The Role of the Adversary Model in Applied Security Research[1]

Quang Do[1], Ben Martini[1], Kim-Kwang Raymond Choo[2,1,*]

[1] School of Information Technology & Mathematical Sciences, University of South Australia, Adelaide, SA 5095, Australia

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

* [Corresponding Author] raymond.choo@fulbrightmail.org

Abstract

Adversary models have been integral to the design of provably-secure cryptographic schemes or protocols. However, their use in other computer science research disciplines is relatively limited, particularly in the case of applied security research (e.g., mobile app and vulnerability studies). In this study, we conduct a survey of prominent adversary models used in the seminal field of cryptography, and more recent mobile and Internet of Things (IoT) research. Motivated by the findings from the cryptography survey, we propose a classification scheme for common app-based adversaries used in mobile security research, and classify key papers using the proposed scheme. Finally, we discuss recent work involving adversary models in the contemporary research field of IoT. We contribute recommendations to aid researchers working in applied (IoT) security based upon our findings from the mobile and cryptography literature. The key recommendation is for authors to clearly define adversary goals, assumptions and capabilities.

1 Introduction

Modeling the role of attackers is an integral concept in cyber defense for helping to ensure that security evaluations are scientifically valid, particularly for conceptual contributions that may not be able to be practically tested or where extensive testing is impractical.

An adversary model is a formalization of an attacker in a computer or networked system. Depending on how complete this formalization is, the adversary may be an algorithm or may simply be a series of statements with regards to capabilities and goals. There are a number of approaches in various fields of computer security that fit within this umbrella.

Adversary models are crucial in the field of cryptography where they are used in the security proof of a particular cryptographic scheme or protocol. Adversaries may be constructed with a varying number of capabilities—each of these customized adversaries is defined as a different type of attacker with different skillsets, advantages and disadvantages. Adversary models have also been used to formalize an attack on a system or protocol. One of the first and most widely used adversary models is the Dolev-Yao Model [1]. It is considered to be one of the most well-known adversary models in the field. The Dolev-Yao adversary has the ability to listen to all traffic on a network and initiate a connection with

[1] This is the pre-print version, and please refer to the following for the final version.

- Do Q, Martini B and Choo K-K R. The Role of the Adversary Model in Applied Security Research. *Computers & Security* [In press]. https://doi.org/10.1016/j.cose.2018.12.002

(and send data to) any other client on the network. The combination of these two powerful capabilities means that the adversary is, essentially, acting as a man-in-the-middle (MITM).

Many scientific fields in computer security currently utilize adversary models for verification of protocols, systems and schemes. Although this is the case, we believe that there are still many fields that can benefit from the inclusion of adversary models. For example, digital forensics is a contemporary field that benefits from the use of adversary models to prove that a forensic process is "forensically sound". McKemmish [2] defined forensic soundness in the context of digital evidence as the combination of four criteria: meaning, errors, transparency and experience. Forensic soundness is integral to the admissibility of evidence, and is analogous to the aim of maintaining data security in cryptographic models (i.e. a cryptographic protocol is flawed if data security cannot be maintained, and a forensic process is flawed if forensic soundness is not maintained). An adversary model can be used to model limitations on the abilities of a forensic practitioner to those that are permitted within the theoretical context of forensic soundness in a similar manner to the limitations of an attacker in a traditional model.

The definitions of adversary models can range from highly detailed to vague descriptions. This may end up being a problem for future researchers who wish to reuse and refine these adversary models in their own research. In fact, Bellare et al. [3] noted that due to the sometimes informal and ambiguous nature of adversary specifications (e.g. a lack of adversary goals, assumptions or capabilities), the presented protocols may not be fully proven by the formalization. They state that a "proper foundation" would be required in order to produce rigorous and strong schemes. Furthermore, the authors remarked that such foundations would need to be built up piece by piece as the usage of these models increases. Canetti and Krawczyk stated that "[i]n order to talk about the security of a protocol we need to define the adversarial setting that determines the capabilities and possible actions of the attacker" [4, p. 6] and Woo and Lam [5] noted that without explicit assumptions regarding the capabilities and execution environment of an adversary, it is very difficult to decide when to use a particular protocol and what its final state will be.

Few fields outside of secure communications (e.g. cryptography and networking) make use of fully formalized adversaries. The use of a similarly strong formalization of adversary capabilities, assumptions and goals would aid in proving the rigor of any adversarial study (e.g. proposed attack techniques and their ensuing countermeasures).

This survey paper looks at adversary models in the field of computer security beginning with the query-enhanced Bellare-Rogaway Model [6], a popular model for public key protocols in the field of cryptography. In order to determine the current state of adversary models in this field and postulate on the usage of these adversary models in both this field and its sub-fields, we perform an in-depth review of the relevant literature. Papers were selected using the following criteria:

- Paper publication date: our survey begins with the introduction of the query-enhanced adversary model of Bellare and Rogaway in 1993, with the vast majority of research works selected being published after 2000.
- Paper publication venue: with regards to comprehensiveness and inclusivity of papers, high quality papers published by prominent computer science venues were favored for this survey.

We examine the adversary models employed by researchers in three different fields of computer security, namely: cryptography, mobile, and Internet of Things (IoT). Cryptography is examined as it is the seminal field for use of adversary models and therefore has a solid foundation of research works. Mobile security and IoT security were selected as they are relatively contemporary topics within computer security. IoT security, particularly, is a research field in its infancy.

The contributions of this paper are three-fold:

- A chronological and in-depth examination of adversary models used in the field of cryptography and the evolution of this model over time. A formal adversary model definition is also proposed, which is divided into three parts.
- An initial prototype of a mobile app-based adversary model classification, which is founded on the commonly used malicious app attacker. This model can be used to classify and compare a significant number of mobile security studies.
- A series of recommendations, based on the literature, with regards to the use, classification and design of future adversary models in IoT research.

In Section 2, we provide an in-depth background on the adversary. Section 3, similarly, provides a chronological background of the use and evolution of the cryptographic adversary model. Section 4 examines and classifies adversary models used in mobile research, with an emphasis on app-based studies and Section 5 surveys adversary models used in IoT research and provides a generalized adversary model for use in future research. Lastly, we conclude the paper and discuss future research directions in Section 6.

2 Background: The Adversary

An adversary, within the field of security, refers to an attacker, often with malicious intent, undertaking an attack on a system or protocol. Typically, the goal of the adversary is to disrupt or prevent proper operation of a secure system (e.g. by violating the confidentiality, data integrity or availability of the system). Examples of non-malicious (i.e. benign) adversaries include 'guardians' (e.g. system administrators) seeking to detect decoded or encrypted packets on their own networks. There is a need to consider an adversary when designing a secure system or demonstrating the proof of a secure system in order to evaluate and benchmark the proposed system's security. Formalization of this adversary allows for the system (and adversary) to be utilized and compared, respectively to other adversaries and systems. Adversaries can be roughly divided into two categories: passive (also known as semi-honest or honest-but curious) and active (also commonly known as malicious attackers) [7].

One of the earliest formalized adversaries in the computer era was in the field of public key cryptography, with the introduction of the adversary model by Dolev and Yao [1]. This model, which we refer to as the "Dolev-Yao Model", allowed for different cryptographic protocols to be benchmarked and proven under a powerful adversary, and its principles are still in use today. Bellare and Rogaway [6, 8] (together with Pointcheval in [9]) presented a more general model, which we refer to as the "Bellare-Rogaway Model" in this paper, that allowed for modeling of different types of attackers (e.g. a benign passive attacker or a malicious active attacker) [6, 8, 9]. They also introduced the notion of queries, which are actions that the adversary may perform, including: ***Send*** (a message), ***Reveal*** (a secret, such as a session key) and ***Corrupt*** (a device, for example to obtain the internal state of the protocol participant), along with the definitions of these queries. Queries accept a number of parameters, as input, and output a result upon execution. These queries are known as adversary capabilities in this paper.

Due to the vast nature and differing types of adversaries, there is a need to model them using a method that can be customized to fit the scenario. Furthermore, different adversaries should be comparable with regards to their capabilities, goals and assumptions. Due to the overwhelming capabilities of the Dolev-Yao Model [1], it may be impractical to use this model in other areas of study (e.g. systems utilizing trusted third parties such as key escrow). Thus, we have selected the adversary model proposed by Bellare and Rogaway [6, 8, 9] as the standard for the adversary's capabilities, goals and assumptions due to the model's flexibility.

We define a fully modeled and formalized adversary to be three parts (see Figure 1), namely: the adversary's assumptions, goals and capabilities. Each adversary part can be either fully formalized (e.g.

mathematically defined using algebraic concepts) or loosely defined (e.g. textual descriptions), with the former being more rigorous and the ideal standard.

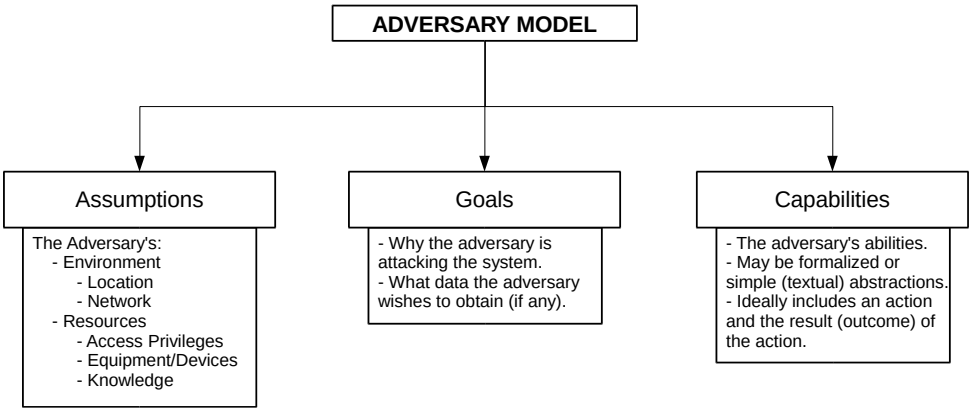

Figure 1: The three components of our adversary model definition.

By separating the adversary into three distinct parts, each part could be independently changed in order to change the strength of an adversary in the three distinct components. For example, an adversary's goal may change as a result of encountering a certain defense mechanism, such as in the context of an air-gapped system, and therefore the adversary may wish to target a different system on the same network that has Internet access. The adversary's capabilities and assumptions may also change as a result of the change in objective. The adversary components are described below.

2.1 Adversary Assumptions

An adversary's assumptions are the conjectures as to the adversary's environment, resources and equipment. For example, an adversary could be external to a system or internal (i.e. performing an insider attack) and may have privileged access on a network by virtue of being an employee (or ex-employee). It may also be the case that the adversary has access to a large number of compromised devices allowing for attacks on availability. Another aspect of the adversary assumptions are the inferences of the adversary's competency or knowledge. For example, the adversary may know a required passphrase or have knowledge of the locations of critical infrastructure (in the case of a physical attacker).

Without a clear set of assumptions, there is great ambiguity in attempting to determine how powerful an adversary is. The security proof of a protocol utilizing such an adversary would not provide much assurance to protocol implementers in practice. Furthermore, precise assumptions aid in the comparison of diverse security schemes (e.g. two protocols using the same adversary assumptions could be benchmarked in terms of their security level).

2.2 Adversary Goals

An adversary's goals refer to the adversary's intentions and to the particular sets of data the adversary is attempting to obtain (if any). An adversary may instead need to exfiltrate data they already hold in a covert manner (e.g. using steganography or inaudible sound waves). A proper set of goals is essential in building a strong and rigorous adversary model. Some general works (such as frameworks or models) may intentionally withhold an adversarial goal in order to make the proposed model more general (e.g. see [8]). In the field of cryptography, it is often the case that an adversary is modeled as a challenger to a "game" [10] whereby in order for the protocol or scheme to be considered secure, there must not exist an adversary that can win the game with a non-negligible advantage.

Without a distinct goal, an adversary is simply a malicious entity intent on wreaking havoc on the system. This, of course, is still a legitimate adversary but such an adversary is difficult to model and use to compare different studies.

Some other works may refer to the proposed system's goals and instead have an adversary seeking to hinder these system goals. A commonly used set of system goals is the CIA(A) Triad [11] which refer to the principals of confidentiality, integrity, availability (and authenticity).

2.3 Adversary Capabilities

Finally, an adversary's capabilities are the most essential part of an adversary model. These capabilities provide the adversary with tangible interactions within the context of a secure protocol or a system. Without capabilities, adversaries are may not provide sufficient means of replicability and comparability with other research in the field.

The field of cryptography employs several placeholder names to refer to different types of attackers. For example, a passive attacker (i.e. one that does not actively modify data) is often referred to as "Eve". These placeholder attackers can be mapped to adversary capabilities. For example, "Eve" may have the ***Listen*** capability that allows her to read traffic flowing through a network and "Mallory" may be a MITM attacker that is able to receive and transmit data on the network.

As previously stated, the Bellare-Rogaway Model introduced the notion of adversary queries (or capabilities). A number of commonly used adversary capabilities are as follows: ***Send***, ***Reveal*** and ***Execute***. Another commonly used adversary capability is the ***Corrupt*** capability, which allows an adversary to take control of a target and learn its internal state—an extremely powerful capability, and one that allows the modeling of a malicious insider within a system.

It should be noted that the adversary capability definition used depends heavily on the requirements of the particular adversary and that there is no overwhelming reason to use a particular standard. For example, it may not be feasible to formally define a particular action that the adversary may take due to the differences that an environment may make on the adversary. Take, for example, an adversary seeking to exfiltrate data from a mobile device by exploiting a flaw on the device, such as a root exploit. A black box style capability could be used to provide the attacker with the flaw as different devices, which may run different versions of operating systems, would have different vulnerabilities. Such vulnerabilities could also exist in different locations on the device (e.g. the bootloader or userland) or even be unknown at the time (e.g. when an attacker is targeting an unreleased device).

2.4 Threat Models

Threat models are an approach to modeling possible attacks on a system, and can be designed based on the perspectives of either a defender (e.g. an asset-centric threat model) or an attacker (e.g. an attacker-centric threat model) [12, 13]. Logically, attacker-centric threat models are the most closely related to adversary models, but the primary difference is that attacker-centric threat models are intended to model distinct attacks in detail (e.g. the steps required to perform a spoofing attack on a system). These attacks are modeled with great detail to allow the developers of a system to reinforce the system from such threats.

Adversary models represent a complete attacker with regards to their goals, assumptions and capabilities. This means that adversary models represent a much more general approach to modeling attacks on a system and, as a result, we consider the applications of threat models and adversary models to be distinct—each formalization is designed to fulfil different needs. As such, this paper does not focus on threat models.

It should be noted that some researchers may use the terms interchangeably. For example, RFC 3552 [14] notes that a threat model is used to define an attacker's methods with which they may deploy against a target and should describe resources available to the attacker, such as knowledge, processing

power and level of system control. This classification is very close to the common definition for an adversary model.

3 Background: Adversary Models in Cryptography

In this section, we provide a chronological review of the literature in order to provide the following sections (i.e. mobile and IoT adversary models) with a historic context and to aid us in providing a number of relevant recommendations.

The aim of cryptography is to provide secure communications over channels that may be insecure or have adversaries seeking to uncover these communications. Provable security, the formal proofs that demonstrate a system to be secure, plays a major role in the design of modern cryptographic protocols or scheme [6, 15]. To show that a cryptographic protocol or scheme is secure, it must be proven that a formalized adversary (with certain capabilities and assumptions/goals) can only defeat the protocol by solving a computationally infeasible problem. For example, in the well-known public key system RSA, an adversary must determine the integer factors of the public key in order to derive the plain text [16].

