# A design science approach to developing an integrated mobile app forensic framework

**Article** *in* Computers & Security · February 2021

DOI: 10.1016/j.cose.2021.102226

Xiaolu Zhang
University of Texas at San Antonio

Kim-Kwang Raymond Choo
University of Texas at San Antonio

# A design science approach to developing an integrated mobile app forensic framework

Xiaolu Zhang, Charles Zhechao Liu, Kim-Kwang Raymond Choo, Jesus A. Alvarado

*Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249, USA*

## Abstract

As our society becomes increasingly interconnected and reliant on technologies, the need for digital forensics is no longer restricted to cybercriminal investigations. For example, when a private organization's system is compromised due to unauthorized access or espionage,, digital forensic investigations can help to reveal the 'what, why, how, who, when, and where' relating to the incident, and the findings can inform decision-making and mitigation strategy formulation. Taking an interdisciplinary lens, this paper uses the design science approach to guide the development of an integrated digital forensic framework. It is then demonstrated how the framework can be used by a forensic investigator to examine Android applications, using a series of case studies (also referred to as "instantiations" of the method artifact in the design science literature).

*Keywords:* Cybercrime; Cybercrime investigation; Digital forensics; Design science; Mobile forensics; Digital investigation

## 1. Introduction

Digital investigation is playing an increasingly important role in both cybercrime and traditional crime investigations, as our society becomes more digitalized and concepts such as smart cities, smart buildings and Industry 4.0 become the norm. For example, in a drug or sex trafficking investigation, potential sources of evidence are likely to include mobile devices that have been used for communications (e.g. Android and iOS devices and instant messaging applications such as WhatsApp), IP-based CCTV cameras installed in the city, Internet of Things (IoT) devices such as wearable devices (e.g. smart watch), in-vehicle infotainment systems (with built-in GPS, WiFi, and Bluetooth capabilities to provide connectivity with external networks and devices), and cloud computing services (e.g. cloud storage services). As a typical use case, data from iPhones and Amazon Echo are automatically uploaded to iCloud and Amazon Drive if automatic synchronization is set.

The increasing variety and volume of computing devices and the remote storage of data (e.g. in cloud servers), however, complicate the process of gathering digital evidence relevant to the criminal activity (e.g. unauthorized access and espionage). There is, therefore, a need to have a systematic approach to support digital investigators in identifying, acquiring, analyzing, and presenting the broad range of potential artifacts of forensic interest – a process also known as digital forensics. Such forensic artifacts can then be used to support legal elements of proof and inform decision or strategy-formulation in different contexts. The legal elements of proof can vary between jurisdictions (e.g. United States *vs.* European or Asian countries) and nature of offence (e.g. unauthorized access to a system *vs.* online child sexual exploitation). The importance of having common standards to guide the investigation and prosecution of cybercrime offences is also iterated by the European-Commission (2010).

In order to make the findings of this study generalizable to a broad range of contexts, this study summarizes the various approaches and methodologies involved in carrying out a forensic investigation into an integrative framework. In addition, this study presents a holistic yet versatile protocol to account for the different situations that may arise in the data recovery and investigation process. Given that there is very little prior work or theory that can be relied on to develop this framework, this study adopts a design science (DS) approach for developing information systems (IS) as advocated by Gregor and Hevner (2013); Hevner et al. (2004); Peffers et al. (2007). Unlike traditional IS theories that focus on exploring the underlying factors, relationships, or organizational structures that drive the development, adoption, and implementation of information technologies, the DS theory proposes a set of principles and guidelines that aim at developing "artifacts" to solve a practical problem, which aligns well with the objective of the study. The main artifact in this study is an integrated forensic framework manifested in the form of a streamline protocol, which characterizes the "method" artifact as advocated in the DS literature (Hevner et al., 2004; March and Smith, 1995). To demonstrate the usability of the

*Email addresses:* `xiaolu.zhang@utsa.edu` (Xiaolu Zhang), `charles.liu@utsa.edu` (Charles Zhechao Liu ), `raymond.choo@fulbrightmail.org` (Kim-Kwang Raymond Choo), `alex@alvaradoonline.com` (Jesus A. Alvarado) method artifact, the protocol is applied to six different cases that encompass a wide range of mobile app and IoT device combinations. These six cases can also be viewed as six "instantiation" artifacts from a DS perspective, which allow one to evaluate the effectiveness of the processes proposed in the protocol in a very comprehensive manner.

The next section briefly reviews the extant literature on digital forensics, particularly in the context of mobile apps, followed by a discussion on the DS guidelines and principles in Section 3. Section 4 describes the proposed forensic framework for Android mobile applications (apps), which is then evaluated using the apps presented in Section 5. In Section 6, a comparative summary between the findings from Section 5 and the reports submitted to the Digital Forensics Research Conference (DFRWS) IoT forensic challenge 2018-2019 (Nambiar et al., 2019; Cha and Park, 2019; Park et al., 2019; Hines et al., 2019; Lee et al., 2019) reinforces the importance of having a dedicated approach, such as the proposed framework, in guiding forensic investigations. Subsequently, the last two sections present discussions of the theoretical and practical implications of this study, and conclusions and directions for future research.

## 2. Related literature

Mobile device and application (app) forensics is a relatively mature research area, although a number of challenges remain. This is partly due to the fast-pace nature of mobile device and app advances, and the broad range of mobile devices that rapidly emerged in the last decade. For example, in recent literature reviews on mobile device and app forensics, Barmpatsalou et al. (2018) and Manral et al. (2019) observed that data acquisition from mobile devices remains a key operational and research challenge. This challenge is also echoed by researchers such as Servida and Casey (2019) and Park, Kim, Park, Lee and Kim (2019), due to the use of data-at-rest and data-in-transit encryption solutions or when newer devices (e.g. IoT devices) are not supported by existing commercial tools. Park, Kim, Park, Lee and Kim (2019), for example, highlighted the importance of examining encrypted backup data of mobile device users. Using KoBackup and HiSuite (i.e. a Huawei mobile app, and the associated computer backup program) as case studies, they demonstrated how a forensic investigator can decrypt the encrypted Huawei smartphone's backup data to obtain access to the user's sensitive data.

A number of authors have also forensically examined different mobile apps, with the aims of understanding where data are stored and how such data can be acquired. For example, Choi et al. (2019) studied three popular Windows instant messaging apps, namely: KakaoTalk, NateOn and QQ messenger. They attempted to determine how the database files of these three apps are stored and encrypted, and whether these encrypted database files can be successfully recovered without relying on the mobile user's password. van Zandwijk and Boztas (2019) examined the popular iPhone health app installed on an iPhone 6, iPhone 7 and iPhone 8. They were able to reliably determine the mobile app user's walking distances, walking style, and walking speed, and consequently provided a digital trace of the user to be used as digital evidence.

To facilitate triage, Guido (2019) explained how a client-side app can be designed to only scan storage locations on the mobile device and report any identified changes in the data (in comparison to a prior scan) to the server. This avoids the need to acquire and examine the entire dataset. Such an approach is more useful in a cyber security / risk management context, rather than a post-event forensic investigation.

