.: Policylint: investigating internal privacy policy contradictions on google play. In: USENIX Security, pp. 585–602 (2019)
4. Andow, B.,et al.: Actions speak louder than words: entity-sensitive privacy policy and data flow analysis with policheck. In: USENIX Security, pp. 985–1002 (2020)
5. Arzt, S., et al.: Flowdroid: precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. ACM Sigplan. Notice. **49**(6), 259–269 (2014)
6. Babun, L., Celik, Z.B., McDaniel, P., Uluagac, A.S.: Real-time analysis of privacy-(un) aware IoT applications. Proc. Privacy Enhanc. Technol. **2021**(1), 145–166 (2021)
7. Backes, M., Bugiel, S., Derr, E., McDaniel, P., Octeau, D., Weisgerber, S.: On demystifying the android application framework: re-visiting android permission specification analysis. In: USENIX Security, pp. 1101–1118 (2016)
8. Bastys, I., Balliu, M., Sabelfeld, A.: If this then what? controlling flows in IoT apps. In: ACM CCS, pp. 1102–1119 (2018)
9. Celik, Z.B., et al.: Sensitive information tracking in commodity IoT. In: USENIX Security, pp. 1687–1704 (2018)
10. Celik, Z.B., Fernandes, E., Pauley, E., Tan, G., McDaniel, P.: Program analysis of commodity IoT applications for security and privacy: challenges and opportunities. ACM Comput. Surv. **52**(4), 1–30 (2019)
11. Celik, Z.B., McDaniel, P., Tan, G.: Soteria: automated IoT safety and security analysis. In: USENIX ATC, pp. 147–158 (2018)
12. Celik, Z.B., Tan, G., McDaniel, P.D.: Iotguard: dynamic enforcement of security and safety policy in commodity IoT. In: NDSS (2019)
13. Degeling, M., Utz, C., Lentzsch, C., Hosseini, H., Schaub, F., Holz, T.: We value your privacy... now take some cookies: measuring the gdpr's impact on web privacy. arXiv preprint arXiv:1808.05096 (2018)
14. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
15. Efron, B.: Bayes' theorem in the 21st century. Science **340**(6137), 1177–1178 (2013)
16. Egelman, S.: Taking responsibility for someone else's code: studying the privacy behaviors of mobile apps at scale. In: USENIX PEPR (2020)
17. Ermakova, T., Fabian, B., Babina, E.: Readability of privacy policies of healthcare websites. Wirtschaftsinformatik **15**, 1–15 (2015)
18. Felt, A.P., Chin, E., Hanna, S., Song, D., Wagner, D.: Android permissions demystified. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, pp. 627–638 (2011)
19. Gibler, C., Crussell, J., Erickson, J., Chen, H.: AndroidLeaks: automatically detecting potential pivacy leaks in android applications on a large scale. In: Katzenbeisser, S., Weippl, E., Camp, L.J., Volkamer, M., Reiter, M., Zhang, X. (eds.) Trust 2012. LNCS, vol. 7344, pp. 291–307. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30921-2_17

20. Gyory, N., Chuah, M.: Iotone: integrated platform for heterogeneous IoT devices. In: 2017 International Conference on Computing, Networking and Communications (ICNC), pp. 783–787. IEEE (2017)
21. Han, C., et al.: The price is (not) right: comparing privacy in free and paid apps. Proc. Privacy Enhanc. Technol. **2020**(3), 222–242 (2020)
22. Harkous, H., Fawaz, K., Lebret, R., Schaub, F., Shin, K.G., Aberer, K.: Polisis: automated analysis and presentation of privacy policies using deep learning. In: USENIX Security, pp. 531–548 (2018)
23. Hatamian, M., Serna, J., Rannenberg, K.: Revealing the unrevealed: mining smartphone users privacy perception on app markets. Comput. Secur. **83**, 332–353 (2019)
24. Jia, Y.J., et al.: Contexlot: towards providing contextual integrity to appified IoT platforms. In: 24th Annual Network and Distributed System Security Symposium, San Diego, CA (2017)
25. Kumar, A.: Internet of things for smart cities. IEEE Internet Things J. **1**(1) (2014)
26. Liao, S., Wilson, C., Cheng, L., Hu, H., Deng, H.: Measuring the effectiveness of privacy policies for voice assistant applications. In: Annual Computer Security Applications Conference, pp. 856–869 (2020)
27. Libert, T.: An automated approach to auditing disclosure of third-party data collection in website privacy policies. In: World Wide Web Conference, pp. 207–216 (2018)