Oppliger [17] stated that a secure cryptographic system must provide a definition for security. He provided the minimal two questions that must be answered with regards to the adversary:

- "What are the capabilities of the defined adversary?"
- "What problem must the adversary solve in order to break the system?"

Although cryptography has been utilized and researched for an extensive period of time, the first formalized adversary model did not appear until 1983, with the introduction of the seminal Dolev-Yao model [1]. Dolev and Yao note that (in 1983) public key systems are often effective against attackers that passively eavesdrop on the system but may be vulnerable to active attacks. They proposed the use of an active adversary model for modeling these attacks and describe the capabilities of this adversary model. The adversary has almost limitless power in that it has complete control of the communication network whilst simultaneously being a legitimate user of the system. This, in turn, means that it can send messages to and receive messages from any other user. This "all or nothing" adversary model (or some aspects of it) has been used to benchmark, analyze and prove the security of many public key cryptographic protocols (e.g. see [18-22]).

In order to model, arguably, more varied adversaries, several researchers have since proposed alternative or enhanced models, with one of the most notable models being that of Bellare and Rogaway, in 1993, with their query-enhanced adversary model [6, 8, 9]. One noteworthy aspect of this adversary model is that it allows for the modeling of adversaries with varying levels of power, as opposed to the singularly powerful adversary of Dolev and Yao [1]. A fully capable (i.e. with all defined capabilities) Bellare-Rogaway adversary model [6, 8, 9] is actually more powerful than a Dolev-Yao adversary. This is primarily due to the ***Corrupt*** and ***Reveal*** capabilities (i.e. queries) that allow the attacker to take control of the device to learn its internal state and to obtain the long-term secret key and current session key, respectively. These capabilities correspond with the modeling of an insider attack on an target and a communication session, respectively.

Other noteworthy adversary models that have been employed heavily in cryptography include the Canetti-Krawczyk Model [4] and Al-Riyami and Paterson's model for certificateless cryptography [23]. These adversary models have been utilized, specialized, extended and modified by numerous researchers. We now examine the evolution of each of these models in detail and also explore other adversary models that have been used in the field of cryptography.

3.1 The Bellare-Rogaway Model

Although adversary models were first used by Dolev and Yao in 1983 [1], the adversary model itself was not significantly extended until the introduction of the Bellare-Rogaway Model in 1993 [6]. The

Bellare-Rogaway Model would become a catalyst towards a revolution of adversary models in cryptography. As Bellare, Rogaway and (in 2000) Pointcheval continued to extend the original 1993 Bellare-Rogaway Model [8, 9], numerous other researchers also implemented their own extensions in the many areas of cryptography. These areas included group key exchange and intrusion-resilient key exchange [24]. It should be noted that Bellare and Rogaway's models [6, 8] from 1993 and 1995 were constructed for key distribution protocols and Bellare, Rogaway and Pointcheval's model [9] in 2000 was designed for password-based key exchange protocols.

Half a decade later, in 2005, Abdalla et al. [25] extended the adversary model of Bellare et al. [9], which they refer to as the "Find-Then-Guess Model", and proposed the "Random-Or-Real Model" in order to prove the security of their password-based authenticated key exchange protocol in a three-party setting. Password-based authenticated key exchange protocols are designed to be secure even when secrets are selected from a limited set (e.g. passwords could be specified to be alphanumeric). The authors proved that their own Random-Or-Real Model is stronger than the Bellare-Rogaway Model even though their model has access to fewer adversarial queries. The Random-Or-Real Model is occasionally referred to as the "AFP Model" in the literature. Wang and Hu [26] later extended the Random-Or-Real Model of Abdalla et al. [25] in order to allow for the modeling of hidden or covert attacks. The adversary, which has full control of the communication network, is able to perform a number of queries. Specifically, passive, active server, active client and misuse of session key attacks are modeled by the authors.

Abdalla and Pointcheval [27] also proposed a new three-party password-based authentication protocol and provided a security proof in the Random-Or-Real Model (or the AFP Model). Initial adversary assumptions, including the state of each protocol participant, are provided along with a formal definition of the adversary's advantage (i.e. the adversary's goal). This adversary model has only a subset of all the capabilities in the original Bellare-Rogaway Model [6, 8, 9], thus making the adversary significantly weaker. Farash and Attari [28] later utilized an adversary with full control of the communication network in the security proof of their three-party password-based authenticated key exchange protocol, where the adversary model was constructed from the adversary model used by Abdalla and Pointcheval [27].

Authenticated group key exchange protocols allow a group of users in a network to share a common secret and also be guaranteed that they are sharing this secret with the intended recipients [29]. In this sub-field, in 2001, Bresson et al. [30] presented a security model for group Diffie-Hellman authenticated key exchange protocols. The authors utilized an adversary model that was derived from that of Bellare et al. [9]. The authors suggested that their model was one of the first formal security models for group Diffie-Hellman authenticated key exchange. Roughly a year later, in 2002, Bresson et al. [31] made further modifications to this evolved model and also used concepts from the adversary model of Bellare et al. [9]. This heavily evolved model defined a number of adversarial queries and was used in their research in order to model an adversary with the ability to perform dictionary brute-force attacks. In 2005, Byun and Lee [32] considered an N-party Diffie-Hellman key exchange scenario involving the use of different passwords for each party, similar to the group key exchange protocols previously mentioned. As part of this specific use case, they contributed a specialized adversary model based on the combination of the adversary model of Bresson et al. [30] and the model of Bresson et al. [31]. Subsequently, in 2007, Bresson et al. [33] extended their previous adversary model [31] (which was, in turn, an extension of Bellare, Pointcheval and Rogaway's model [9]) in order to provide a security proof for Diffie-Hellman key exchange in IEEE 802.11 ad-hoc mode. Abdalla and Pointcheval [34] utilized this adversary model [33], which was published a year after this work, for their group key exchange protocol. The authors note that their model does not take into account the ***Corrupt*** adversary query as defined by Katz and Yung [29] and considered this a significant limitation of their proposed model.

Katz and Yung [29] defined an adversary model similar to the Dolev-Yao model [1] in that the adversary has complete control of the communication network and could initiate connections with other parties.

This adversary model was further extended in that it also had specific capabilities (based on the variables used by Bellare et al. [9]). A weaker, passive adversary was also defined with a subset of the capabilities.

Other works that derive from or extend the Bellare-Rogaway Model include the work of Cash et al. [24], who directly extended the adversarial queries of Bellare, Pointcheval and Rogaway's model [9] in order to demonstrate their intrusion-proof authenticated key exchange protocol. As a consequence of their study, the authors introduced a new capability to the Bellare-Rogaway Model: the ***Intrude*** capability. Chen et al. [35] utilized a modified version of the Bellare-Rogaway Model [6] to analyze the security of a number of identity-based key agreement protocols that exist in the literature. Inspired by this approach, Zhang et al. [36] proposed their own adversary model based on that of Chen et al. [35] for use specifically in certificateless cryptography.

Interestingly, in the 2010s, there exist a number of works that directly modify the original Bellare-Rogaway Model [6] in their works (i.e. directly extending or modifying the 1993 Model and they do not utilize any of the extended models that were derived from the previous decade). For example, in 2011, Brzuska et al. [37] presented a framework to aid in the security analysis of cryptographic protocols, specifically by using a game-based approach. The authors utilized an adversary model that was heavily based on the Bellare-Rogaway Model [6, 8, 9], and made use of their adversarial queries.

More recently, in 2015, Dowling et al. [38] provided an analysis of the security of the Transport Layer Security (TLS) version 1.3 protocol involving a Bellare-Rogaway style query-based adversary model that was based on the work by Brzuska et al. [37]. The adversarial queries provided were constructed specifically for TLS 1.3. Jager et al. [39] also examined the security of TLS and provided a formal security proof for cipher suites that were based on Diffie-Hellman key exchange [40] (known as TLS-DHE). The adversary model utilized was based on the Bellare-Rogaway Model [6], and a set of formal goals (denoted as "games") and assumptions (denoted as the execution environment). Krawczyk et al. [41] later analyzed the security of the TLS handshake protocol using a generalized approach. Part of this approach involved the construction of a modular adversary model, which was heavily based on that of Bellare and Rogaway [6] and Jager et al. [39].

3.2 The Canetti-Krawczyk Model

Bellare et al. [42] presented a framework for the analysis and creation of realistic cryptographic protocols in 1998. Two adversary models, called the "Authenticated Links Model" and "Unauthenticated Links Model", were proposed where the former adversary cannot modify messages on the network but may choose to withhold incoming and outgoing messages to clients. The latter adversary is able to fully control the communications network, similar to the Dolev-Yao model. The adversaries do not have any explicit goals or assumptions, which may be due to the fact that the work proposed by the authors is a general framework used for the construction of protocols rather than a proposed protocol and its security evaluation.

Later, in 2001, Canetti and Krawczyk [4] extended the "Unauthenticated Links Model" and the "Unauthenticated Links Model" of Bellare et al. [42] in their proposed formalization for the analysis of cryptographic protocols, specifically for key-exchange protocols. These extended adversary models are specifically designed for use in key exchange protocols and are commonly known in the literature as the Canetti-Krawczyk Model. It was not until 2007, when LaMacchia et al. [43] extended the Canetti-Krawczyk Model that it would see a major transformation. The modifications to the model by LaMacchia et al. [43] also included concepts of the Bellare-Rogaway Model [6] and could capture key leakage attacks, specifically ephemeral and long-term secret keys. This model is commonly known in the literature as the extended Canetti-Krawczyk Model or the eCK Model. Subsequently Yoneyama [44] strengthened this extended model to allow it to fully capture the security requirements of three-party password-based server aided key exchange. This adversary was given a number of additional capabilities including the ability to discretely obtain static and short-term secrets. Yoneyama [44] also

noted that some commonly used models, such as those of Wang and Hu [26] and Abdalla et al. [25], may not fully consider certain security notions such as that of forward secrecy. Lippold et al. [45] presented an extension to the work of Swanson [46], which was in itself an modification of the extended Canetti-Krawczyk Model of LaMacchia et al. [43]. The adversary in this model is more powerful than the original eCK Model adversary in that it is able to compromise a system whilst having less secret knowledge (the eCK Model original worked with four "pieces" of information whilst this model had a total of six). Although Lippold et al. [45] proposed a model for certificateless key exchange protocols, they did not utilize the original certificateless cryptography adversary model of Al-Riyami and Paterson [23]. Swanson and Jao [47] later proposed another modification of the extended Canetti-Krawczyk Model [43]. The authors divided adversaries into either inside or outside attackers. Inside attackers have access to the master secret key can cannot replace the public key of users. Outside attackers do not have access to the master secret key and are able to replace the public key of users. The authors also explained one of the major differences between their extension and that of Lippold et al. [45] was that their model was strictly weaker.

Other modifications to the eCK Model include those of Wang and Zhang [48] who made use of the authenticated and unauthenticated links adversary models for their certificateless session initiation protocol. Adversary goals were specified as a series of security goals and a definition of the adversarial advantage for the protocol. Although Wang and Zhang's work was based on the concept of certificateless public key cryptography, which was introduced by Al-Riyami and Paterson [23], it utilized a disjoint adversary model.

3.3 The Al-Riyami-Paterson Model

In 2003, Al-Riyami and Paterson [23] proposed the concept of a certificateless public key system, which removed the need for a key escrow to have knowledge of the private key. In this work, the authors defined a number of adversary capabilities in their adversary model, which they refer to as algorithms, and considered the case of a general adversary who utilizes all defined capabilities along with two (weaker) adversaries that employed different constrained subsets of these capabilities. Of these weaker adversaries, the Type I adversary is able to request and replace public keys whilst the Type II adversary does not have this ability. The defined algorithms are specified precisely for an adversary targeting a certificate-less public key system and, as such, may not be applicable in other areas. For instance, the adversary is able to extract a subset of the victim's private key, which may not be feasible in a typical public key system. We refer to this model as the Al-Riyami-Paterson Model.

Yum and Lee [49] provided a generic security model for the certificateless cryptographic protocol proposed by Al-Riyami and Paterson [23] (which, they noted, was lacking such a model). Their model extended on the definitions of the Type I and Type II adversaries, which were originally offered by Al-Riyami and Paterson [23]. Hu et al. [50] later determined that the certificateless security model provided by Yum and Lee [49] was, in fact, insecure against a specific type of key replacement attack. The authors provided a modified adversary model that, not only, resolved the flaw but also assisted in simplifying the model.

Huang et al. [51] also determined that the model provided by Al-Riyami and Paterson [23] for certificateless cryptography was insufficient in terms of fulfilling the security requirements of certificateless cryptography. They provided an extension to the Type I and Type II adversary models proposed by Al-Riyami and Paterson [23]. Zhang et al. [52] later noted that the security assumption of the adversary of Huang et al. [51] may, in fact, be too strong to be considered reasonable in practice.

Several years later, Bentahar et al. [53] examined the Type I and Type II adversaries of Al-Riyami and Paterson [23] and proposed their own, weaker, version of the Type I adversary called the Type I^{-} adversary. They, and a number of other researchers [49, 54], have noted that the original Type I adversary is not feasible in a real-life situation. One of the capabilities of a normal Type I adversary is

the ability to perform decryption without the knowledge of the secret key. The weaker Type I⁻ adversary instead cannot perform decryption without knowledge of this secret key.

Dent [55] performed a survey of the literature with regards to schemes and security models used within certificateless cryptography. The survey begins with the first work in this area, that of Al-Riyami and Paterson [23], up to works performed in 2008. The author categorized the typical Type I and Type II adversaries used in certificateless cryptography into, ultimately, six different types of adversaries: four Type I adversaries and two Type II adversaries.

3.4 Other Cryptographic Adversary Models

It is often the case that researchers choose to utilize their own adversary models, which have specialized capabilities and resources, in order to complement their research. For example, Hopper and Blum [56] presented several defined goals suitable for use in secure human identification and demonstrated the feasibility of these definitions. The authors utilized two adversary models in the course of their proof: a passive adversary and a malicious adversary (referred to by the authors as an arbitrary adversary). The malicious adversary used was able to control the communications between all parties. Unlike the previously described adversary models, specific adversary capabilities or queries were not outlined in this model.

Bellare et al. [3] presented their own adversary model as part of their foundations for dynamic group signature-based protocols. Similar to the Bellare-Rogaway Model, the adversarial queries were fully modeled in this study.

Huang et al. [57] categorized adversaries into three levels in increasing order of adversarial knowledge. For example, in their analysis of the short proxy signature security scheme, a Type I adversary has only the public keys of A and B, whilst a Type II adversary has, in addition, the secret key of B and a Type III adversary has, instead, the secret key of A. The aim of each of the adversaries was to forge a new valid signature. Wu et al. [58] also utilized a three level adversary model that was based on the archetype proposed by Huang et al. [57] in their identity-based proxy signature scheme.

Barbosa and Farshim [59] applied an alternate categorization of adversaries and also categorized them as Type I and Type II in their certificateless cryptography system. A Type I adversary in this work is a typical user of the system and does not possess any secret keys. The Type II adversary in this model was an honest-but-curious adversary.