Nowadays, sources of potential digital evidence are not limited to more traditional mobile devices such as smartphones, IoT devices such as wearable devices (e.g. smart watch and FitBit) (Rondeau et al., 2019; Servida and Casey, 2019; Zhao et al., 2019) are increasingly adopted by consumers, which generates significant interests among researchers to examine whether the traditional forensic approaches can still be applied to these newer generation of computing devices. For example, Zhao et al. (2019) examined the smart watches worn by 25 participants over a six-month period. Based on their analysis of these 25 smart watches, they demonstrated how a forensic investigator can reliably determine whether the user was drinking (alcohol) and the level of intoxication. Li et al. (2019) examined the Amazon Echo and the Alexa companion app, and demonstrated how different forensic artifacts (e.g. location data, timestamp information, user search history and streamed recordings) can be collectively used to profile the home user.

While there have been a number of studies focusing on mobile device and app forensics, we observe that there is a lack of a holistic yet versatile protocol, which accounts for the different situations that may arise in the data recovery and investigation process. The existing findings are largely case dependent and cannot be easily generalized to other similar devices or contexts. This is the gap we seek to address in this paper. In the next section, we will briefly review the role of the DS theory in the IS field and and introduce the DS approach that will guide the development of such an approach.

It is observed that DS has been previously adopted in a small number of forensic studies. For example, Armstrong and Armstrong (2010), and Kurpierz and Smith (2020) explained how DS can facilitate forensic evidence processing, database forensics, and forensic accounting via forensic acculturation, respectively. In a follow-up work, Al-Dhaqm et al. (2020) presented a high-level database forensic investigation process based on DS. The process comprises the planning, preparation and pre-response phase, the acquisition and preservation phase, and the analysis and reconstruction phase. Since database is only one of several sources of evidence in a mobile forensic investigation, the proposed process of Al-Dhaqm et al. (2020) can complement the proposed Android mobile app forensic framework described in Figure 1.

## 3. Design science (DS)

DS is a research paradigm widely adopted in computer science (Anthes, 2010; Preston and Mehandjiev, 2004), engineering (Eekels and Roozenburg, 1991; Fulcher and Hills, 1996), operation research (van Aken et al., 2016) and other science disciplines (Simon, 1996) to create tangible and functional artifacts (i.e. software, hardware, or processes) to address a real-world issue. An artifact developed based on the DS principles typically follows a systematic design process and a set of well-accepted guidelines. Therefore, such an artifact can often be applied and replicated by other researchers in similar contexts to maximize the value of the artifact and facilitate knowledge dissemination.

March and Smith (1995) are among the first scholars to introduce DS into the IS field. The notion of "IT Artifact" is formally introduced and defined as a technology product that may take the form of a construct, model, method, and instantiation, all of which require both a "build" process and an "evaluate" process that work together to fulfill the design objective. Since then, DS-based research starts to emerge in various IS journals. This trend continues to grow and reaches an important milestone when Hevner et al. (2004) published their seminal work "Design Science in Information Systems Research" in MISQ in 2004. As noted by the editors note of Goes (2014), the work of Hevner et al. (2004) has a profound impact on IS community and leads to a paradigm shift in IS research methodologies. A set of guidelines have been put forth which include (1) design as an artifact, (2) problem relevance, (3) design evaluation, (4) research contributions, (5) research rigor, (6) design as a search process, and (7) communication of research (see Table 1 of Hevner et al. (2004)). These guidelines were subsequently extended to highlight a series of interdependent, and sometimes iterative activities as summarized by Peffers et al. (2007).

Building on the foundation of these pioneer work, a number of IS studies have embraced the DS methodology and established a young but thriving stream of research in the IS field. Gregor and Hevner (2013) conduct a systematic review of a total of 13 DS papers published by MISQ between 2006 and 2011 and conclude that "that DSR has yet to attain its full potential impact on the development and use of information systems due to gaps in the understanding and application of DSR concepts and methods" (Gregor and Hevner, 2013, p. 337) . Due to such gaps, a DSR knowledge contribution framework with two dimensions in both the problem and solution domains and a communication schema are proposed (Gregor and Hevner, 2013). More recently, to help researchers better understand the value of DS research and promote the publication of more DS studies in IS journals, Baskerville et al. (2018) provide a constructive framework in positioning DS research in technology and science evolution cycles. Such a view precisely highlights the characteristic that "synergistic interactions of technology and science" (Baskerville et al., 2018) occur throughout the evolution of the IS discipline. Hence, a reciprocal and iterative approach must be used in order to correctly apply the DS theory to position and study an IS phenomenon.

The guidelines and frameworks prescribed in these studies have provided an important basis for this study. In this research, we closely follow these recommendations and attempt to move beyond a superficial discovery of process or algorithm that limits itself to a situated implementation of a local solution (Gregor and Hevner, 2013). Instead, we focus on developing a streamline protocol that can be generalized to a wide range of forensic settings and easily adapted to be applied to a variety of mobile devices.

For the "build" activity, based on incremental experiments with individual apps and devices, we put together a protocol that aggregates a series of processes and technical approaches into an actionable protocol (in the form of a decision tree) that can be unambiguously adopted by a user who seeks to perform forensic investigation on an Android device. This protocol fits nicely with the definition of the "method" artifact proposed by March and Smith (1995) and Gregor and Hevner (2013). Based on their definition, a method is a set of steps (an algorithm or guideline) used to perform a task. Methods define processes, and provide guidance on how to solve problems (e.g. how to search the solution space). In developing such an artifact, this study adopts the "design as a search process" (Gregor and Hevner, 2013) approach and iteratively refines the processes and solutions involved to achieve a versatile framework and maximize generalizability.

For the "evaluate" activity, the protocol is applied to six different apps, with each characterizing a representative use case in the real-world. Based on the classification of artifacts by March and Smith (1995) and Gregor and Hevner (2013), the test cases in this study can be viewed as instantiations which are the concrete forms of the realization of an artifact in its environment. Instantiations operationalize constructs, models, and methods to demonstrate the feasibility and effectiveness of the models and methods they contain, which greatly complement the method artifact developed in this study.

The below summarizes how this study follows the seven guidelines advocated by Gregor and Hevner (2013).

- *Guideline 1 – Design as an artifact:* The requirements are to produce a viable artifact in the form of a construct, a model, a method or an instantiation. Hence, this paper develops a method artifact and six instantiation artifacts that meet the definitions established in the DS literature.

- *Guideline 2 – Problem relevance:* The objective is to develop technology-based solutions to important and relevant business problems. Hence, in this paper the method artifact is designed to address the increasingly import forensic and data retrieval tasks as outlined in the introduction, and consistently reported in mainstream media.

- *Guideline 3 – Design evaluation:* The utility, quality and efficacy of an artifact must be rigorously demonstrated via well-executed evaluation methods. Hence, the method artifact in this paper is designed to accommodate a broad spectrum of forensic settings. The six cases included in this study provide a comprehensive test plan to evaluate our method artifact. Each of the instantiation artifacts characterizes a popular yet distinct app/device combination. The collection of these cases thus represents a wide array of similar use cases in the real-world.