28. Matwin, S., Sazonova, V.: Direct comparison between support vector machine and multinomial Naive Bayes algorithms for medical abstract classification. J. Am. Med. Inf. Assoc. **19**(5), 917–917 (2012)
29. McDonald, A.M., Cranor, L.F.: The cost of reading privacy policies. ISJLP **4**, 543 (2008)
30. Monkey. Google, inc. ui/application exerciser monkey. https://developer.android.com/tools/help/monkey.html. Accessed Aug 2021
31. Okoyomon, E., et al.: On the ridiculousness of notice and consent: contradictions in app privacy policies. In: Workshop on Technology and Consumer Protection (ConPro 2019), in Conjunction with the 39th IEEE Symposium on Security and Privacy (2019)
32. Qark. Tool to look for several security related android application vulnerabilities. https://github.com/linkedin/qark. Accessed Aug 2021
33. Rahmati, A., Fernandes, E., Jung, J., Prakash, A.: Ifttt vs. zapier: a comparative study of trigger-action programming frameworks. arXiv preprint arXiv:1709.02788 (2017)
34. Rasthofer, S., Arzt, S., Bodden, E.: A machine-learning approach for classifying and categorizing android sources and sinks. In: NDSS, vol. 14, p. 1125 (2014)
35. Rosen, S., Qian, Z., Mao, Z.M.: Appprofiler: a flexible method of exposing privacy-related behavior in android applications to end users. In: ACM CODASPY, pp. 221–232 (2013)
36. Schmeidl, F., Nazzal, B., Alalfi, M.H.: Security analysis for smart things IoT applications. In: 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft), pp. 25–29. IEEE (2019)
37. Slavin, R., et al.: Toward a framework for detecting privacy policy violations in android application code. In: Proceedings of the 38th International Conference on Software Engineering, pp. 25–36 (2016)
38. StevenArzt. Soot-a java optimization framework (2021). https://github.com/Sable/soot. Accessed Aug 2021
39. A. STUDIO. Apkanalyzer (2020). https://developer.android.com/studio/command-line/apkanalyzer. Accessed Aug 2021

40. Subahi, A., Theodorakopoulos, G.: Ensuring compliance of IoT devices with their privacy policy agreement. In: 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 100–107. IEEE (2018)
41. Subahi, A., Theodorakopoulos, G.: Detecting IoT user behavior and sensitive information in encrypted IoT-app traffic. Sensors **19**(21), 4777 (2019)
42. Tesfay, W.B., Hofmann, P., Nakamura, T., Kiyomoto, S., Serna, J.: Privacyguide: towards an implementation of the EU GDPR on internet privacy policy evaluation. In: ACM Workshop on Security and Privacy Analytics, pp. 15–21 (2018)
43. Voigt, P., von dem Bussche, A.: The EU General Data Protection Regulation (GDPR). Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57959-7
44. Wang, H., Lai, T. T.-T., Roy Choudhury, R.: Mole: Motion leaks through smartwatch sensors. In: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, pp. 155–166 (2015)
45. Wang, S.I., Manning, C.D.: Baselines and bigrams: simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 90–94 (2012)
46. Wang, X., Qin, X., Hosseini, M.B., Slavin, R., Breaux, T.D., Niu, J.: Guileak: tracing privacy policy claims on user input data for android applications. In: Proceedings of the 40th International Conference on Software Engineering, pp. 37–47 (2018)
47. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45 (2020)
48. Yu, H., Hua, J., Julien, C.: Dataset: analysis of IFTTT recipes to study how humans use internet-of-things (IOT) devices. arXiv preprint arXiv:2110.00068 (2021)
49. Yu, L., Luo, X., Liu, X., Zhang, T.: Can we trust the privacy policies of android apps? In: 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 538–549. IEEE (2016)
50. Yu, L., Zhang, T., Luo, X., Xue, L., Chang, H.: Toward automatically generating privacy policy for android apps. IEEE Trans. Inf. Forens. Secur. **12**(4), 865–880 (2016)
51. Zimmeck, S., et al.: Maps: scaling privacy compliance analysis to a million apps. Proc. Priv. Enhancing Tech. **2019**, 66 (2019)
52. Zimmeck, S., et al.: Automated analysis of privacy requirements for mobile apps. In: AAAI Fall Symposium (2016)