Aumann and Lindell [60] argued that there may be security scenarios where passive or honest-but-curious adversaries may be too weak, but malicious adversaries may conversely be too strong to defend against. They introduced the notion of a “covert” adversary which is an active adversary that seeks to hide their existence from the target of their attack. The primary assumption with regards to the covert adversary is that any messages that it sends would be indistinguishable from an honest user’s messages within the network.

Geng and Zhang [61] noted that a number of protocols would be insecure when used with modular models such as those of Bellare and Rogaway [6] and Canetti and Krawczyk [4] (as determined by Lippold et al. [45]). Thus, the authors proposed a new formalized security model for protocols that are not dependent on pairing. Two types of adversaries are modeled: Type I adversaries do not have the master key but have the ability to replace the public key of any user with a key of their choice and Type II adversaries have access to the master key but cannot replace public user keys. Each type of adversary was also given a number of queries reflecting their capabilities as previously described.

An in-depth tutorial on privacy-preserving secure multi-party computation was provided by Lindell and Pinkas [62]. As part of this discussion, the authors provided a set of basic assumptions (referred to as “behaviors”) for the adversary, which were mapped directly to both honest-but-curious and malicious adversaries. Zhao et al. [63] demonstrated a key leakage flaw on an existing protocol [64] and proposed

an improvement to this group key agreement protocol. The adversary model utilized by Zhao et al. [63] was comparable to the Dolev-Yao Model.

3.5 Trends in Cryptography

What began as a paradigm shift into the modeling of adversaries in order to prove the security of cryptographic protocols in 1983 with the Dolev-Yao Model [1] has developed into fully fledged query-enabled adversaries such as those of Bellare, Rogaway [6, 8] and Pointcheval [9]. These models have been further modified to support the requirements of different sub-fields, such as authenticated key exchange [29] and TLS [38, 39, 41], and to model new attacks (e.g. intrusions [24]). These additions sometimes included the inclusion of new adversary queries, such as the ***Intrude*** capability proposed as part of the model of Cash et al. [24].

Figure 2 presents the above review of the literature in a chronological flowchart format.

It is important to note that there is no complete advantage or disadvantage to a certain adversary model utilizing (or not utilizing) a particular component – additional features may serve to make a model more detailed whilst simultaneously reducing the flexibility and reusability of the adversary model. On the other hand, a simple adversary model may be easier to understand and more suitable for use in other research.

From our review, it is clear that adversary models have matured and are a crucial aspect of cryptographic research, particularly in the security proofs of protocols. Other security-based research should look to cryptographic protocols as the gold standard for adversary models as the vast majority of adversary models used in this field are fully realized.

One still under-utilized adversary in cryptographic research is the physical attacker, which, as IoT and cyber-physical systems become more and more important and omnipresent, may become an important factor in future cryptographic research. In the following sections, we examine the adversary models currently used by IoT and mobile researchers and the problems that these models may encounter in the future.

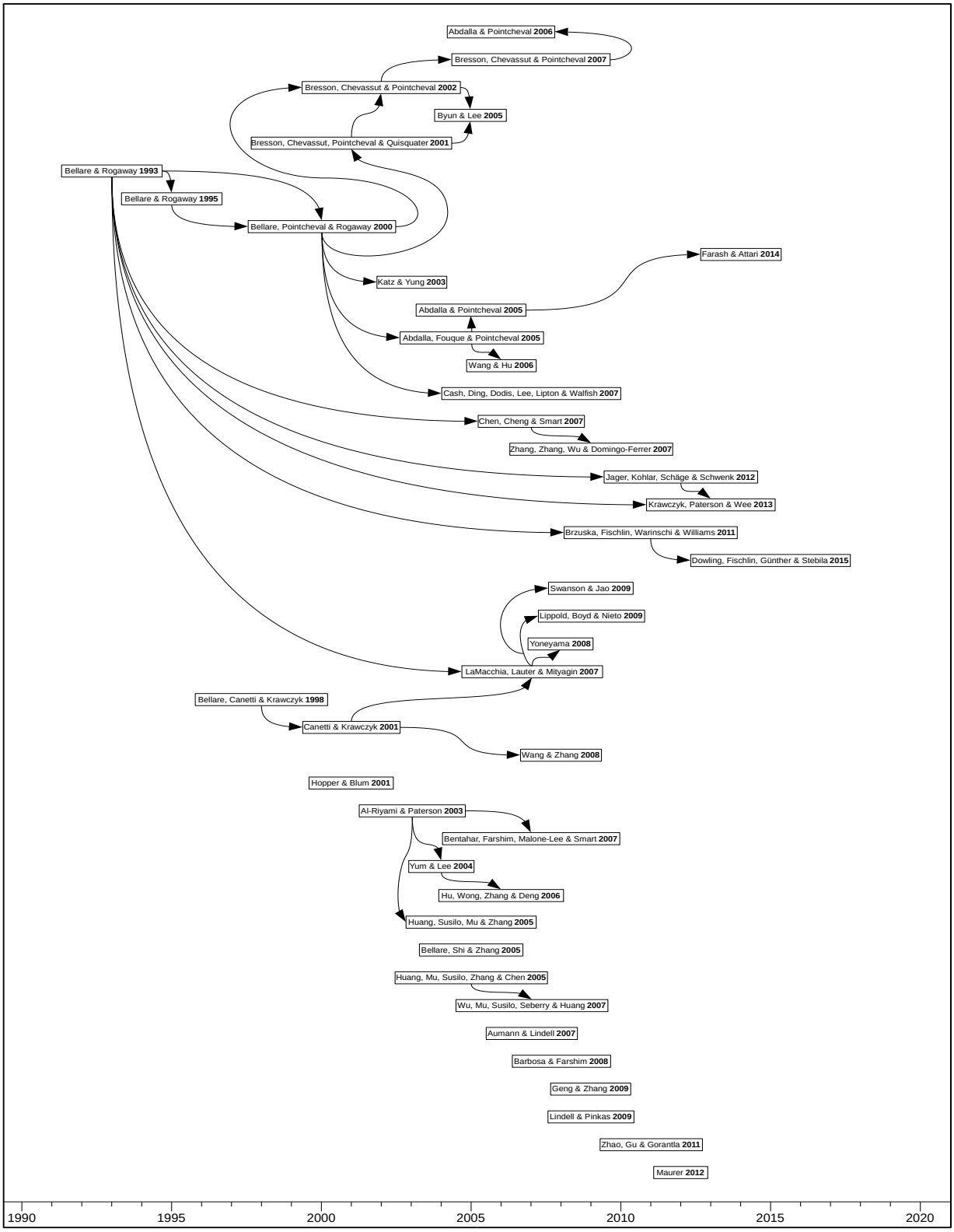

Figure 2: The evolution of the adversary model in cryptography research.

4 Mobile Adversary Models

Mobile security research can, roughly, be divided into two periods with the point of division being the introduction of the modern smartphone. Smartphones are defined as mobile phones that have complete support of third-party apps and the capabilities that this entails (see Table 1). Our review focuses on research performed targeting and utilizing contemporary smartphones, such as Android and iOS devices. In this section, we consider adversary models used within the field of mobile research and not the similarly termed mobile adversary model proposed by Ostrovsky and Yung [65], which involves the modeling of adversaries interested in the destruction of systems (i.e. viruses).

Mobile Phone Type	Level of Sensitive Information	Examples of Sensitive Information
Standard Phone	Medium	• Phone call logs • SMS messages • Contact phone numbers
Feature Phone	High	• Personal photos • Phone call logs • SMS and MMS messages • Contact information • Web browsing history • Limited third-party app information (e.g. social networking apps) • Location information
Smartphone	Very High	• High resolution personal photos • Phone call logs • SMS, MMS and instant messaging messages (e.g. Skype, WhatsApp, LINE and WeChat) • Contact information • Web browsing history • Third-party app information (e.g. dating, social networking, banking, email and shopping apps) • Accurate location information

Table 1: A comparison of the amount of sensitive information contained on different mobile phone types.

4.1 App-Based Adversary Models

Mobile malware is a common area of research in mobile security that makes use of adversary models. Adversary models are very useful in this area of research as they provide researchers with a framework and capabilities with which an attack may be modeled and therefore, potentially, prevented. Mobile malware research can be categorized as research into mobile device or app vulnerabilities and associated countermeasures or systems to detect or prevent malware infections on devices.

Third party apps are one of the primary attack vectors in smartphone security research. This may be due to the prevalence of these apps and the fact that they are easily obtainable by the end user. This lead to the development of the app-based adversary model.

Davi et al. [66] demonstrated one of the earliest privilege escalation attacks on Android devices and considered what they termed a “strong” adversary model. The adversary was assumed to be a non-malicious app, which was also flawed in such a way that an attacker could exploit the app to use permissions not previously granted to the app. The adversary’s goal was to send SMS messages to a premium-rate number. While the authors stated that they assumed a strong adversary model, they did not provide further details that could allow for assessment of the study by a third party.

In their more recent work, Bugiel et al. [67] proposed a system which was capable of monitoring communications between apps in an effort to detect and prevent privilege escalation attacks on Android devices. The authors categorized privilege escalation attacks into two types, namely: confused deputy attacks and attacks via application collusion. The former attack type involved a malicious app misleading a previously installed benign app in order to access resources which the malicious app was not originally permitted. The latter refers to a number of malicious apps colluding in order to gain permission or resource sets unapproved by the user. Similarly, Luyi et al. [68] considered a malicious app adversary model that initially requested no dangerous permissions but may later request dangerous permissions via an app update. This model was used by the authors to demonstrate a privilege escalation attack on Android devices.

On mobile operating systems, such as Android, apps are often required to request a specific permission to a protected resource before that app is granted access to this resource. This system works as a whitelist—by default, no permissions are granted to an app. For example, a camera app would require, at the very least, permission to use the device's camera and to write to device storage. Resource permissions on most mobile platforms, such as the Android operating system, can be categorized into either "normal" or "dangerous" permissions, with the former concerning access to data with little risk to the user's privacy and the latter relating to access of potentially privacy-endangering resources. The permissions presented to a user upon installation of an app consist solely of dangerous permissions and normal permissions are implicitly granted when requested by an app [69].

4.1.1 The Proposed App-based Adversary Model Classification

We have observed that app-based adversary models used in the literature can be categorized into three types, namely: adversaries that request zero permissions, adversaries that request only normal permissions and adversaries that request dangerous permissions (along with normal permissions). The level of permission inherently limits the capabilities of the adversaries and could be formalized as a query-based adversary model (ala Bellare-Rogaway). We provide an initial formalization of this particular adversary model classification in Table 2 and examine its use cases below.

App-based Adversary Model Classification	
Adversary Assumptions	Adversary assumptions for the app-based adversary model can be classified into one of three increasingly powerful categories, namely: **zero permission**, **normal permission** and **dangerous permission**. The classification depends wholly on the permissions required by the app-based adversary model (analogous with the adversary capabilities in the case of this proposed adversary model). In addition to the above three classifications, adversary assumptions include the target operating system (e.g. Android or iOS), OS version (e.g. iOS 9.0 or Android 6.0) and device model (e.g. iPhone 6S or Galaxy S7). Furthermore, the app-based adversary may be a remote (e.g. Internet or LAN-based) or a physical attacker.
Adversary Goals	This particular section of the proposed adversary model differs greatly based on the research objective. Examples of common adversary goals in the literature include the collection of sensitive user information and user files.
Adversary Capabilities	The adversary capabilities are the most important part of the proposed app-based adversary model as the capabilities directly affect the adversary assumptions. In the app-based adversary model, the adversary's capabilities are the permissions it requests from the OS. For example, an app-based adversary model may request the permission required to send and receive data via the Internet and, as such (based on Google's classification scheme [69]), the adversary model is classified as a normal permission app-based adversary model with the adversary capability being the Internet permission.

	In many cases, app-based adversaries are able to perform many tasks even without requiring access to any system permissions (e.g. accessing files stored on the device's external storage). This is due to the fact that the model encompasses the capabilities of an app installed on a particular system, and apps often require a certain level of access to system resources in order to be functional.

Table 2: The proposed app-based adversary model for classifying models used in the literature.

The value of this adversary model classification comes from the fact that it allows different studies to be compared with each other based on the type of app-based adversary model used. Furthermore, the classification allows us to determine and catalogue the defining characteristics of app-based research in the field. We now adopt the zero, normal and dangerous permission app-based adversary model approach for categorizing and classifying the existing research in this field in the remainder of the paper. The studies discussed in the following subsections do not all explicitly use an adversary model, however we have classified their technical approaches based on the adversary model proposed above.

4.1.2 Zero Permission Adversaries

Zero permission adversaries are, seemingly, some of the least commonly used app-based adversary models in the literature. These adversaries cannot request any direct resources that are enforced by the operating system and are only granted permission to basic resources. For example, obtaining a list of installed apps on an Android device requires no explicit permissions to be requested by an app and producing sound via the device speaker typically does not require the use of a system permission [70]. Implicitly, due to the nature of the permissions system, normal permission adversaries have access to the capabilities of zero permission adversaries and dangerous permission adversaries have access to the capabilities of both the zero and normal permission adversaries (see Figure 3).

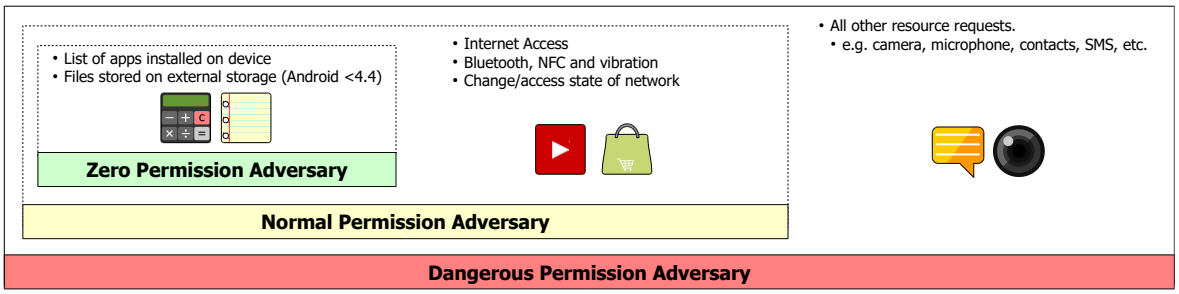

Figure 3: A comparison of the app-based adversary models used in the literature.

It should be noted that zero permission apps running on Android versions below 4.4 KitKat, released in mid-2014, are able to access the device's external storage whilst apps running on Android version 4.4 and above require a (dangerous) permission to perform the same task [71]. Such changes mean that zero permission adversaries considered before this alteration may be vastly more powerful than those proposed after Android 4.4. Furthermore, Android 6.0 moved app permission requests to run-time (e.g. when an app is launched) instead of when an app is installed [72]. In addition, the user is given the ability to deny or allow certain permissions when the app is launched. As a result, the app-based adversary model would require that its adversary assumptions reflect this requirement.

In 2013, Zhou et al. [73] demonstrated that sensitive user information (such as information that could be used to infer the user's location) could be obtained from an Android device via a malicious app which requested no permissions. This zero permission malicious app formed the basis of their adversary model. This app was able to collect and analyze information contained on the device (which it was able

to access whilst simultaneously requiring zero resource permissions). One of the capabilities of the adversary was a zero permission method which could transmit data via the Internet using HTTP GET payloads.