- *Guideline 4 – Research contribution:* Must provide clear and verifiable contributions in the areas of design artifact, design foundation, and/or design methodology & Our study contributes to both artifact development and design foundation. Hence, the method artifact in this paper is based on publicly accessible technical approaches, and all of the instantiation artifacts can be replicated by other researchers in identical settings. The method artifact (protocol) is also one of the first works to establish a set of guidelines to perform forensic and data recovery tasks in mobile devices, which sets a design foundation for future studies to extend to other application domains.

- *Guideline 5 – Research rigor:* DS research relies on the application of rigorous methods in both the construction and evaluation of the artifact. Hence, this paper strictly follows the principle that the methods and tools used in the digital forensic investigation should be forensically sound (McKemmish, 2008), in the sense that (1) the process does not affect the meaning and interpretation of the digital evidence, (2) any potential errors introduced during the process are identified and can be adequately explained, (3) the process is repeatable and the findings are reproducible under the same settings, and (4) the digital forensic investigator has the relevant (and up-to-date) expertise.

- *Guideline 6 – Design as a search process:* The search for an effective artifact requires utilizing available means to reach desirable ends. Hence, this paper adopts an iterative process in searching for the test cases, which also leads to incremental design improvement of the method artifact (protocol). The final product summarized in the following sections are the cumulative results of iterative refinement of individual cases (including those not reported in the study).

- *Guideline 7 – Communication of research:* DS research must be presented effectively both to technology-oriented and management-oriented audiences, as well as an intelligent lay audience. Hence, in this paper the choice of a decision tree (described in the next section) and the detailed step by step instructions of the individual cases are developed to ensure that they can be comprehended and executed by a general audience that does not have a strong technical background.

## 4. Proposed Android mobile app forensic framework

In this section, we will describe the artifact from the DS approach, namely: the Android[1] mobile app forensic framework – see Fig. 1.

Following the DS guidelines, we adopt a decision tree structure to ensure that the development of our framework follows a "design as a search process" principle and all possible scenarios are accounted for. As a result of this design principle, users of the framework do not need to impose their own assumptions when executing the protocol in a forensic task.

In a typical forensic investigation, the investigators will avoid directly analyzing the original case exhibit as far as practical, to avoid contamination of the digital evidence. Instead, the forensic investigator will perform either a logical acquisition or a physical acquisition (i.e. software-based data acquisition). Physical acquisition is usually achieved using some commercial forensic tools (e.g. EnCase and FTK), where a bit-by-bit copy of the device's data is taken. This allows the investigator to recover deleted data that has not been overwritten. For devices that do not support physical acquisition (e.g. the device make and model are not supported by existing forensic tools), an investigator can perform a logical acquisition, which contains only files that have not yet been deleted. To acquire a logical image, an investigator generally must be able to unlock the device and switch on the USB Debug mode, as well as gaining root access to the device (also referred to as rooting for Android devices, or jailbreaking for iOS devices). If either of the (software-based) logical or physical acquisition approaches fail, then hardware acquisition using chip-off (Heckmann et al., 2019) or a universal hardware interface such as JTAG, UART and SPI would be the last approach to acquire data from the device.

Next, the forensic investigator will start examining the acquired (physical / logical) forensic image. First, (s)he should search for the folder name of the app. We remark that an app's folder name must be identical to its package name. If the folder exists, the investigator would manually examine the files in the app's folder. Typically (if the app developer follows the common standard), an Android app folder should include sub-folders such as *shared_prefs*, *cache*, *files*, *databases*, and *lib*. In other words, different

[1] Android has the biggest market share for mobile devices (Statista, 2019).

types of forensic artifacts can be recovered from these sub-folders. However, based on our experience, folders such as *files* and *cache* do not necessarily store a single type of evidence. The forensic investigator can also rely on automated forensic knowledge sharing platforms, such as ForKas (Zhang et al., 2019) and Hansken (van Beek et al., 2015). ForKas is an automated forensic knowledge sharing platform, which stores a range of schemas contributed by the digital forensic community, and the schemas can facilitate automated extraction of forensic artifacts from mobile devices. Hansken, developed by the Netherlands Forensic Institute, is a centralized system designed to facilitate large scale forensic data analysis.

Therefore, rather than categorizing the expected evidence based on the sub-folders' name, we determined that the file formats in these folders are a better indication of the potential evidence. For example, XML files are usually found storing login credentials, encryption/decryption keys and users' profile information. The *cache* folder usually stores the web cache data generated internally in the appellation. Web browser apps are also widely used for Internet browsing, and hence the web cache is likely to contain user event, browsing and search history. For example, a user's historical events can potentially be recovered from a Json file extracted from the web cache. SQLite files are usually (but not necessarily) found in the *databases* folder. The data included in a SQLite file could contain information such as user activity / login history, previous chat messages, audio / video clips recorded/cached on the local device, and so on. In some cases, a forensic investigator may find standalone media files (pictures, videos or documents) stored in the 'files' folder.

It is worth noting that the integration of the above steps and processes are not derived in a straight forward manner. Various trials and errors occur during the design process which results in multiple re-iteration of the discovery process.

## 5. Case studies

Consistent with the DS principles and recommendations, we use a series of case studies to demonstrate that the protocol designed in this study can account for the forensic tasks in a variety of devices and apps. These cases serve as "instantiations" of the method artifact and provide a basis for us to evaluate the usability and effectiveness of the proposed protocol.

Table 1 summarizes the apps examined in this study, along with their connected devices and package names. These apps and devices represent some of the most popular consumer choices in the market, and the apps are installed on a Xiaomi Redmi Android 8.1 smart phone.

The framework described in Figure 1 is then used to guide the investigation of six popular apps. In the case studies, a 'software-based data acquisition' is performed which results in the physical image of the mobile device. The findings of the analysis are briefly presented in Sections 5.1 to 5.6.

### *5.1. iSmartAlarm*

*iSmartAlarm* can be used to access multiple connected IoT devices / sensors from the user's smart phone, via the IoT controller *CubeOne*. Following the protocol, we locate the app data of iSmartAlam in folder `/data/data/iSA.common/`, and its outsourced data in folder `/media/0/iSmartAlarm`. Also, in the app's `shared_prefs` folder, we locate the file `iSmartAlermData.xml` and obtain the user's password, phone number, registered country and country code in clear-text (i.e. not encrypted). The located phone number and password can facilitate a forensic investigator to gain access to the user's account and data stored in the cloud server.

Subsequently, we locate the database file `../iSA.common/databases/iSmartAlarm.DB`, which records the user's action and system activities are recorded. In this file, table `TB_IPUDairy` reveals the actions that change the status of the system (in the 'profileName' column), the time of the action accrued (in the 'data' column), the user who performs the action (in the 'operator' column), and so on.