A zero permission malicious app adversary was also utilized by Diao et al. [74] in their proposed Android-based attack which made use of the preloaded Google Voice Search service, present on the majority of Android smartphones, to obtain sensitive user information. This attack was feasible as Android does not require an app to request permission to transmit sound via the speaker and as such the attackers were able to issue voice commands to the phone via this channel. Another covert attack was implemented by Lipp et al. [75] who performed a number of non-root covert channel attacks on Android devices. These attacks aimed to collect and exfiltrate user data, such as keystrokes, in order to infer sensitive user information. They considered a malicious app adversary model with no requested permissions executing on a non-rooted device which, in addition, did not exploit Android version-specific vulnerabilities.

Meanwhile, Meng et al. [76] introduced a type of power charging attack for smartphones which they referred to as "juice filming attacks". The attack involved performing automatic screen-captures of the device, during device charging, of sensitive information such as PINs, patterns or passwords. The authors considered three adversary scenarios in their adversary model, namely: a public charging station, a semi-public charging station (e.g. a hotel or rented apartment) and a borrowed charger. The screen capturing itself did not require any permissions or apps to be installed onto the device in order to function and perform the attack but a physical attack vector was necessary (i.e. the charging station). In another side-channel attack, Spreitzer et al. [77] demonstrated that a zero permission adversary could accurately infer sensitive information (e.g. websites visited by the owner of a device) via analysis of the proc filesystem (procfs) present in many operating systems, including Android.

The zero permission adversaries discussed thus far can be classified under our definition of the zero permission adversary model (described in Section 4.1.1). These models are summarized in Table 3. Furthermore, all of the adversary models discussed were textually defined without explicit adversary queries or capabilities (e.g. see Section 2). We have discovered that a number of researchers refer to the "latest version of the OS" or the device model instead of providing specific OS version numbers. As a result, the adversary assumptions made by these researchers is unclear and must be determined based on circumstantial details (e.g. time of publication and the default version available to the specified device model). Zero permission adversaries can be considered the most restricted as well as the most difficult to defend against, from the point of view of an OS developer, as a user cannot arbitrarily change the permissions that the app has access to due to the fact that it does not explicitly request any granular permissions.

Zero Permission Adversary Model	Adversary Assumptions/Capabilities	Adversary Goals
Zhou et al. [73]	• Android 4.1.1 was, presumably, considered by the authors based on the devices they utilized. This means that, for example, the adversary is able to read files on external storage.	• Collect sensitive user information.
Diao et al. [74]	• Android 4.1 and newer were, presumably, considered by the authors as, according to the authors, only such systems are able to run Google Voice Search.	• Collect sensitive user information.
Lipp et al. [75]	• Android 4.4.1 to 5.0.1+ (based on the devices utilized by the authors). The authors do not specifically state the versions of the Android OS for the devices under consideration.	• Collect user keystrokes and touchscreen inputs.
Meng et al. [76]	• A compromised charger or charging station is assumed by the authors—no app-based adversary model is utilized in this study.	• Collect user touchscreen video and images.

Spreitzer et al. [77]	• Android 7 (N) and Android 8 (O) systems. Two Android phones and one emulator were used in this study.	• Infer user activities, including web browsing, videos watched, etc.

Table 3: A summary of the zero permission adversary models reviewed.

4.1.3 Normal Permission Adversaries

Normal permission app-based adversary models refer to the use of malicious apps that do not request sensitive (i.e. dangerous) permissions from the user. This classification is based on that used by Google's Android OS [69]. Although this classification is rather nebulous, it allows researchers to construct an adversary model that is stronger than a zero permission adversary but without the level of power available to a dangerous adversary. It should be noted that both the normal and dangerous permission adversaries are capable of accessing resources outside of an app's sandbox—the zero permission adversary model is the only model constrained to the sandbox by the OS. Examples of permissions commonly considered to be at the "normal" level include: Bluetooth, Internet, NFC and device vibration. As is immediately apparent, these permissions are not quite benign.

Zhou and Jiang [78] studied two specific vulnerabilities that allowed apps to obtain sensitive data and to modify in-app configurations. In order to perform their attacks, the authors proposed an adversary model consisting of a malicious app which did not request any Android-designated dangerous permissions, but was allowed to request "normal" permissions. This work, published in 2013, seems to be one of the earliest examples of the use of a normal permission app-based. Wu and Chang [79] demonstrated indirect file leakage in mobile apps on both Android and iOS devices. They made use of an adversary model comprising three distinct adversaries, namely: an installed app with zero or few permissions, a local network (i.e. LAN) attacker and an Internet-based attacker.

An indispensable smartphone app is the software keyboard and, on Android devices, the keyboard app can be changed by the user. Diao et al. [80] studied the potential for adversaries to obtain the user's customized dictionary, stored by their keyboard app, via cross-app injection attacks. The authors utilized the malicious app adversary model with the app requiring the WAKE_LOCK and INTERNET permissions (two "normal" permissions). Furthermore, the adversary's operating environment (e.g. Android 2.3.7 and up to Android 5.0) and aims (to collect the user's commonly used words or sensitive words) were provided by the authors.

Cheng et al. [81] introduced a method for obtaining a subset of the root privileges available on Android without requiring the device be rooted (e.g. unlocking of the bootloader or via an exploit). The authors considered the situation where a malicious app (with only Internet access) would use this method in order to attack the device and modeled this with the help of a loosely defined adversary model.

Covertly obtaining device screenshots in order to acquire sensitive on-screen information is a common attack vector in mobile security research. Lin et al. [82] proposed Screenmilker, a malicious app that was able to periodically and covertly take screenshots of the device in order to obtain sensitive user information that was displayed on-screen. As with many normal permission adversaries, a malicious app adversary model with the INTERNET permission was utilized by the authors.

Naveed et al. [83] presented one of the earliest works (for Android devices) on "external device mis-bonding" in Bluetooth-enabled Android devices. This security flaw refers to the use of spoofed devices, which resembled legitimate health devices to their respective smartphone apps. This allowed the device to obtain sensitive user information. The authors considered two types of Bluetooth-based attacks for their adversary model: a malicious app (which requested the two Android Bluetooth permissions) and a malicious fake Bluetooth device.

Vendor customization is often considered by researchers to be a potential attack vector for malicious attackers. Aafer et al. [84] examined the inadvertent threat of Android operating system vendor customizations by analyzing 97 different vendor factory images. These customizations could allow for malware to exploit a device and obtain sensitive user data. As with some other mobile security works, the authors considered a malicious app adversary model, specifically a normal permission adversary that requested network communication capabilities in order to transmit the sensitive user data obtained.

Similar to the work of Aafer et al. [84], Wu et al. [85] considered the effects of vendor customization on the security of the Android operating system. The authors analyzed a total of ten different Android OEM images and noted a number of vulnerabilities. They utilized the malicious app adversary model with the malicious app able to request any non-sensitive permission. The adversary's goal was to exploit vulnerable vendor apps and indirectly obtain sensitive information in this manner. Likewise, Zhou et al. [86] also considered the dangers of OS customization by manufacturers. They specifically looked at what types of sensitive information could be obtained from a malicious app adversary model where the app itself was not allowed to request the permission which it sought to obtain sensitive data from. For example, the malicious app may try to obtain audio recordings of the user but may not request the (dangerous) RECORD_AUDIO permission in order to do so. The app was allowed to request other unrelated permissions.

We observed that there are significantly more normal permission adversary models proposed in the literature than zero permission adversary models. Furthermore, the collection of sensitive user information and files is the primary goal of the vast majority of the research surveyed thus far. These findings are summarized and categorized in Table 4.

Normal Permission Adversary Model	**Adversary Assumptions/Capabilities**	**Adversary Goals**
Zhou and Jiang [78]	• Android 4.2 is presumed to be the OS utilized based on the devices examined. • A malicious app which was allowed to request any non-dangerous permission.	• Collect sensitive user information and modify app settings.
Wu and Chang [79]	• iOS 7 and 8, and an unknown version of Android were considered by the authors. • The adversary also does not perform any screenshot attacks.	• Collect sensitive user files.
Diao et al. [80]	• Android versions 2.3.7 up to 5.0 were considered by the authors. It is unclear if all versions in this range were utilized by the authors. • The INTERNET and WAKE_LOCK (normal) permissions are required in order to exfiltrate the data and allow this to occur even whilst the device screen is off, respectively.	• Collect user dictionaries in order to infer sensitive information (e.g. passwords and usernames).
Cheng et al. [81]	• Android 4.x to 5.x were specified by the authors as the platforms utilized. • The malicious app requests only the INTERNET permission.	• No formal adversary goal was provided by the authors.
Lin et al. [82]	• Android 4.1.1 was specified by the authors. • The malicious app requests only the INTERNET permission.	• Collect device screenshots containing sensitive user data.
Naveed et al. [83]	• Android 4.2 was specified as the platform considered by the authors. • The malicious app requests two normal permissions: BLUETOOTH and BLUETOOTH_ADMIN.	• Collect sensitive user information. • Inject fake data into device apps.
Aafer et al. [84]	• Android versions 4.0.4 up to 5.0.2 were examined. • A total of 97 vendor stock images were examined by the authors and, as such, the adversary assumptions of the adversary would change based on the OS.	• Collect and exfiltrate sensitive user information.
Wu et al. [85]	• A number of Android OS versions between 4.0.3 to 4.3 were considered.	• Locate vulnerabilities that would allow for: stealing

	• The malicious app does not request any "sensitive" permissions. Presumably, the authors equate dangerous and sensitive permissions.	of the user's money, collecting of the user's sensitive information and destroying data in a malicious manner.
Zhou et al. [86]	• Android OS versions 4.2 and 4.3 were considered by the authors. • The malicious app was unable to directly request an app with which it wanted to obtain the data from (e.g. in order to reduce the user's suspicions). Due to this restriction, we place this particular model under the normal permission adversary model classification.	• Obtain access to an unrequested permission resource through the use of vulnerabilities.

Table 4: A summary of the normal permission adversary models reviewed.

4.1.4 Dangerous Permission Adversaries

The dangerous permission app-based adversary model is another popular model used in the literature. This model allows for the simulation of a typical smartphone app, which can be considered one of the most common attack vectors on mobile devices. Typically, users download and install apps from official (and unofficial) app stores and these apps are highly likely to request access to sensitive resources. The apps must explicitly request this resource and the user is required to agree to these resource requests upon installation of the app. Some versions of the Android OS allow for the user to block requests to certain device resources in a fine-grained manner. Resources which are considered dangerous include: reading/writing contacts data, sending/receiving SMS messages, initializing phone calls and recording of microphone audio.

One of the earliest works to use this particular adversary model was Luo et al. [87] who, in 2011, considered two such models: a malicious app and a malicious webpage, in their research which examined webview attacks on Android. As opposed to other malicious app adversary models, which tried to directly attack the device, this particular adversary model targeted specific web applications (e.g. Facebook). The malicious webpage model, meanwhile, sought to attack benign apps in order to coerce the victim app into opening the attacker's webpage (as opposed to the legitimate web application). The goal of the adversary, beyond the compromising of the targeted web application, was unclear.

Due to the widespread reliance on app sandboxing on mobile operating systems, this in turn has become a common attack vector for malicious apps. A number of works have endeavored to strengthen the mobile operating system's sandboxing defenses. AirBag [88] is a virtualization technique for malware resistance on Android devices. Apps on the device were executed in an isolated environment to prevent them from leaking sensitive data or infecting the operating system itself. An adversary model involving malicious apps that attempted to exploit device vulnerabilities was considered by the authors. Permissions requested by the malicious apps were not considered by the authors as their system was able to completely isolate an app and therefore sandbox the app at a lower level than the permissions system, but (presumably) apps would be able to request any legitimate permission.

Zhang et al. [89] introduced App Guardian, a technique to defend against attacks which attempted to infer or steal private user data during an app's execution. This attack was termed by the authors as "runtime information gathering (RIG)". App Guardian itself was, interestingly, not an operating system modification but rather a normal Android app which was similarly restricted by Android sandboxing. Nonetheless, the authors claimed that their approach was able to secure devices from RIG attacks. With regards to the adversary model, the authors employed the malicious app adversary model, specifically apps that may request any non-system permission. It should be noted that one permission required by App Guardian has since (see [71]) been elevated to above a dangerous permission into a "signature" level permission. Such permissions are typically only requested by system apps with a small number

available to third party apps, but this requires the user to manually configure this setting in the Settings app.

Similarly, AppCage, introduced by Zhou et al. [90], was a system for complete sandboxing of third-party Android apps in order to protect devices against privacy leaks. As with many other protections for app data leakage, the authors made use of the malicious app adversary model. Due to AppCage operating as a wrapper containing the malicious third-party app, the malicious app was able to request any resource permission. However, the authors did not specifically provide the goal of the malicious app. Yan et al. [91] presented SplitDroid, another OS-level modification to the Android OS that enhanced device security by providing an isolated environment for app execution. The system was designed around an app-based adversary model involving a malicious app with the ability to compromise all parts of the user space. Presumably, this app was able to request any non-system resource permissions. Meanwhile, Xing et al. [92] examined cross-app resource attacks on iOS and OS X devices (which also employed app sandboxing). The authors utilized a malicious app adversary model involving a malicious app, requesting a small number of innocuous permissions, which was uploaded to the Apple app store and underwent Apple's reviewing process. The adversary's goal was to collect sensitive user data and access data owned by other installed apps.

Mutchler et al. [93] investigated the dangers of Android apps which employed webviews (embedded browsers inside apps with, possibly, reduced functionality). The authors considered three adversaries in their adversary model: a malicious app adversary, a malicious network adversary and a restricted web adversary. The restricted web adversary was capable of setting up an unlimited number of malicious web sites but the mobile device user was only able to navigate websites that were allowed by the app. This model differed from typical web adversaries in that it was able to employ any number of websites and could assume that a user would visit these websites. A mobile user interacting with a webview does not necessarily have full browser functionality (e.g. there may not be a URL bar) and therefore the user must act in accordance with the app's functionality within the webview. The malicious app adversary was capable of writing to and reading from the device's (presumably) external storage and, hence, requires at the very least this dangerous permission.

Ren et al. [94] determined that many versions of the Android operating system were vulnerable to an attack known as "task hijacking". Such an attack takes place when a malicious app hijacks a user's app session with their own malicious user interface and misleads the user into providing personal details. They also used the malicious app adversary model, with the permission requirements including Internet access and "other commonly requested permissions". We classified this particular adversary model under the dangerous permission category. A similar study into hijacking was performed by Lee et al. [95]. They, instead, examined app installation hijacking via the use of an app which requested a single dangerous permission—WRITE_EXTERNAL_STORAGE. The authors demonstrated that it was possible to silently install malicious apps through the use of this single permission and analyzed over 1000 Android device images in their study.

Marforio et al. [96] proposed a system to detect illegitimate (i.e. counterfeit) apps on an Android device. They considered an adversary with the goal of obtaining legitimate user details by performing a phishing attack. The adversary was able to mislead the user into installing a malicious app and the adversary was also able to control and read all network traffic on the mobile device. As a direct result, the malicious app requested a number of dangerous permissions.

Similar to previously discussed research (e.g. [84, 85]), Xu et al. [97] examined the potential for leakage of sensitive data via malicious background apps on Android devices. They employed a malicious app adversary model able to read and transmit data on the device. Presumably, this indicates that the app would require the (dangerous) permission to read data on the device's storage. This specification differs from the models used by other researchers who examined vendor customization security (see [84, 85]). This model represented potentially malicious apps which could covertly obtain sensitive user

information during background operations. The authors further classified private data into four categories: basic phone data (e.g. call logs and contact information), application data (e.g. browser bookmarks), sensory data (e.g. GPS information) and hardware information (e.g. IMEI). All four of these categories required the use of at least one dangerous permission.