Next, we locate Table `TB_SensorDairy`, which allows a forensic investigator to discover the IoT devices connected to the system. We remark that iSmartAlarm only supports the Z-Wave protocol (in the communication with the connected sensors) and the device address in a Z-wave network is 32-bit/4-byte long, therefore the located 4-byte MAC addresses in column 'sensorID' of Table `TB_SensorDairy` is the sensor's MAC address (i.e. a network interface controller's unique identifier). The data in the column "action" can be used to determine the action of the sensors. For example, we determine the definition of the numbers and actions by decoding the Android Application Package (APK) file extracted from `/data/app/iSA.common`. In file `/res/values/strings.xml`, tags named with "`sensor_act_N_ACTION`" show that number $N$ is associated with a specific $ACTION$ of a sensor – see also Table 2.

In the outsource data folder, the most significant data is in the log files under (`../iSmartAlarm/Log`). From the file name, we can determine the name of the file when it was first created. In the file, a forensic investigator can also find errors generated by the app. In addition, at the beginning of the the log file, we locate the TimeZone information for the app, and using this information we are able to determine the exact timeline of the various activities. We also determine from the log file the WiFi networks that the app had connect to or attempted to connect to (from "wifiInfo", which includes the network name (SSID), and the router's MAC address (BSSID)).

Table 1: IoT apps studied in this paper

| Android App | Version | Package name | Connected device |
|---|---|---|---|
| iSmartAlarm | 5.1.1 | iSA.common | iSmartAlarm |
| Nest | 5.21.1.2 | com.nest.android | Nest Camera, Nest Hello, Nest Thermostat |
| Arlo | 2.7.11 | com.netgear.android | Arlo Pro Camera, Arlo Base Station |
| Blink | 5.0.11.6 | com.immediasemi.android.blink | Blink Camera XT2 |
| Yi | 4.10.0 | ants360.yicamera.international | Yi 720p Camera, Yi Dome Camera 1080p |
| Wink - Smart Home | 6.9.0.33 | com.quirky.android.wink.wink | Wink hub |

### 5.2. *Nest*

*Nest* app allows the user to control all Nest IoT devices in the same network (e.g. a smart home or smart building) via the user's mobile device.

In our analysis, we locate the XML file `../com.nest.android/shared_prefs/com.nest.android.preferences.xml` and hence, obtain the username in the tag "userLogin" and the user's authentication token in the tag "user_token". This allows a forensic investigator to login to user's account using another device. This in itself is a security weakness, and introducing an expiration for the user's authentication token is unlikely to be an effective mitigation strategy. For example, one can simply move the located XML file to another device, in order to login to the user's account.

Prior activities associated with the IoT devices registered to the Nest app can also be retrieved from the app's cache files we located in the analysis. The cache files are in clear-text *Json* files in the folder `../com.nest.android/cache/cache`. These files were created when the user logged in. Given that Json file consists of one or an array of Json objects, each of these objects has an "object_key" element. The latter indicates the categories of the object. For example, the "value" element (which is also a sub-object) of the object stores the data of the object. The objects of potential forensic interest are explained below – see also Table 3.

**geofence_info.ID** object stores the geographical location information of the registered IoT devices, where can be used to pinpoint the device user (e.g. Fitbit). For example, as observed from Table 3 shows, the sub-object "fences" stores the latitude and longitude of the physical address of the IoT device.

**structure.ID** object (e.g., structure.a1abcc10-b9fb-11e7-bab1-0e967d55f198) includes in-depth information (found in dedicated objects) about the IoT environment. For instance, element "swarm" in "value" always includes the unique identifier of all registered IoT devices, which can be used to identify the IoT devices and the associated activities.

**DEVICE_NAME.ID** object (e.g., quartz.fb83ebdfcf6c4ff49e7d82cf0da371c6) is associated with the device's identifier, which also specifies the features of an IoT device. Therefore, the number of the "DEVICE_NAME.ID" objects should be identical to the identifiers found in the "structure.ID" object. The object for different IoT devices may be structured differently. The features of the object can be used to identify the make and model of the device, IP address, etc. For example, feature "streaming.cameraprofile.VIDEO_H264_100KBIT_L30" and "streaming.data-usage-tier.120" are unique for a camera, and "upper_safety_temp" and "temperature_lock_low_temp" are associated with a thermostat. Therefore, based on the features we located in the object "quartz.fb83ebdfcf6c4ff49e7d82cf0da371c6", we determine that device "quartz.fb83ebdfcf6c4ff49e7d82cf0da371c6" is a "Topaz-2.7" Nest Camera.

**message_center.ID** object (e.g., message_center.11977577) stores the emergency notifications that the app received from the IoT devices. The "value" of the object is consisted with an array of sub-objects where the "value" (e.g. value "protect_smoke_warn" indicates that a smoke detector had detected smoke) and the "timestamp" revealed the activities of the IoT devices and the timestamp.

The above is also a snapshot of some of the data we located during the analysis. For example, we also located cache files in the folder `../com.nest.android/cache/dcnetwork`. The cache files were generated during the caching of snapshot picture captured when a Nest camera was triggered by a sound or movement. As the pictures are stored in clear-text, a forensic investigator can simply locate the unique number of the picture (e.g. 'JFIF' for a JPEG file) for subsequent data craving.

### 5.3. *Arlo*

*Arlo* camera is another popular device, where over 12.4 million devices have reportedly been sold as of May 8th, 2019[2].

During initial investigation of the Arlo app, we locate the access token in the `../share_prefs/Phoenix.xml` file and are able to gain access to user's app account. The located Phoenix.xml file also includes information about whether the user has touch ID setup, and the registered email address. In addition, user private information such as first name, last name, address (street number, street name, postal code), credit card information (including the expiration month and year, and CVV number) and timestamp information can be recovered from the `../app_webview/Web Data` file.

[2] https://investor.arlo.com/ir-home/default.aspx

We also locate the snapshots taken by the camera in the directory `../cache/http, cams, and thumbs`, which appears to be the default location for most cameras when storing pictures or videos.

### 5.4. *Blink*

Our analysis of *Blink* reveals that images stored in the `image_manager_disk_cache` folder are unencrypted and contain pictures captured by the app. We also locate the unencrypted videos stored in the `videos` folder and the unencrypted video clips in the `share` folder (saved as mp4 files), and these videos can be viewed using any video player.

In the `../database` folder, we determine that the blink.db stores all motion activities in the media table. The network table within the `blink.db` allows us to determine the device name, as provided by the user, time zone, and last update of the camera. From the syncmodule table, we are able to determine when the camera was last online, Wi-Fi strength and the serial number.

In the `../files/.Fabric/com.crashlytics.sdk.android.crashlytics-core` folder, we locate several Json files that contain information about the app and the mobile device it was installed on. The SessionApp.json file, for example, contains information about the app version. The file SessionDevice.json contains information about the manufacturer, disk space, total RAM, build model, whether the software is running on an emulator or not, and the available processors on the mobile device. The SessionOS.json file contains information about the the Android version running on the mobile device.

In `../shared_prefs`, we locate the access token in the file `com.google.android.gms.appid.xml`. This finding echoes the analysis of other apps, in the sense that the user's authentication / access token is generally located in the `shared_prefs` folder. We are also able to locate information about the user, such as email address, and phone make and model.

### 5.5. *YI*

Our analysis of the *YI* camera app reveals that all pictures are stored in the sub-folder `image_manager_disk_cache` of the `../cache` folder. We also determine that the `../databases` folder contains multiple useful database files. For example, from `alert.db` we can determine when each motion alert was detected by the camera. The `ants.db` stores device name (can be changed by the user), viewer account and password, model number, SSID, local IP address, MAC address, time zone, and user name. In `ants_gallery.db`, we determine that video files are stored in `/sdcard/DCIM/YiCamera/`. Within the `ants_message.db` file, we can access the userId column, which contains the ID number corresponding to the user account that accessed the application. This database also describes the device type under the devType column, and devOSVersion column will allow a forensic investigator to determine the operating system of the login device. The loginIP column contains the public IP address of the equipment used to log in.

Again, we locate the user's authentication / access token in the `../shared_prefs` folder, and hence a forensic investigator can gain access to the user's account and view cameras that are associated with this particular account. The forensic investigator may also be able to playback the video clips (e.g. when the camera was triggered due to motion detection) stored locally or in the cloud.

```
{
 "action":{
  "reading":{
   "loudness":true
  },
  ...
 },
 "context":{
  "media_animated_url":"https://www.dropcam.com/
      api/wwn.get_animated_image/[...]","
      media_url":"https://www.dropcam.com/api/
      wwn.get_image/[...]"
 },
 "created_at":[...],
  "object":{
   "object_id":"[...]",
   "object_name":"[...] Camera",
   "object_type":"camera"
  },
 ...
}
```

Listing 1: to avoid potential privacy implications.]Samples of access URL(API) for Nest Camera. Note that the sensitive data is replaced with [...] to avoid potential privacy implications.

### 5.6. *Wink - Smart Home*

The *Wink - Smart Home* app is developed for *Wink Hub*, an IoT controller / base station that allows the user to interact with over a hundred IoT devices on the market. From the analysis, we locate the user token in the “shared preferences" file, which can be utilized to facilitate account access. The file also includes the user's email address. We are also able to determine the geographical location of the Wink Hub device and recover the historical activities of the connected IoT devices from the SQLite database, and obtain the camera snapshots taken by the IoT devices connected to the Wink Hub from both the “cache" folder and the database.

Forensic artifacts of interest from the app analysis include the following:

**geofence** entry stores the geographical location of the Wink Hub, such as the latitude and the longitude, and the location.

**linked_service** entries save the services registered on the Wink Hub, such as “google_now" (google voice assistant service), “amazon_alexa", and “nest" (service for connecting to Nest IoT products). These services allow Wink Hub to interact with a third-party IoT product,

Table 2: Sensor actions and action codes

| Code | Action | Relative sensor |
|---|---|---|
| 1 | Available | All sensors |
| 2 | Unavailable | All sensors |
| 3 | Open | Door sensor |
| 4 | Closed | All sensor |
| 5 | Block | Contact sensor |
| 6 | Low Battery | All sensors |
| 7 | N/A | N/A |
| 8 | Glass Breakage Detected | Glass Break Detecter |
| 9 | Not trigger | Unknown |
| 10 | Online | All sensors |
| 11 | Offline | All sensors |
| 12 | LED Open | Unknown |
| 13 | LED Close | Unknown |

and can also help the forensic investigator determine the IoT devices involved in the case.

**DEVICE_TYPE** entry's naming convention follows that of the device type (e.g. "camera" indicates that the device is a web camera). A forensic investigator can also find other IoT device-specific information such as MAC address, device model, connection state, and the last interaction time.

**activity** entries store the activities of the IoT devices. Listing 1 shows a sample content of an activity retrieved from the sample dataset. The listing only includes some relevant data. For example, element "created_at" reveals the timestamp of the activity, element "object" specifies the device who performed this activity, element "reading" of the "action" object indicates what the activity is (in our example, "loudness" means a noise was detected), and element "context" includes the data along with the activity. As the device is a "camera", element "context" contains the URL of the images taken by the camera when "loudness" was detected. In terms of the Nest API [3], the `wwn.get_image` URL allows a forensic investigator to retrieve an image file captured for a sound or motion event and the `wwn.get_animated_image` URL returns a Graphic Interchange Format (GIF) file captured for a sound or motion event.

Cached pictures can be found in the folder `../com.quirky.android.wink.wink/cache/image_manager_disk_cache`.

## 6. A comparative summary

We posit that the DS approach can guide investigations, particularly those involving newer devices or inexperienced investigators (by providing them a systematic approach). For example, the four applications (iSmartAlarm, Nest, Arlo and Wink-Smart Home) we analyzed in Sec. 5 are also included in the DFRWS IoT forensic challenge 2018-2019[4]. A total of five reports were submitted by different research teams[5], and we include four of these reports in our comparison (see Table 4) since one of these reports (Hines et al., 2019) did not include details of their app analysis.

Table 3: Selected artifact types located in Nest app's cache file

| Key | Value | Content |
|---|---|---|
| geofence_info.ID | fences | latitude, longitude, etc |
| structure.ID | swarm | List of the registered IoT device(s) |
| DEVICE_NAME.ID | model | Model of the IoT device |
| | mac_address | MAC address of the IoT device |
| message_center.ID | timestamp | When the message was received |
| | value | Notification received from IoT device(s) |

In Table 4, the location and the type of the key artifacts are listed in the first column, where the ✕ symbol denotes the successful recovery of the relevant artifacts. For example, the same artifact was recovered from an XML file in the iSmartAlarm's `shared_prefs` folder in our approach as well as those of Nambiar et al. (2019); Park et al. (2019); Lee et al. (2019). We observe that using our approach, we were able to recover 12 out of 14 key artifacts, which ranks on the top of the five methods being compared (outperforms three other research teams and ties with one team).

A post-analysis review of the challenge reports suggested that our method did not include the evidence at `../databases/cache`, which was included in other reports for the Nest application. The reason of the absence of this evidence in our report is that the SQLite file '`../databases/cache`' already contains the index of the cache files under the `../cache/cache/` folder. If the cache files that includes the forensic artifacts had been found in the `../cache/cache/` folder, then it is no longer necessary to access the `cache` SQLite file. The only artifact that was missing from our proposed case study is how to parse the 'frame_database' SQLite file's content under the `../cache` folder. The data must be parsed with a dedicated method that could be found from page 73 in the 'TapiocaPearlo' team's report (Lee et al., 2019).