The use of device power consumption in detecting malware on devices has been a relatively common endeavor within the computer security literature. Due to the low power consumption of mobile devices and, therefore, the significance of smaller changes in power consumption, it can be argued that these techniques may be even more effective than on larger devices. Dixon and Mishra [98] examined the efficiency of two power consumption profiling-based malware detection techniques which were specific to smartphones. The authors employed two adversaries in their adversary model: a malicious app capable of generating spam SMS messages (i.e. a dangerous permission) and a root kit that was able to track an unsuspecting user. The authors justified their use of these two adversaries due to their ability to represent the bulk of malware in the wild. They later added the additional constraints of time and location to their power consumption-based malware detection technique [99]. Similar study, which also considered dangerous permission adversaries, was known as the POWERFUL framework [100].

Hong et al. [101] proposed a technique for regulation of third-party native libraries on Android devices. They considered a model that was very closely related to the malicious app adversary which, instead, modeled a malicious third-party native library. The authors note that most approaches in the literature that sought to determine if an app was malicious typically would analyze the app's java code and do not examine at the native library C/C++ compiled code.

We have noted that, as with the zero and normal permission adversary models discussed, dangerous permission adversary models used in the literature also lack specifics with regards to OS versions and, less often, devices that are used in the study. As previously discussed, such omissions in the study can critically affect future research and researchers as well as the study itself. Furthermore, there also exist significantly more studies involving dangerous permission adversary models than the two previously discussed models. This can be interpreted as dangerous permission apps, most likely, being some of the most downloaded and used apps (e.g. communications apps and social networking apps) on these platforms. Specifics regarding adversary assumptions, capabilities and goals have been summarized in Table 5.

Dangerous Permission Adversary Model	**Adversary Assumptions/Capabilities**	**Adversary Goals**
Luo et al. [87]	• An unspecified version of Android was utilized. • The malicious app is designed to work with a web application (e.g. Facebook or Twitter) and, as such, is able to request all the necessary permissions. • The adversary is assumed to have enticed the user to navigate to the web application within their app.	• Compromise the first party web application.
Wu et al. [88]	• Android 2.1 to 4.3 were, presumably, considered based on the devices utilized by the authors. • Malicious apps attempt to exploit device vulnerabilities and, presumably, are allowed access to all resource permissions.	• Gain unauthorized access to system resources in a manner unknown or unpermitted by the user.
Zhang et al. [89]	• The authors do not specify the Android OS used but we can conclude that Android 5.0 or above was considered based on the devices examined by the authors • Malicious apps were able to request any non-system permissions.	• Collect sensitive user data during app execution.
Zhou et al. [90]	• Android 4.1.2 was specified by the authors as the OS under consideration.	• No specific adversary goal was specified by the authors.

	• The malicious app may request any legitimate permission and does not have root access to the device.	
Yan et al. [91]	• CyanogenMod 11 was considered by the authors, which is based on Android 4.4. • The malicious app was able to gain access to all of the user land (e.g. it was able to request all legitimate permissions). • The authors specifically noted that side-channel and physical attacks were not considered as part of the adversary model.	• Collect sensitive user data.
Xing et al. [92]	• Mac OS X 10.10 was considered by the authors. • Apps could request any "inconspicuous" capability (e.g. Internet access).	• Collect sensitive user data. • Access resources belonging to other installed apps.
Mutchler et al. [93]	• No specific version of Android was specified by the authors for their study. • The malicious app was able to read and write to storage.	• Exploit device vulnerabilities.
Ren et al. [94]	• Android versions 3.x, 4.x and 5.0.x were specified by the authors as the OS versions considered in the study. • The malicious app adversary was able to request the Internet access along with a "minimum set of widely-requested permissions".	• Collect user credentials.
Lee et al. [95]	• Android versions between 4.0.3 and 5.1. • A single permission to write to external storage was requested by the adversary.	• The adversary sought to install malicious apps on the device without the user's knowledge.
Marforio et al. [96]	• The researchers specified the device used was a Samsung Galaxy S3 but did not state the Android OS it was running. This indicates that the OS version could be between Android 4.0.4 and 4.3, inclusively. • The malicious app was able to control the device's network traffic in both directions.	• Collect user banking login credentials.
Xu et al. [97]	• A customized version of Android 4.3 was considered by the authors. • The malicious app was able to read sensitive data stored on the device and transmit the data via a network.	• Collect sensitive user data.
Dixon and Mishra [98]	• The authors did not specify the Android OS version considered or the models of the devices utilized in their study. • The malicious app adversary was able to transmit SMS messages as well as obtain and transmit the user's locational information.	• Transmit SMS spam and collect user location information without alerting malware detectors.
Chen et al. [100]	• The authors considered two Android devices, which were running Android 4.4.4 and Android 5.1.1. The adversary has knowledge of the device's model, OS and power consumption history.	• Infer apps that the user is using.
Hong et al. [101]	• The authors performed their study on Android 4.4.2. • The malicious adversary did not consist of an app but rather a malicious native library that is executed as part of an installed app.	• Collect the user's location.

Table 5: A summary of the dangerous permission adversary models reviewed.

In addition to app-based adversary models, a wide range of other adversary models have been utilized by authors in the literature. In the following sections, we examine these models in greater detail in order to provide a contrasting view to app-based adversary models. This information also serves to demonstrate the wide range and capabilities of adversaries used in mobile security research.

4.2 Mobile Authentication

As smart devices become more and more powerful, capable of storing a wealth of sensitive information, the loss of these devices becomes all the more detrimental to a user or organization. In fact, monetary payments are increasingly performed on mobile devices, which further serves to demonstrate the importance of secure mobile authentication [102]. Current authentication schemes are often deemed to be insecure [103] or unsuitable for consumer use [104]. Authentication on smartphones can be

categorized into either textual passwords or alternative authentication methods. These alternatives can be further divided into the three categories of graphical-based (e.g. using the touchscreen to draw a shape), question-based (answering a customized query), passive (e.g. phone location) or biometric (e.g. fingerprints). Contemporary mobile authentication research typically focuses on the former two classifications as biometric and some forms of passive authentication (e.g. Bluetooth paired device unlocking) are present on many flagship devices. A notable aspect of adversary models used in mobile authentication is that adversaries tend to have physical and direct access to the mobile device.

4.2.1 Graphical-based Authentication Adversaries

Sun et al. [105] presented TouchIn, a secure graphical authentication system for multi-touch mobile devices that made use of two authentication paradigms—what-you-know and something-you-are. The user is authenticated by drawing on the touchscreen of the mobile device and the system verifies the user based on the geometry of the shape and other characteristics (e.g. touch pressure and hand geometry). TouchIn was designed to be resistant against a number of adversaries, categorized by the authors in their adversary model into four types: Types I to IV, from weakest to strongest. Type I adversaries had no knowledge of the password or how the owner of the device drew the password. Type II adversaries were able to observe how the owner draws and Type III adversaries also knew the rough shape of the password drawings. Finally, Type IV adversaries had complete knowledge of how the device owner draws and the exact drawing password.

4.2.2 Question-based Authentication Adversaries

Das et al. [106] proposed autobiographical question-based authentication for smartphones which made use of data present on the smartphone to authenticate the user. Types of authentication questions included phone usage, app usage, website visits, communications and locations. The authors considered five adversaries for their authentication scheme ranging from a naïve adversary capable only of guessing at random to an adversary that could correctly answer any question.

Another question-based authentication scheme was proposed by Hang et al. [107] which also made use of autobiographical data located on a smartphone to authenticate the user. Similar to the work by Das et al. [106], authentication questions involved topics such as calls made, text messages and app usage. The authors noted that the five adversary models proposed by Das et al. [106] were not able to fully capture all adversary circumstances which could influence adversarial performance. They also noted that a key difference with their adversary was that it was able to evaluate the security of the approach in a practical sense rather than the theoretical approach of Das et al. [106]. Hang et al. [107] assumed a physical attacker with access to the victim's smartphone in their adversary model. Other adversaries considered had varying levels of user knowledge.

4.2.3 Other Mobile Authentication Adversaries

A number of researchers in the field of mobile authentication have proposed passive or user biometric features as the primary form of device authentication. Examples of such authentication schemes include fingerprints, iris scanners and location detection. For example, Xu et al. [108] studied the feasibility of a passive and continuous mobile authentication system for smartphones. A malicious adversary with physical access to an unprotected (or compromised) smartphone was assumed and the adversary's goal was to violate the device owner's privacy by performing tasks such as reading emails or SMS messages.

Gasti et al. [109] presented an authentication scheme for mobile devices. The authors demonstrated the security of their proposal by testing against an adversary model. The model assumed a number of different scenarios, including: a malicious server, a malicious mobile device and collusion of these entities.Mobile authentication is a relatively popular field of research and, with this popularity, comes a number of different approaches, including the modeling of adversaries. Typically, mobile authentication adversaries have been modeled as attackers who have a primary goal of gaining illegitimate access to a secured device. This makes a mobile authentication adversary's goals clear and,

therefore, adversaries in the literature differ, largely, based on adversary capabilities (e.g. varying levels of access to dictionaries or relevant user data) and assumptions (e.g. a remote or physical attacker or different levels of prior knowledge).

4.3 Mobile Forensics

As human beings become more and more reliant on mobile devices, these devices can contain an ever-increasing amount of evidential data that may be of value in a forensic investigation. There have been several works in this field to date. Adversary models have been used to aid in the modeling of both forensic practitioners and the potential suspects.

Saltaformaggio et al. [110] presented a memory-based forensics technique for recovering photographic data from Android devices. With regards to the adversary model, the adversary was assumed to have removed all images on the device and had also removed and destroyed the device's external storage (i.e. SD cards).

In our previous work [111], we proposed an adversary model for use in digital forensics investigations, specifically targeting mobile devices. A number of distinct adversary capabilities were provided in the vein of Bellare-Rogaway adversary queries. Similarly, Azfar et al. [112] proposed an adversary model for use in Android social networking app forensics. The model also made use of a number of adversary queries.

Adversary models provide a means to validate contribute to the validation of forensic soundness for a proposed forensic process. This is achieved through the modeling of the forensic practitioner as the adversary, with the adversary capabilities being limited to only those capabilities that are considered forensically sound. It is, therefore, recommended that researchers working in the digital forensics field consider applying adversary models to proposed digital forensic processes or techniques.

4.4 Physical Adversary Models

Physical adversaries are a type of adversary which usually hold far greater power than logical adversaries. Adversary models utilizing these adversaries tend to be much rarer in the literature due to their physical requirement. In our previous research [70], we proposed one such adversary model for Android devices for use in covert data exfiltration that was able to systematically represent a malicious attacker. The adversary had physical access to the victim device and was able to perform a number of actions, which were modeled as adversary capabilities, including: ***Inject***, ***Modify*** and ***Transmit***. The covert channels considered included the use of frequencies outside the range of human hearing. A covert channel for mobile devices was also proposed by Zhang et al. [113] for use in devices communicating via voice over LTE (VoLTE). As opposed to audio frequencies, the authors used silence periods in between device communications in order to exfiltrate data.

Another physical adversary was considered by Wang et al. [114] who proposed an adversary model involving an attacker with passive eavesdropping capabilities who was located on the same wireless network as a victim. All traffic was encrypted and the adversary did not have the ability to decrypt this traffic (i.e. the adversary did not have knowledge of the private encryption keys). The authors were able to infer smartphone user activities based on this encrypted traffic. In a similar vein, Yang et al. [115] determined that it was possible to infer browsing history through analysis of smartphone USB power analysis. The authors did not require the data pins be available on the connection and considered a physical adversary that did not have direct access to the device under attack.

D'Orazio and Choo [116] proposed an adversary model to emulate a physical attacker targeting digital rights management (DRM) on iOS devices. The adversary was provided with a total of 13 capabilities, such as ***Modify*** and ***Extract***. The model was latter extended to detect vulnerabilities in iOS devices and apps, with adversary capabilities such as ***Re-sign***, ***Brute-force***, ***Hook***, ***Disassemble*** and ***Erase*** [117-119].

Jiang et al. [120] developed AppShell, a system for protecting sensitive information on Android devices that may be obtained by an attacker in case of theft. They considered a physical adversary model with the ability to read the memory of the obtained device (among other capabilities available to a physical attacker).

We have noted that with the greater presence of cyber-physical devices (e.g. smart cities), the use of physical adversaries in mobile research has started to increase in number over recent years. These adversary models will, ideally, prompt future researchers to consider more closely the physical security aspects of forthcoming devices.

4.5 Trends in Mobile Adversary Models

Unlike cryptography, mobile adversary models do not appear to have a common seminal origin. These models have instead been formed based on the characteristics that are present on mobile devices. We have observed that the vast majority of research into mobile malware makes use of the malicious app-based adversary model. This may be, primarily, due to the fact that one of the few feasible attack vectors on the heavily sandboxed modern mobile operating system is the third-party app. The two leading smartphone operating systems, iOS and Android (which make up about 99.6% of the market share as of late 2016 [121]), both make use of this app sandboxing architecture in order to bolster the security of their respective devices. By a significant margin, works employing adversary models in mobile security are Android-based. The Android OS may be so commonly researched due to its open-source nature compared to other popular smartphone operating systems such as Apple's iOS and Microsoft's Windows Phone. Android is also considered to be the most popular smartphone operating system, accounting for 76% of all smartphones sold globally in the first quarter of 2016 [122].

The malicious app adversary model is exceptionally popular in mobile research as first- and third-party apps are the primary method of accessing services and performing tasks on smartphones. Typically, apps are sandboxed by the operating system in order to improve user security and prevent apps from covertly obtaining sensitive information from other installed apps. As sandboxing of apps, which is employed by all three of the most popular smartphone operating systems (i.e. Android, iOS and Windows Phone), becomes more and more secure, usage of this particular adversary model may start to decline. As a result, in Section 4.1.1, we proposed an app-based adversary model that is used to classify app-based adversaries.

The weakest commonly used type of app-based adversary model in our adversary model classification is the zero permission app-based adversary model. Compared to the weakest adversary models used in other research fields, such as the passive or honest-but-curious adversary often used in cryptography and networking, it should be noted that the zero permission app-based adversary model is comparatively far more powerful due to the presence of its active capabilities. The next most powerful of the types of app-based adversary models are the adversary models in which the attacker requests "normal" permissions, permissions that are considered safer and less likely to lead to a leakage of user information or data. The final classification for app-based adversary models in our model is the dangerous permissions app-based adversary model. These adversaries are able to access user data that is deemed sensitive.

We were able to successfully classify all mobile app-based studies as relevant app-based adversary models. We have observed that a substantial number of researchers do not provide sufficient or detailed device environment information (e.g. OS versions) which, as we previously noted, could significantly affect the findings and future applicability of the study. For example, a particular study may describe the finding of a particular vulnerability, which may later be fixed by the OS developers. A future researcher may discover that a (newer) version of the OS may exhibit a similar vulnerability but due to the lack of version information in the prior publication, the future researchers may interpret the prior study as having already disseminated this particular vulnerability. These researchers may also,

therefore, conclude that the OS developers are already aware of the vulnerability and may not take any further steps in order to publicize or rectify the vulnerability.