Despite the absence of this dedicated method, our approach allows us to identify artifacts missed by the other teams. For instance, the out-sourced folder for iSmartAlarm could potentially contain the log data of the application, and the SQLite file `../app_webview/Web Data` in the Arlo application could contain user's private information such as home address and credit card information. In addition, although the XML files under the `shared_prefs` folder were investigated in some of these reports (which allowed for recovery of some personal/account information

[3] https://developers.nest.com/reference/api-camera

[4] https://github.com/dfrws/dfrws2018-challenge, last accessed Jan 1, 2021

[5] https://github.com/dfrws/dfrws2018-challenge/tree/master/challenge-submissions, last accessed Jan 1, 2021

Table 4: Case-based comparison

| | **DS approach** | **AIForensics** (Cha and Park, 2019) | **DF&C lab** (Park et al., 2019) | **AMRITA** (Nambiar et al., 2019) | **TapiocaPearlo** (Lee et al., 2019) |
|---|---|---|---|---|---|
| **iSmartAlarm** | | | | | |
| ../shared_prefs/ –> XML | × | | × | × | × |
| ../databases/ –> SQLite | × | × | × | × | × |
| other (out-sourced) folder –> Clear-text | × | | | | |
| **Nest** | | | | | |
| ../shared_prefs/ –> XML | × | | ×* | | ×* |
| ../cache/cache/ –> Web Cache (Json) | × | × | × | | × |
| ../cache/dcnetwork/ –> Web Cache (Media) | × | | × | | × |
| ../databases/cache –> SQLite | | × | × | | × |
| ../cache/ –> SQLite | | | | | × |
| **Arlo** | | | | | |
| ../shared_prefs/ –> XML | × | | ×* | | ×* |
| ../app_webview/Web Data –> SQLite | × | | | | |
| ../cache/{http, cams, or thumbs}/ –> Web Cache (Json, Media) | × | × | × | | × |
| **Wink** | | | | | |
| ../shared_prefs –> XML | × | | ×* | | ×* |
| ../databases/ –> SQLite | × | × | × | | × |
| ../cache –> Web Cache (Media) | × | × | × | | × |
| Artifacts recovered | 12/14 | 6/14 | 11/14 | 2/14 | 12/14 |

Symbol × indicates that the same evidence recovered based on the DS approach was also included in the corresponding DFRWS report.
∗ the report had claimed most of the key evidence in the XML file except the user's access token.

of the applications), the access token (listed in the framework, Fig. 1) in the XML files were missed. Such access tokens could be potentially utilized to log into a suspect's account to recover further evidence. To ensure the usability and the reliability of the access tokens in the DFRWS challenge, we analyzed the dataset of the challenge, recovered the access token from the applications based on our approach, and then confirmed that the access token of the 'Wink' application could be utilized for logging the suspect's Wink account on another Android device.

Another important observation from the published reports of (Nambiar et al., 2019; Cha and Park, 2019; Park et al., 2019; Hines et al., 2019; Lee et al., 2019) is that there does not appear to be a systematic approach taken to perform the investigations which may explain the differences in the types and number of artifacts recovered (see Table 4). Several reports included in our comparison are driven by device-specific knowledge obtained from prior individual investigations which may not provide sufficient generalizability for other types of devices. In contrast, the iterative approach that we followed based on the DS principles allows to constructive a comprehensive decision tree as demonstrated in Fig. 1.

Such a structural protocol not only strengthens the performance of our method but also provides a common basis for future studies to replicate and extend our work. This reinforces the importance of having a standardized approach, such as our proposed DS approach, to guide future forensic investigations.

## 7. Discussion

Following the recommendations from the DS literature, once an artifact is developed, the researcher should focus on demonstrating the usability and applicability of the artifact in an environment that the artifact is intended to work in (March and Smith, 1995). The six cases provide a comprehensive testbed for our proposed framework (method artifact) and its associated processes. The choice of newer generation of smart devices in the market is consistent with the notion that the artifact must be properly positioned in the technology evolution cycles to justify its relevance and maximize its impact. The number of users adopting these devices substantiates the relevance of our finding and its potential positive impact on minimizing the risk of data breach and privacy intrusion among these users.

Table 5 summarizes the main results of these test cases. It can be shown that our streamline protocol greatly facilitates the forensic process by shortening unnecessary exploratory examination of alternative pathways in a forensic investigation. A variety of sensitive data such as username and password, pictures, email address, geographical location, SSID, device name and id, and activity history, can be retrieved by following the steps outlined in our protocol. From a DS perspective, each of these outputs can be viewed as a standalone construct which may become the subjects of interest in other forensic studies. Therefore, even though the proposed integrated framework does not include individual construct artifacts, it is capable of serving as a new source for deriving new privacy constructs in the era of big data and pervasive privacy intrusion.

In terms of evaluating the contributions of the artifacts

Table 5: Case study summary of results.

| **Application** | **Login credentials** | **Access token** | **User private data** | **Historical events** | **Media files** |
|---|---|---|---|---|---|
| iSmartAlarm | $S_X$ | | | $D_Q$, $O_A$ | |
| Nest | | $S_X$ | $C_J$ | $C_J$ | $C_M$ |
| Arlo | | $S_X$ | $S_X$,$O_Q$ | $F_A$ | $C_M$ |
| Blink | | $S_X$ | $D_Q$, $F_J$ | $D_Q$ | $C_M$ |
| Yi | | $S_X$ | $D_Q$ | $C_M$ | |
| Wink-Smart Home | | $S_X$ | $D_Q$ | $D_Q$ | $C_M$ |

Folder of artifact$_{File\ format}$

**Folder of artifact:**
S: shared_prefs
C: cache
F: files
D: databases
O: other/out sourced folder

**File format:**
X: XML
J: JSON
H: HTML
M: known media formats (e.g. JPEG, GIF)
Q: SQLite
A: plain-text/other formats

developed in this study, as pointed out in other seminal DS research in the IS literature (Gregor and Hevner, 2013; Hevner et al., 2004; Nunamaker Jr and Briggs, 2011; Jr. et al., 2013), the major contributions of a DS paper can be assessed in terms of *proof-of-concept* and *proof-of-value*. Proof-of-concept is established when there is enough evidence to show that the described conceptual design is feasible and promising, at least in a limited context, whereas proof-of-value is demonstrated when an IT artifact actually works in reality (Nunamaker Jr and Briggs, 2011; Jr. et al., 2013). Our results show that both the *proof-of-concept* and *proof-of-value* have been achieved in our study, thus meeting the criteria set forth in the extant DS literature.

From a theory building standpoint, the framework proposed in this study can complement any behavior theory or framework employed to study user behavior that causes or mitigates privacy intrusion or security vulnerabilities. Given that actual data breaches and security intrusions are difficult to predict and detect, it is often impossible (or costly) to observe and capture user behaviors associated with these incidents *ex ante*, leading to a large gap and data limitation in studying the behavioral antecedents of a privacy intrusion. The forensic framework developed in this study can help researchers target specific data breach objects, and focus on tracking specific user behaviors associated with the generation, use, and transmission of these objects, which will significantly improve the feasibility of data collection and enable some behavior studies that could not have been carried out due to the difficulty of identifying actual victims and tracking their behavior histories.