Another commonly used adversary model is the physical attacker. As previously stated, such an attacker is less feasible when compared with the malicious app adversary model but also allows the modeling of attacks not suitable for the aforementioned adversary model. For example, researchers (see [76]) have used this adversary model to demonstrate cable-based (e.g. USB) and touch-screen authentication-based attacks, which would not be practical with an app-based adversary model.

5 Adversary Models in IoT

In this paper, IoT devices refer to all smart devices which are not classified as smartphones (as this research is described in the previous section) and also have Internet connectivity. This Internet connectivity may be indirect (e.g. some smartwatches may make use of a paired smartphone in order to communicate via the Internet) or direct. As previously noted, IoT is a relatively new and popular paradigm of ubiquitous everyday objects having the ability to collaborate and communicate with each other in order to reach a common objective [123]. With regards to research in IoT security, we consider research involving software security/forensics, hardware security/forensics, and software/hardware attacks.

A very similar field of study is that of cyber physical systems, which typically involve closer integration of physical components, a high level of automation and additional dynamic or adaptive capabilities [124]. Cardenas et al. [125] proposed an adversary model for use in securing cyber physical systems and provided a list of potential attackers. These attackers were described with potential resources and motivations, and included cybercriminals, disgruntled employees, (cyber) terrorists and nation states. These attackers encompass the vast majority of potential adversaries in computer security. A number of topics in the field of networking, such as wireless sensor networks, also have some overlap with IoT security research.

In the following section, we examine prominent IoT research that has employed the use of adversary models and discuss potential solutions to problems that may occur due to the increasing ubiquity of IoT. This examination serves to contrast and compare the differences, as well as similarities, between security research in cryptography, mobile security and IoT security with a view to the development of a common model or set of guidelines.

5.1 Current Research

One of the earliest envisioned forms of IoT was proposed in 1999 and involved the integration of a large numbers of RFID-enabled devices [126]. As a result, there have been a number of (earlier) works based on improving the security and privacy of RFID-based IoT architectures.

Zhu et al. [127] presented one such security (and privacy) model for the use of RFID in IoT systems. The authors outlined an in-depth adversary model, where the adversary had access to a number of specific capabilities, including the ability to corrupt RFID readers and tags. The adversary capabilities are, what we term, textually defined—the capabilities are described through light use of algebraic notation and heavily described using textual descriptions (also see Section 2). The capabilities themselves are specifically designed for use in an RFID environment and, therefore, would not be practical in other fields. For example, one adversary capability employed is ***CreateTag***, which allows an adversary to create a legitimate RFID tag and generate its initial state. Other RFID-based IoT adversary models were employed in [128-132].

In a similar study by Jin et al. [133], the authors examined the possible countermeasures for spectrum misuse (e.g. unlicensed used for frequencies). Similar to the ***CreateTag*** capability employed by the aforementioned RFID studies, the adversary in this model was able to freely choose frequencies to

transmit on and had full control of the transceiver. Such powerful adversaries appear to be very commonly used in IoT research (and related) fields.

The importance of IoT security is increasing due to the rapidly growing number of sensitive IoT use cases. For example, IoT devices are now used in power management, healthcare, supply management, public well-being/safety, home automation and battlefields (i.e. Internet of Battlefield Things, a term coined by the US Army Research Laboratory). An additional potential use case for IoT is smart vehicles, which if not adequately protected, could be a considerable danger to the user and their surroundings (similarly, car-sharing is another related area that would also require high levels of protection [134]). Another forthcoming technology is the concept of smart cities in which there is a high level of integration and collaboration within the city's infrastructure [135].

The Internet Threat Model was introduced in RFC 3552 [14] in 2003. The proposal was a type of adversary model where the (malicious) adversary was able to read, transmit and forge all network traffic. This model has been used by a number of authors in order to evaluate IoT systems (e.g. see [136, 137]).

The Dolev-Yao Model [1] has also been used to validate a number of IoT-based proposals and could be considered one of the more commonly used adversary models in this field (research that employs the Dolev-Yao Model includes [138, 139]). Nonetheless, due to the ubiquitous nature of IoT devices, physical attacks are a much more feasible attacker vector for malicious attackers when compared with typical mobile or network attacks. In fact, Køien [140] notes that the commonly used Dolev-Yao Model has a number of limitations that make it unsuitable for use in IoT security analysis. A major limitation is that IoT devices may be located in public-facing locations, thereby being vulnerable to physical modifications, and the Dolev-Yao adversary does not have this physical attacking capability.

In our previous research [141], we sought to alleviate this problem and examined the types of sensitive data that could be collected from contemporary smartwatches when under attack by a physical adversary. In order to formalize the approach, we presented an adversary model consisting of a physical adversary with a number of specific adversary capabilities. Also on the topic of smartwatches, Muslukhov et al. [142] designed a system which used a wearable device (e.g. a smartwatch) to perform device locking/unlocking. They assumed two adversaries in their adversary model, namely: a physical attacker and a network-based attacker. We believe that future research must take into account physical adversaries due to the prevalence of IoT and the potential for smart cities. In addition to IoT physical security, there have also been researchers focused on the very new field of IoT privacy. Alpár et al. [143] noted that, typically, cryptographic and security researchers define an adversary with goals and capabilities for use in security evaluations. They further specified that such a model's goals may be irrelevant in the case of many IoT privacy threats. For example, an adversary may collect sensitive information which they may then transfer to a different entity in order to process the data. As a result, an adversary's goals may be unclear or extraneous in such a situation. This leads to a need for specialized adversary models not only for privacy threats but also specifically for IoT privacy threats.

One such study was presented by Alcaide et al. [144] who proposed an anonymous and decentralized authentication protocol for use in IoT applications with an additional goal being the preservation of privacy. An honest-but-curious adversary model was utilized by the authors for their security analysis. Although, typically, honest-but-curious adversaries are passive, it should be noted that, unusually, in this research the adversary had a limited number of active capabilities. Later, in 2015, Lin et al. [145] demonstrated that the protocol proposed by Alcaide et al. [144] was vulnerable to an impersonation attack. Honest-but-curious adversary models have also been utilized in a number of other IoT privacy studies (e.g. see [146-148]).

Unlike mobile adversary models, where the majority of studies conducted can be classified under the app-based adversary model, IoT adversary models tended to be specific to the requirements of the study

undertaken. For example, a number of studies utilized a text-based description of an adversary based on the Dolev-Yao Model whilst others have specialized adversary models unique to the particular field (e.g. RFID). We posit that these models could be classified into a singular generalized scheme in order to allow for different approaches to be benchmarked and compared as well as to simplify the creation of the adversary model for different fields. This is discussed in greater detail in the following section.

5.2 Recommendations

Based on our learnings from the seminal cryptography work and the more recent (but still more mature) mobile research, we firstly recommend that IoT researchers adopt a more formal approach that would allow for validation, replication and formalization of an attacker, attack technique or secure system. This formal approach would be the adversary models explored throughout this paper and one such example was proposed in Section 4.1.1 for use in app-based mobile security studies. Due to the ubiquity of IoT devices, it is also important to model attacks from all types of attack vectors, such as remote (Internet and Intranet), physical and also passive adversaries.

Furthermore, adversary models are useful as they would also allow developers and manufacturers to predict where attackers will target and improve their systems based on this information. Developers could estimate attack vectors by taking into account adversary capabilities (e.g. each resource permission could be a potential security problem). Attackers could also be profiled before, during and after an attack. This would be especially useful in incident response situations where the responder seeks to derive a complete picture of the attack. It may also be applicable to real-world adversarial situations such as battlefields.

We now propose a generalized framework that would be able to classify the majority of attackers in an IoT environment. Based on the types of adversary models we have surveyed, we suggest that adversaries can be classified into the categories of Active (e.g. malicious) or Passive (e.g. honest-but-curious), and Active Adversaries could be further classified to be either Physical or Remote. Furthermore, Remote Adversaries could be classified as Internet-based or Intranet-based. This generalized adversary model framework is presented in Figure 4.

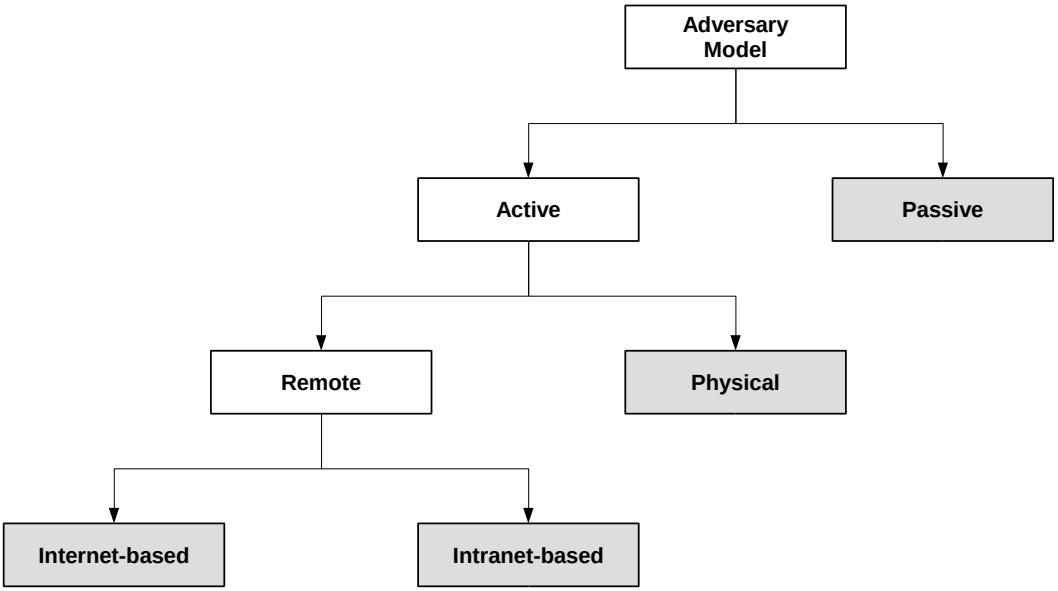

Figure 4: A classification scheme for adversary model types. The tree leaves (i.e. the grey boxes) represent possible adversary types.

Remote attackers are unable to access the physical hardware of the secured device whilst physical attackers may have varying levels of access and knowledge of the hardware. For example, a physical attacker may have no knowledge of a target device and instead can physically destroy the device (e.g. in a battlefield, where enemy forces can physically destroy an unmanned aerial vehicle—one of many devices in the Internet of Battlefield Things environment). Alternately a physical attacker may wish to insert malware into a device (e.g. install malware on a captured unmanned aerial vehicle for subsequent surveillance and data exfiltration) and therefore would not wish to destroy the target device. This attacker would also be obstructed by any software (and hardware) protections on the device such as storage encryption, unlike the physically destructive attacker. These modifications can allow for an adversary model to have varying levels of power and comparability.

Based on our survey, cryptographic adversary models tend to define and utilize fully realized adversaries. These adversaries would generally have adversary capabilities, formalized goals and assumptions. In the majority of cases, these models are used for security proofs of cryptographic protocols and, as such, are required to be rigorous and well-defined as they form an important part of these respective works. In this paper, we refer to an ideal adversary model as having these three key pillars of assumptions, goals and capabilities (see Section 2). A generalized guideline regarding these features is provided in the following sub-sections.

5.2.1 Adversary Assumptions

Adversary assumptions for IoT-based research will vary depending on the study undertaken but most models should, at the very least, endeavor to include the adversary environment such as the devices or software under scrutiny including full version details (e.g. OS and app versions as well as hardware revisions).

The classification scheme for adversary types rests on top of the three features and, as such, affect each feature. The classification directly relates to the adversary assumptions, which itself affects the adversary capabilities. Least affected is the adversary goals, which may even be unchanged. Therefore, the act of selecting an adversary type would, effectively, comprise a crucial part of the adversary assumptions.

5.2.2 Adversary Goals

Based on our survey, we found that most adversary models in the literature broadly sought to obtain sensitive user data. This ambiguous goal increases the difficulty for quantifiable validation of the adversary model (e.g. modeling of the attack steps or countermeasures). We recommend that all adversary models include a goal, and that the goal be as discrete and well-defined as possible. The goal should indicate the interests of the perceived adversary (i.e. not every adversary would be interested in all possible sensitive user data).

5.2.3 Adversary Capabilities

In addition to the overall adversary model classification, an IoT adversary model should also have a number of adversary capabilities. These capabilities are important as they allow formal modeling of an attacker's steps. We have observed that very few studies in the fields of mobile and IoT make use of adversary capabilities but, rather, specify adversaries in a general descriptive format.

The importance of having clearly defined and discrete adversary capabilities is apparent. It is an integral part of a formalized adversary in a quantifiable and measurable model. This allows third parties to evaluate the study and formally determine that no adversary capabilities have been breached in implementation.

5.3 Related Work

There have been a (small) number of surveys that have examined and classified the adversary models in the literature. In 2013, Roman et al. [149] examined challenges in IoT security and privacy. As part

of their review, the authors performed an analysis of the attacker models used in the literature and categorized these models into five distinct groups. These attacker categories were denial-of-service (e.g. jamming), physical damage (e.g. hardware destruction), passive (e.g. eavesdropping), physical active (e.g. hardware tampering) and control (e.g. corruption).

Also in 2013, Pék et al. [150] performed a survey of security issues that exist in hardware virtualization. They classified attacks into the categories of local and network attackers. These attackers were similar to our classification of remote and physical attackers, with the local attacker of Pék et al. [150] having access to the system's hardware. The authors also further categorized local attackers into three types, namely: Guest Grey Box (GGB), Infrastructure Grey Box (IGB) and White Box (WB). GGB attackers could perform attacks with privileged access to a single non-administrative device, IGB attackers had the ability to compromise multiple devices and WB attackers resembled malicious employees with physical access to the internal resources of every system in the organization. Pék et al. [150] also noted that WB adversaries are important to consider as it is a feasible problem in large computer systems where individuals have privileged access. The classifications proposed by Pék et al. [150] also coincide with our grouping of physical adversaries. In addition to these models, the authors considered a Black Box attacker who did not have network or physical access to a system—resembling an air gapped system. We do not consider such an attacker in our generalized framework.

Abomhara and Køien [151] performed a survey of IoT research up until early 2014 and noted that threats in IoT security could be classified into denial-of-service attacks, physical attacks, privacy attacks and intruder models. The authors note that a Dolev-Yao adversary model should be assumed (unless otherwise stated) but also add that this intruder should, in addition to the normal capabilities, have limited physical access to target devices. As IoT becomes more and more ubiquitous and the smart city and other applications (e.g. Internet of Battlefield) become an eventuality, such an assumption may no longer hold true due to the significantly larger attack vector for physical attackers.

6 Conclusion and Future Research

In this paper, we examined and critiqued the adversary models used in a number of different fields. Specifically, we chronicled adversary models used in cryptography and then closely examined and classified mobile adversary models. We also proposed a classification for app-based adversaries, used in Section 4 of this paper and suitable for use in future research. Finally, we contributed generalized guidelines for the design of future adversary models for use in applied security research. We summarized many of the key characteristics and weaknesses of adversary models specific to mobile and IoT research as well as possible future developments that may occur to these models due to changes in these fields.

More specifically, app-based studies, either demonstrating an attack or a countermeasure/secure system, often lack OS and device version details. For example, authors may use "the latest version of the OS" or provide a similar statement. This results in a study that is lacking a certain level of detail and applicability in future research. We hope that our proposed adversary model classification is able to encourage the inclusion of such details in future work, not only in mobile research, but also other such fields where this information is pertinent and required in order for replication studies or future work. Such a classification allows the cyber defender to make explicit assumptions on the adversary's capabilities and goals (e.g. the potential attack vectors and attacker's capability in manipulating the input data), which can then be used to inform risk mitigation strategy formulation.