## 8. Concluding remarks

The importance of digital forensics will be more pronounced in the foreseeable future, as malicious cyber activities are a rapidly expanding form of criminality that knows no borders, affecting both public and private sector organizations, as well as individual users. Clearly, addressing the threats of complex cyber security problems would demand a holistic and systematic approach to help the practitioner community defend against the increasingly complex cyberthreat, and to investigate and respond to cyber incidents (Newhouse et al., 2017). According to the Routine Activity Theory (Cohen and Felson, 1979; Yar, 2005), one could possibly reduce crime by targeting one of the three constructs, namely: opportunity, guardianship and motivation. Digital forensics could enhance guardianship, by providing forensic investigators the capability to investigate and respond to criminal activities, and consequently decrease the incentive to commit a crime (Cohen-Almagor, 2013; Tilley and French, 2017). However, digital forensic investigation is generally not a trivial task, as acknowledged by both criminologists and forensic researchers (Servida and Casey, 2019; Van der Wagen and Pieters, 2015).

Thus, in this paper we present an integrated forensic framework for Android devices and apps, developed using the DS guideline. The framework is designed to expedite the digital forensic process and reduce the learning curve required of new and inexperienced digital forensic investigators. For example, using the framework to guide the investigation of six popular Android IoT apps, we demonstrated where and how the different forensic artifacts can be recovered. These forensic artifacts can facilitate the reconstruction of the event, for example ’what’ happened, ’how’ and ’when’ and ‘where’ did the event occur, ‘who’ was involved, and ‘why’ did the event happen (e.g. the motivation construct in the Routine Activity Theory (Cohen and Felson, 1979; Yar, 2005)).

As with other DS research, our work is not without limitations. In designing the integrated mobile forensic framework, we focus on Android mobile devices and do not include any iOS devices and apps in our artifact development. This is largely due to the vast differences between Android and iOS apps and their respective underlying mobile OS structures. However, we believe that our work lays a good foundation for other researchers to build on our Android forensic framework and extend it to iOS devices and apps, as well as other popular or emerging mobile and IoT ecosystems. In addition, we intend to introduce the use of the proposed framework to students enrolled in the undergraduate- and graduate-level digital forensic courses taught by the authors. This will allow us to pilot test the effectiveness of the framework in making forensic investigation more systematic or complete, since most of the students do not have prior digital forensic experience.

## References

Al-Dhaqm, A., Abd Razak, S., Dampier, D. A., Choo, K.-K. R., Siddique, K., Ikuesan, R. A., Alqarni, A. and Kebande, V. R. (2020), 'Categorization and organization of database forensic investigation processes', *IEEE Access* **8**, 112846–112858.

Anthes, G. (2010), 'Mechanism design meets computer science', *Communications of the ACM* **53**(8), 11–13.

Armstrong, C. and Armstrong, H. (2010), Modeling forensic evidence systems using design science, *in* 'IFIP Working Conference on Human Benefit through the Diffusion of Information Systems Design Science Research', Springer, pp. 282–300.

Barmpatsalou, K., Cruz, T., Monteiro, E. and Simoes, P. (2018), 'Current and future trends in mobile device forensics: A survey', *ACM Computing Surveys* **51**(3), 46.

Baskerville, R., Baiyere, A., Gregor, S., Hevner, A. and Rossi, M. (2018), 'Design science research contributions: finding a balance between artifact and theory', *Journal of the Association for Information Systems* **19**(5), 3.

Cha, I. and Park, A. (2019), Team aiforensics, Technical report.

Choi, J., Yu, J., Hyun, S. and Kim, H. (2019), 'Digital forensic analysis of encrypted database files in instant messaging applications on windows operating systems: Case study with kakaotalk, nateon and QQ messenger', *Digital Investigation* **28**(Supplement), S50–S59.

Cohen-Almagor, R. (2013), 'Online child sex offenders: Challenges and counter-measures', *The Howard Journal of Criminal Justice* **52**(2), 190–215.

Cohen, L. E. and Felson, M. (1979), 'Social change and crime rate trends: A routine activity approach', *American sociological review* pp. 588–608.

Eekels, J. and Roozenburg, N. F. (1991), 'A methodological comparison of the structures of scientific research and engineering design: their similarities and differences', *Design studies* **12**(4), 197–203.

European-Commission (2010), 'Communication from the commission to the european parliament and the council'.
**URL:** *https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2010:0673:FIN:EN:PDF*

Fulcher, A. and Hills, P. (1996), 'Towards a strategic framework for design research', *Journal of Engineering Design* **7**(2), 183–193.

Goes, P. B. (2014), 'Design science research in top information systems journals', *MIS Quarterly: Management Information Systems* **38**(1), iii–viii.

Gregor, S. and Hevner, A. R. (2013), 'Positioning and presenting design science research for maximum impact', *MIS quarterly* pp. 337–355.

Guido, M. D. (2019), 'Periodic mobile forensics'. US Patent App. 10/194,321.

Heckmann, T., McEvoy, J. P., Markantonakis, K., Akram, R. N. and Naccache, D. (2019), 'Removing epoxy underfill between neighbouring components using acid for component chip-off', *Digital Investigation* **29**, 198–209.

Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004), 'Design science information systems research mis quarterly', *MISQ Discovery* **28**(1).

Hines, K., Lewis, J., Phillpott, N., Sharo, R. and Zwach, S. (2019), Team niwc lant, Technical report.

Jr., J. F. N., Twyman, N. W. and Giboney, J. S. (2013), Breaking out of the design science box: High-value impact through multidisciplinary design science programs of research, *in* '19th Americas Conference on Information Systems, AMCIS 2013, Chicago, Illinois, USA, August 15-17, 2013', Association for Information Systems.
**URL:** *http://aisel.aisnet.org/amcis2013/ResearchMethods/GeneralPresentations/1*

Kurpierz, J. and Smith, K. A. (2020), 'Forensic acculturation for accountability in local governments: A design science approach for school leaders and citizens', *Journal of Forensic Accounting Research* pp. 0000–0000.

Lee, G., Choi, H., Kim, N., Jin, P., Park, S., Kim, S., Lee, S., Kim, S. and Jin, S. (2019), Team tapiocapearlo, Technical report.

Li, S., Choo, K. R., Sun, Q., Buchanan, W. J. and Cao, J. (2019), 'Iot forensics: Amazon echo as a use case', *IEEE Internet of Things Journal* **6**(4), 6487–6497.
**URL:** *https://doi.org/10.1109/JIOT.2019.2906946*

Manral, B., Somani, G., Choo, K.-K. R., Conti, M. and Gaur, M. S. (2019), 'A systematic survey on cloud forensics challenges, solutions, and future directions', *ACM Computing Surveys* .

March, S. T. and Smith, G. F. (1995), 'Design and natural science research on information technology', *Decision support systems* **15**(4), 251–266.

McKemmish, R. (2008), When is digital evidence forensically sound?, *in* 'IFIP international conference on digital forensics', Springer, pp. 3–15.

Nambiar, E. T. K., Kumar, P. A., R, H. and B, J. (2019), Team amrita, Technical report.

Newhouse, W., Keith, S., Scribner, B. and Witte, G. (2017), 'National initiative for cybersecurity education (nice) cybersecurity workforce framework (nist special publication 800-181)', *National Institute of Standards and Technology (NIST)* pp. 800–181.

Nunamaker Jr, J. F. and Briggs, R. O. (2011), 'Toward a broader vision for information systems', *ACM Transactions on Management Information Systems (TMIS)* **2**(4), 20.

Park, M., Kim, G., Park, Y., Lee, I. and Kim, J. (2019), 'Decrypting password-based encrypted backup data for huawei smartphones', *Digital Investigation* **28**(Supplement), 119–125.

Park, M. et al. (2019), Team df&c, Technical report.

Peffers, K., Tuunanen, T., Rothenberger, M. A. and Chatterjee, S. (2007), 'A design science research methodology for information systems research', *Journal of management information systems* **24**(3), 45–77.

Preston, M. and Mehandjiev, N. (2004), A framework for classifying intelligent design theories, *in* 'Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research', ACM, pp. 49–54.

Rondeau, C. M., Temple, M. A. and Lopez, J. (2019), 'Industrial iot cross-layer forensic investigation', *Wiley Interdisciplinary Reviews: Forensic Science* **1**(1), e1322.

Servida, F. and Casey, E. (2019), 'Iot forensic challenges and opportunities for digital traces', *Digital Investigation* **28**(Supplement), S22–S29.

Simon, H. (1996), 'The sciences of artificial, cambridge ma and london'.

Statista (2019), 'Subscriber share held by smartphone operating systems in the united states from 2012 to 2019'.
**URL:** *Subscriber share held by smartphone operating systems in the United States from 2012 to 2019*

Tilley, N. and French, J. (2017), Prevention and forensic science: How forensic evidence supports prevention, *in* 'The Routledge International Handbook of Forensic Intelligence and Criminology', Routledge, pp. 149–159.

van Aken, J., Chandrasekaran, A. and Halman, J. (2016), 'Conducting and publishing design science research: Inaugural essay of the design science department of the journal of operations management', *Journal of Operations Management* **47**, 1–8.

van Beek, H. M. A., van Eijk, E. J., van Baar, R. B., Ugen, M., Bodde, J. N. C. and Siemelink, A. J. (2015), 'Digital forensics as a service: Game on', *Digital Investigation* **15**, 20–38.

Van der Wagen, W. and Pieters, W. (2015), 'From cybercrime to cyborg crime: Botnets as hybrid criminal actor-networks', *British journal of criminology* **55**(3), 578–595.

van Zandwijk, J. P. and Boztas, A. (2019), 'The iphone health app from a forensic perspective: can steps and distances registered during walking and running be used as digital evidence?', *Digital Investigation* **28**(Supplement), S126–S133.

Yar, M. (2005), 'The novelty of âĂŸcybercrimeâĂŹ an assessment in light of routine activity theory', *European Journal of Criminology* **2**(4), 407–427.

Zhang, X., Choo, K. R. and Beebe, N. L. (2019), 'How do I share my iot forensic experience with the broader community? an automated knowledge sharing iot forensic platform', *IEEE Internet of Things Journal* **6**(4), 6850–6861.

Zhao, L., Kang, Y., Guo, L., Long, Y., Xing, G., Bao, M., Zhang, Y., Liu, S. and Wang, C. (2019), 'The research of alcohol drinking state analyzing based on smart watch data', *Digital Investigation* **28**(Supplement), S143.

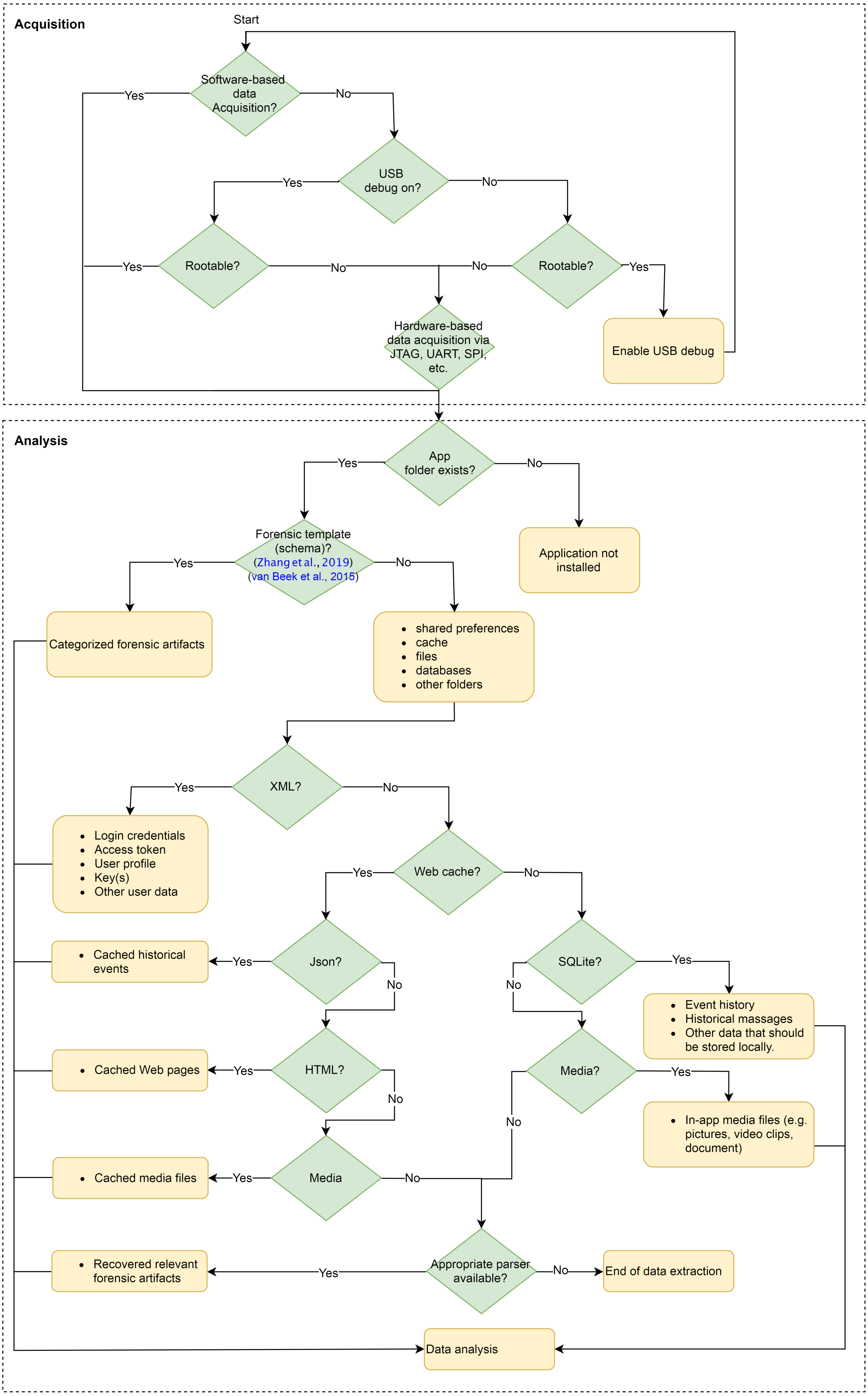

Figure 1: Android mobile app forensic framework