Another potential research direction is to explore the potential of training machine learning algorithms to classify existing real-world attacks, such as the U.S. Office of Personnel Management and Equifax breaches based on publicly available information and/or information obtained via collaboration with relevant stakeholders or from court proceedings, in order to design a taxonomy of adversary capabilities.

Acknowledgements

The authors thank the editor and the three reviewers for their constructive feedback. The corresponding author also acknowledge the support of the Cloud Technology Endowed Professorship.

References

[1] D. Dolev, and A. C. Yao, “On the Security of Public Key Protocols,” *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198-208, 1983.

[2] R. McKemmish, "When is Digital Evidence Forensically Sound?," *Advances in Digital Forensics IV*, IFIP — The International Federation for Information Processing I. Ray and S. Shenoi, eds., pp. 3-15: Springer US, 2008.

[3] M. Bellare, H. Shi, and C. Zhang, "Foundations of Group Signatures: The Case of Dynamic Groups," *Topics in Cryptology – CT-RSA 2005*, Lecture Notes in Computer Science A. Menezes, ed., pp. 136-153: Springer Berlin Heidelberg, 2005.

[4] R. Canetti, and H. Krawczyk, "Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels," *Advances in Cryptology — EUROCRYPT 2001: 2001 Proceedings International Conference on the Theory and Application of Cryptographic Techniques Innsbruck, Austria, May 6–10,* , B. Pfitzmann, ed., pp. 453-474, Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.

[5] T. Y. Woo, and S. S. Lam, “Authentication for Distributed Systems,” *Computer*, vol. 25, no. 1, pp. 39-52, 1992.

[6] M. Bellare, and P. Rogaway, "Entity Authentication and Key Distribution," *Advances in Cryptology — CRYPTO’ 93*, Lecture Notes in Computer Science D. R. Stinson, ed., pp. 232-249: Springer Berlin Heidelberg, 1993.

[7] O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*, New York, NY, USA: Cambridge University Press, 2004.

[8] M. Bellare, and P. Rogaway, “Provably Secure Session Key Distribution: The Three Party Case,” in Proceedings of the 27th Annual ACM Symposium on Theory of Computing, Las Vegas, Nevada, USA, 1995, pp. 57-66.

[9] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated Key Exchange Secure against Dictionary Attacks," *Advances in Cryptology — EUROCRYPT 2000*, Lecture Notes in Computer Science B. Preneel, ed., pp. 139-155: Springer Berlin Heidelberg, 2000.

[10] U. Maurer, "Constructive Cryptography – A New Paradigm for Security Definitions and Proofs," *Theory of Security and Applications*, Lecture Notes in Computer Science S. Mödersheim and C. Palamidessi, eds., pp. 33-56: Springer Berlin Heidelberg, 2012.

[11] M. E. Whitman, and H. J. Mattord, *Principles of Information Security*, Boston, MA, United States: Course Technology Press, 2011.

[12] P. Liu, W. Zang, and M. Yu, “Incentive-based modeling and inference of attacker intent, objectives, and strategies,” *ACM Transactions on Information and System Security*, vol. 8, no. 1, pp. 78-118, 2005.

[13] S. Myagmar, A. J. Lee, and W. Yurcik, “Threat modeling as a basis for security requirements,” in Symposium on Requirements Engineering for Information Security, 2005, pp. 1-8.

[14] E. Rescorla, and B. Korver. "[RFC 3552] Guidelines for Writing RFC Text on Security Considerations," Accessed 31st October 2016; https://tools.ietf.org/html/rfc3552.

[15] G. S. Poh, J.-J. Chin, W.-C. Yau, K.-K. R. Choo, and M. S. Mohamad, “Searchable Symmetric Encryption: Designs and Challenges,” *ACM Computing Surveys*, pp. [In Press].

[16] R. L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[17] R. Oppliger, *Contemporary Cryptography*, Norwood, MA, USA: Artech House Publishers, 2011.

[18] J. Millen, and V. Shmatikov, “Constraint Solving for Bounded-Brocess Cryptographic Protocol Analysis,” in Proceedings of the 8th ACM Conference on Computer and Communications Security, 2001, pp. 166-175.

[19] M. Backes, "A Cryptographically Sound Dolev-Yao Style Security Proof of the Otway-Rees Protocol," *Computer Security – ESORICS 2004*, Lecture Notes in Computer Science P. Samarati, P. Ryan, D. Gollmann and R. Molva, eds., pp. 89-108: Springer Berlin Heidelberg, 2004.

[20] M. Backes, B. Pfitzmann, and M. Waidner, “The reactive simulatability (RSIM) framework for asynchronous systems,” *Information and Computation*, vol. 205, no. 12, pp. 1685-1720, 2007.

[21] R. A. Kemmerer, “Analyzing Encryption Protocols Using Formal Verification Techniques,” *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 4, pp. 448-457, 1989.

[22] C. Meadows, “Using Narrowing in the Analysis of Key Management Protocols,” in Proceedings of the 1989 IEEE Symposium on Security and Privacy, 1989, pp. 138-147.

[23] S. S. Al-Riyami, and K. G. Paterson, “Certificateless Public Key Cryptography,” in Proceedings of the 9th International Conference on the Theory and Application of Cryptology and Information Security, 2003, pp. 452-473.

[24] D. Cash, Y. Z. Ding, Y. Dodis, W. Lee, R. Lipton, and S. Walfish, "Intrusion-Resilient Key Exchange in the Bounded Retrieval Model," *Theory of Cryptography: 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007. Proceedings*, S. P. Vadhan, ed., pp. 479-498, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[25] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-Based Authenticated Key Exchange in the Three-Party Setting," *Public Key Cryptography - PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, January 23-26, 2005. Proceedings*, S. Vaudenay, ed., pp. 65-84, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[26] W. Wang, and L. Hu, "Efficient and Provably Secure Generic Construction of Three-Party Password-Based Authenticated Key Exchange Protocols," *Progress in Cryptology - INDOCRYPT 2006: 7th International Conference on Cryptology in India, Kolkata, India, December 11-13, 2006. Proceedings*, R. Barua and T. Lange, eds., pp. 118-132, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[27] M. Abdalla, and D. Pointcheval, "Interactive Diffie-Hellman Assumptions with Applications to Password-Based Authentication," *Financial Cryptography and Data Security: 9th International Conference, FC 2005, Roseau, The Commonwealth Of Dominica, February 28 – March 3, 2005. Revised Papers*, A. S. Patrick and M. Yung, eds., pp. 341-356, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[28] M. S. Farash, and M. A. Attari, “An efficient and provably secure three-party password-based authenticated key exchange protocol based on Chebyshev chaotic maps,” *Nonlinear Dynamics*, vol. 77, no. 1-2, pp. 399-411, 2014/07/01, 2014.

[29] J. Katz, and M. Yung, "Scalable Protocols for Authenticated Group Key Exchange," *Advances in Cryptology - CRYPTO 2003*, Lecture Notes in Computer Science D. Boneh, ed., pp. 110-125: Springer Berlin Heidelberg, 2003.

[30] E. Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater, “Provably authenticated group Diffie-Hellman key exchange,” in Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, PA, USA, 2001, pp. 255-264.

[31] E. Bresson, O. Chevassut, and D. Pointcheval, "Group Diffie-Hellman Key Exchange Secure against Dictionary Attacks," *Advances in Cryptology — ASIACRYPT 2002: 8th International Conference on the Theory and Application of Cryptology and Information Security Queenstown, New Zealand, December 1–5, 2002 Proceedings*, Y. Zheng, ed., pp. 497-514, Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[32] J. W. Byun, and D. H. Lee, "N-Party Encrypted Diffie-Hellman Key Exchange Using Different Passwords," *Applied Cryptography and Network Security: Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005. Proceedings*, J. Ioannidis, A. Keromytis and M. Yung, eds., pp. 75-90, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[33] E. Bresson, O. Chevassut, and D. Pointcheval, “A security solution for IEEE 802.11's ad hoc mode: password-authentication and group Diffie-Hellman key exchange,” *International Journal of Wireless and Mobile Computing*, vol. 2, no. 1, pp. 4-13, 2007.

[34] M. Abdalla, and D. Pointcheval, "A Scalable Password-Based Group Key Exchange Protocol in the Standard Model," *Advances in Cryptology – ASIACRYPT 2006: 12th International Conference on the Theory and Application of Cryptology and Information Security, Shanghai, China, December 3-7, 2006. Proceedings*, X. Lai and K. Chen, eds., pp. 332-347, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[35] L. Chen, Z. Cheng, and N. P. Smart, “Identity-based key agreement protocols from pairings,” *International Journal of Information Security*, vol. 6, no. 4, pp. 213-241, 2007.

[36] L. Zhang, F. Zhang, Q. Wu, and J. Domingo-Ferrer, “Simulatable certificateless two-party authenticated key agreement protocol,” *Information Sciences*, vol. 180, no. 6, pp. 1020-1030, 3/15/, 2010.

[37] C. Brzuska, M. Fischlin, B. Warinschi, and S. C. Williams, “Composability of Bellare-Rogaway Key Exchange Protocols,” in Proceedings of the 18th ACM Conference on Computer and Communications Security, Chicago, Illinois, USA, 2011, pp. 51-62.

[38] B. Dowling, M. Fischlin, F. Günther, and D. Stebila, “A Cryptographic Analysis of the TLS 1.3 Handshake Protocol Candidates,” in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, USA, 2015, pp. 1197-1210.

[39] T. Jager, F. Kohlar, S. Schäge, and J. Schwenk, "On the Security of TLS-DHE in the Standard Model," *Advances in Cryptology – CRYPTO 2012*, Lecture Notes in Computer Science R. Safavi-Naini and R. Canetti, eds., pp. 273-293: Springer Berlin Heidelberg, 2012.

[40] W. Diffie, and M. E. Hellman, “New Directions in Cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.

[41] H. Krawczyk, K. Paterson, and H. Wee, "On the Security of the TLS Protocol: A Systematic Analysis," *Advances in Cryptology – CRYPTO 2013*, Lecture Notes in Computer Science R. Canetti and J. Garay, eds., pp. 429-448: Springer Berlin Heidelberg, 2013.

[42] M. Bellare, R. Canetti, and H. Krawczyk, “A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols (Extended Abstract),” in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, Dallas, Texas, USA, 1998, pp. 419-428.

[43] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger Security of Authenticated Key Exchange," *Provable Security: First International Conference, ProvSec 2007, Wollongong, Australia, November 1-2, 2007. Proceedings*, W. Susilo, J. K. Liu and Y. Mu, eds., pp. 1-16, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[44] K. Yoneyama, "Efficient and Strongly Secure Password-Based Server Aided Key Exchange (Extended Abstract)," *Progress in Cryptology - INDOCRYPT 2008: 9th International Conference on Cryptology in India, Kharagpur, India, December 14-17, 2008. Proceedings*, D. R. Chowdhury, V. Rijmen and A. Das, eds., pp. 172-184, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[45] G. Lippold, C. Boyd, and J. M. G. Nieto, "Strongly Secure Certificateless Key Agreement," *Pairing-Based Cryptography – Pairing 2009: Third International Conference Palo Alto, CA, USA, August 12-14, 2009 Proceedings*, H. Shacham and B. Waters, eds., pp. 206-230, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[46] C. M. Swanson, “Security in Key Agreement: Two-Party Certificateless Schemes,” Master Thesis, University of Waterloo, 2009.

[47] C. Swanson, and D. Jao, "A Study of Two-Party Certificateless Authenticated Key-Agreement Protocols," *Progress in Cryptology - INDOCRYPT 2009: 10th International Conference on Cryptology in India, New Delhi, India, December 13-16, 2009. Proceedings*, B. Roy and N. Sendrier, eds., pp. 57-71, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[48] F. Wang, and Y. Zhang, “A new provably secure authentication and key agreement mechanism for SIP using certificateless public-key cryptography,” *Computer Communications*, vol. 31, no. 10, pp. 2142-2149, 6/25/, 2008.

[49] D. H. Yum, and P. J. Lee, "Generic Construction of Certificateless Signature," *Information Security and Privacy: 9th Australasian Conference, ACISP 2004, Sydney, Australia, July 13-*

15, 2004. Proceedings, H. Wang, J. Pieprzyk and V. Varadharajan, eds., pp. 200-211, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[50] B. C. Hu, D. S. Wong, Z. Zhang, and X. Deng, "Key Replacement Attack Against a Generic Construction of Certificateless Signature," *Information Security and Privacy: 11th Australasian Conference, ACISP 2006, Melbourne, Australia, July 3-5, 2006. Proceedings*, L. M. Batten and R. Safavi-Naini, eds., pp. 235-246, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[51] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the Security of Certificateless Signature Schemes from Asiacrypt 2003," *Cryptology and Network Security: 4th International Conference, CANS 2005, Xiamen, China, December 14-16, 2005. Proceedings*, Y. G. Desmedt, H. Wang, Y. Mu and Y. Li, eds., pp. 13-25, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[52] Z. Zhang, D. S. Wong, J. Xu, and D. Feng, "Certificateless Public-Key Signature: Security Model and Efficient Construction," *Applied Cryptography and Network Security: 4th International Conference, ACNS 2006, Singapore, June 6-9, 2006. Proceedings*, J. Zhou, M. Yung and F. Bao, eds., pp. 293-308, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[53] K. Bentahar, P. Farshim, J. Malone-Lee, and N. P. Smart, "Generic Constructions of Identity-Based and Certificateless KEMs," *Journal of Cryptology*, vol. 21, no. 2, pp. 178-199, 2007.

[54] J. K. Liu, M. H. Au, and W. Susilo, "Self-Generated-Certificate Public Key Cryptography and certificateless signature/encryption scheme in the standard model: extended abstract," in Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, Singapore, 2007, pp. 273-283.

[55] A. W. Dent, "A survey of certificateless encryption schemes and security models," *International Journal of Information Security*, vol. 7, no. 5, pp. 349-377, 2008.

[56] N. J. Hopper, and M. Blum, "Secure Human Identification Protocols," *Advances in Cryptology — ASIACRYPT 2001*, Lecture Notes in Computer Science C. Boyd, ed., pp. 52-66: Springer Berlin Heidelberg, 2001.

[57] X. Huang, Y. Mu, W. Susilo, F. Zhang, and X. Chen, "A Short Proxy Signature Scheme: Efficient Authentication in the Ubiquitous World," *Embedded and Ubiquitous Computing – EUC 2005 Workshops*, Lecture Notes in Computer Science T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai and L. Yang, eds., pp. 480-489: Springer Berlin Heidelberg, 2005.

[58] W. Wu, Y. Mu, W. Susilo, J. Seberry, and X. Huang, "Identity-Based Proxy Signature from Pairings," *Autonomic and Trusted Computing*, Lecture Notes in Computer Science B. Xiao, L. Yang, J. Ma, C. Muller-Schloer and Y. Hua, eds., pp. 22-31: Springer Berlin Heidelberg, 2007.

[59] M. Barbosa, and P. Farshim, "Certificateless Signcryption," in Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, Tokyo, Japan, 2008, pp. 369-372.

[60] Y. Aumann, and Y. Lindell, "Security Against Covert Adversaries: Efficient Protocols for Realistic Adversaries," *Theory of Cryptography*, Lecture Notes in Computer Science S. Vadhan, ed., pp. 137-156: Springer Berlin Heidelberg, 2007.

[61] M. Geng, and F. Zhang, "Provably Secure Certificateless Two-Party Authenticated Key Agreement Protocol without Pairing," in Proceedings of the 2009 International Conference on Computational Intelligence and Security, 2009, pp. 208-212.

[62] Y. Lindell, and B. Pinkas, "Secure Multiparty Computation for Privacy-Preserving Data Mining," *Journal of Privacy and Confidentiality*, vol. 1, no. 1, pp. 59-98, 2009.

[63] J. Zhao, D. Gu, and M. C. Gorantla, "Stronger Security Model of Group Key Agreement," in Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, Hong Kong, China, 2011, pp. 435-440.

[64] M. C. Gorantla, C. Boyd, and J. M. González Nieto, "Modeling Key Compromise Impersonation Attacks on Group Key Exchange Protocols," *Public Key Cryptography – PKC 2009: 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings*, S. Jarecki and G. Tsudik, eds., pp. 105-123, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[65] R. Ostrovsky, and M. Yung, “How to withstand mobile virus attacks,” in Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing, Montreal, Quebec, Canada, 1991, pp. 51-59.
[66] L. Davi, A. Dmitrienko, A.-R. Sadeghi, and M. Winandy, "Privilege Escalation Attacks on Android," *Information Security*, Lecture Notes in Computer Science M. Burmester, G. Tsudik, S. Magliveras and I. Ilić, eds., pp. 346-360: Springer Berlin Heidelberg, 2010.
[67] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, A.-R. Sadeghi, and B. Shastry, “Towards Taming Privilege-Escalation Attacks on Android,” in Proceedings of the 19th Annual Symposium on Network and Distributed System Security, 2012, pp. 1-18.
[68] X. Luyi, P. Xiaorui, W. Rui, Y. Kan, and W. Xiaofeng, “Upgrading Your Android, Elevating My Malware: Privilege Escalation through Mobile OS Updating,” in Proceedings of the 2014 IEEE Symposium on Security and Privacy, 2014, pp. 393-408.
[69] Google. "Requesting Permissions," Accessed 21st March 2017; https://developer.android.com/guide/topics/permissions/requesting.html#normal-dangerous.
[70] Q. Do, B. Martini, and K.-K. R. Choo, “Exfiltrating Data from Android Devices,” *Computers & Security*, vol. 48, pp. 74-91, 2015.
[71] Google. "Manifest.permission," Accessed 21st March 2017; https://developer.android.com/reference/android/Manifest.permission.html#READ_EXTERNAL_STORAGE.
[72] Google. "Android 6.0 Changes," Accessed 26th April 2017; https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html.
[73] X. Zhou, S. Demetriou, D. He, M. Naveed, X. Pan, X. Wang, C. A. Gunter, and K. Nahrstedt, “Identity, Location, Disease and More: Inferring Your Secrets from Android Public Resources,” in Proceedings of the 20th Conference on Computer and Communications Security, 2013, pp. 1017-1028.
[74] W. Diao, X. Liu, Z. Zhou, and K. Zhang, “Your Voice Assistant is Mine: How to Abuse Speakers to Steal Information and Control Your Phone,” in Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices, Scottsdale, Arizona, USA, 2014, pp. 63-74.
[75] M. Lipp, D. Gruss, R. Spreitzer, C. Maurice, and S. Mangard, “ARMageddon: Cache Attacks on Mobile Devices,” in Proceedings of the 25th USENIX Security Symposium, 2016, pp. 549-564.
[76] W. Meng, W. H. Lee, S. R. Murali, and S. P. T. Krishnan, “Charging Me and I Know Your Secrets!: Towards Juice Filming Attacks on Smartphones,” in Proceedings of the 1st ACM Workshop on Cyber-Physical System Security, Singapore, Republic of Singapore, 2015, pp. 89-98.
[77] R. Spreitzer, F. Kirchengast, D. Gruss, and S. Mangard, “ProcHarvester: Fully Automated Analysis of Procfs Side-Channel Leaks on Android,” in Proceedings of the 2018 on Asia Conference on Computer and Communications Security, Incheon, Republic of Korea, 2018, pp. 749-763.
[78] Y. Zhou, and X. Jiang, “Detecting Passive Content Leaks and Pollution in Android Applications,” in Proceedings of the 20th Network and Distributed System Security Symposium, 2013, pp. 1-16.
[79] D. Wu, and R. K. C. Chang, “Indirect File Leaks in Mobile Applications,” in IEEE Mobile Security Technologies, 2015, pp. 1-10.
[80] W. Diao, X. Liu, Z. Zhou, K. Zhang, and Z. Li, "Mind-Reading: Privacy Attacks Exploiting Cross-App KeyEvent Injections," *Computer Security -- ESORICS 2015: 20th European Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceedings, Part II*, G. Pernul, P. Y A Ryan and E. Weippl, eds., pp. 20-39, Cham: Springer International Publishing, 2015.
[81] Y. Cheng, Y. Li, and R. H. Deng, "A Feasible No-Root Approach on Android," *Information Security and Privacy: 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part II*, J. K. Liu and R. Steinfeld, eds., pp. 481-489, Cham: Springer International Publishing, 2016.

[82] C.-C. Lin, H. Li, X. Zhou, and X. Wang, “Screenmilker: How to Milk Your Android Screen for Secrets,” in Proceedings of the 21st Network and Distributed System Symposium, 2014, pp. 1-14.
[83] M. Naveed, X. Zhou, S. Demetriou, X. Wang, and C. A. Gunter, “Inside Job: Understanding and Mitigating the Threat of External Device Mis-Bonding on Android,” in Proceedings of the 21st Network and Distributed System Symposium, 2014, pp. 1-14.
[84] Y. Aafer, N. Zhang, Z. Zhang, X. Zhang, K. Chen, X. Wang, X. Zhou, W. Du, and M. Grace, “Hare Hunting in the Wild Android: A Study on the Threat of Hanging Attribute References,” in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, USA, 2015, pp. 1248-1259.
[85] L. Wu, M. Grace, Y. Zhou, C. Wu, and X. Jiang, “The Impact of Vendor Customizations on Android Security,” in Proceedings of the 20th Conference on Computer and Communications Security, 2013, pp. 623-634.
[86] X. Zhou, Y. Lee, N. Zhang, M. Naveed, and X. Wang, “The Peril of Fragmentation: Security Hazards in Android Device Driver Customizations,” in Proceedings of the 2014 IEEE Symposium on Security and Privacy, 2014, pp. 409-423.
[87] T. Luo, H. Hao, W. Du, Y. Wang, and H. Yin, “Attacks on WebView in the Android System,” in Proceedings of the 27th Annual Computer Security Applications Conference, Orlando, Florida, USA, 2011, pp. 343-352.
[88] C. Wu, Y. Zhou, K. Patel, Z. Liang, and X. Jiang, “AirBag: Boosting Smartphone Resistance to Malware Infection,” in Proceedings of the 21st Network and Distributed System Symposium, 2014, pp. 1-13.
[89] N. Zhang, K. Yuan, M. Naveed, X. Zhou, and X. Wang, “Leave Me Alone: App-Level Protection against Runtime Information Gathering on Android,” in Proceedings of the 2015 IEEE Symposium on Security and Privacy, 2015, pp. 915-930.
[90] Y. Zhou, K. Patel, L. Wu, Z. Wang, and X. Jiang, “Hybrid User-level Sandboxing of Third-party Android Apps,” in Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, Singapore, Republic of Singapore, 2015, pp. 19-30.
[91] L. Yan, Y. Guo, and X. Chen, "SplitDroid: Isolated Execution of Sensitive Components for Mobile Applications," *Security and Privacy in Communication Networks: 11th International Conference, SecureComm 2015, Dallas, TX, USA, October 26-29, 2015, Revised Selected Papers*, B. Thuraisingham, X. Wang and V. Yegneswaran, eds., pp. 78-96, Cham: Springer International Publishing, 2015.
[92] L. Xing, X. Bai, T. Li, X. Wang, K. Chen, X. Liao, S.-M. Hu, and X. Han, “Cracking App Isolation on Apple: Unauthorized Cross-App Resource Access on MAC OS X and iOS,” in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, USA, 2015, pp. 31-43.
[93] P. Mutchler, A. Doupé, J. Mitchell, C. Kruegel, and G. Vigna, “A Large-Scale Study of Mobile Web App Security,” in Proceedings of the 2015 Mobile Security Technologies Workshop, 2015, pp. 1-11.
[94] C. Ren, Y. Zhang, H. Xue, T. Wei, and P. Liu, “Towards Discovering and Understanding Task Hijacking in Android,” in Proceedings of the 24th USENIX Security Symposium, 2015, pp. 945-959.
[95] Y. Lee, T. Li, N. Zhang, S. Demetriou, M. Zha, X. Wang, K. Chen, X. Zhou, X. Han, and M. Grace, “Ghost Installer in the Shadow: Security Analysis of App Installation on Android,” in Proccedings of the 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2017, pp. 403-414.
[96] C. Marforio, R. J. Masti, C. Soriente, K. Kostiainen, and S. Capkun, “Hardened Setup of Personalized Security Indicators to Counter Phishing Attacks in Mobile Banking,” in Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices, Vienna, Austria, 2016, pp. 83-92.
[97] H. Xu, Y. Zhou, C. Gao, Y. Kang, and M. R. Lyu, “SpyAware: Investigating the Privacy Leakage Signatures in App Execution Traces,” in Proceedings of the 26th International Symposium on Software Reliability Engineering, 2015, pp. 348-358.

[98] B. Dixon, and S. Mishra, “Power Based Malicious Code Detection Techniques for Smartphones,” in Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2013, pp. 142-149.
[99] B. Dixon, S. Mishra, and J. Pepin, “Time and Location Power Based Malicious Code Detection Techniques for Smartphones,” in Proceedings of the 13th International Symposium on Network Computing and Applications, 2014, pp. 261-268.
[100] Y. Chen, X. Jin, J. Sun, R. Zhang, and Y. Zhang, “POWERFUL: Mobile App Fingerprinting via Power Analysis,” in Proceedings of the 2017 IEEE Conference on Computer Communications, 2017, pp. 1-9.
[101] Y.-Y. Hong, Y.-P. Wang, and J. Yin, "NativeProtector: Protecting Android Applications by Isolating and Intercepting Third-Party Native Libraries," *ICT Systems Security and Privacy Protection: 31st IFIP TC 11 International Conference, SEC 2016, Ghent, Belgium, May 30 - June 1, 2016, Proceedings*, J.-H. Hoepman and S. Katzenbeisser, eds., pp. 337-351, Cham: Springer International Publishing, 2016.
[102] W. Yang, Y. Zhang, J. Li, H. Liu, Q. Wang, Y. Zhang, and D. Gu, “Show Me the Money! Finding Flawed Implementations of Third-party In-app Payment in Android Apps,” in Proceedings of the 2017 Network and Distributed System Security Symposium, 2017, pp. 1-15.
[103] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, “Smudge Attacks on Smartphone Touch Screens,” in Proceedings of the 4th USENIX conference on Offensive Technologies 2010, pp. 1-7.
[104] T. Feng, Z. Liu, K. A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen, “Continuous Mobile Authentication using Touchscreen Gestures,” in Proceedings of the 2012 IEEE Conference on Technologies forHomeland Security, 2012, pp. 451-456.
[105] J. Sun, R. Zhang, J. Zhang, and Y. Zhang, “TouchIn: Sightless Two-factor Authentication on Multi-touch Mobile Devices,” in Proceedings of the 2014 IEEE Conference on Communications and Network Security, 2014, pp. 436-444.
[106] S. Das, E. Hayashi, and J. I. Hong, “Exploring Capturable Everyday Memory for Autobiographical Authentication,” in Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Zurich, Switzerland, 2013, pp. 211-220.
[107] A. Hang, A. D. Luca, and H. Hussmann, “I Know What You Did Last Week! Do You?: Dynamic Security Questions for Fallback Authentication on Smartphones,” in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, Seoul, Republic of Korea, 2015, pp. 1383-1392.
[108] H. Xu, Y. Zhou, and M. R. Lyu, “Towards Continuous and Passive Authentication via Touch Biometrics: An Experimental Study on Smartphones,” in Proceedings of the 2014 Symposium On Usable Privacy and Security, 2014, pp. 187-198.
[109] P. Gasti, J. Šeděnka, Q. Yang, G. Zhou, and K. S. Balagani, “Secure, Fast, and Energy-Efficient Outsourced Authentication for Smartphones,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2556-2571, 2016.
[110] B. Saltaformaggio, R. Bhatia, Z. Gu, X. Zhang, and D. Xu, “VCR: App-Agnostic Recovery of Photographic Evidence from Android Device Memory Images,” in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, USA, 2015, pp. 146-157.
[111] Q. Do, B. Martini, and K.-K. R. Choo, “A Forensically Sound Adversary Model for Mobile Devices,” *PLoS ONE*, vol. 10, no. 9, pp. 1-15, 2015.
[112] A. Azfar, K. K. R. Choo, and L. Liu, “An Android Social App Forensics Adversary Model,” in Proceedings of the 49th Hawaii International Conference on System Sciences, 2016, pp. 5597-5606.
[113] X. Zhang, Y. Tan, C. Liang, Y. Li, and J. Li, “A Covert Channel Over VoLTE via Adjusting Silence Periods,” *IEEE Access*, vol. 6, pp. 9292-9302, 2018.
[114] Q. Wang, A. Yahyavi, B. Kemme, and W. He, “I Know What You Did On Your Smartphone: Inferring App Usage Over Encrypted Data Traffic,” in Proceedings of the 2015 IEEE Conference on Communications and Network Security, 2015, pp. 433-441.

[115] Q. Yang, P. Gasti, G. Zhou, A. Farajidavar, and K. S. Balagani, "On Inferring Browsing Activity on Smartphones via USB Power Analysis Side-Channel," *IEEE Transactions on Information Forensics and Security,* vol. 12, no. 5, pp. 1056-1066, 2017.
[116] C. D'Orazio, and K.-K. R. Choo, "An adversary model to evaluate DRM protection of video contents on iOS devices," *Computers & Security,* vol. 56, pp. 94-110, 2016.
[117] C. J. D'Orazio, R. Lu, K.-K. R. Choo, and A. V. Vasilakos, "A Markov adversary model to detect vulnerable iOS devices and vulnerabilities in iOS apps," *Applied Mathematics and Computation,* vol. 293, pp. 523-544, 2017.
[118] C. J. D'Orazio, and K.-K. R. Choo, "Circumventing iOS Security Mechanisms for APT Forensic Investigations: A Security Taxonomy for Cloud Apps," *Future Generation Computer Systems,* pp. [In Press].
[119] C. J. D'Orazio, and K.-K. R. Choo, "A Technique to Circumvent SSL/TLS Validations on iOS Devices," *Future Generation Computer Systems,* pp. [In Press].
[120] X. Jiang, K. Singh, and Y. Zhou, "AppShell: Making Data Protection Practical for Lost or Stolen Android Devices," in Proceedings of the 2016 IEEE/IFIP Network Operations and Management Symposium, 2016, pp. 502-508.
[121] Gartner. "Gartner Says Worldwide Sales of Smartphones Grew 7 Percent in the Fourth Quarter of 2016," Accessed 16th April 2017; http://www.gartner.com/newsroom/id/3609817.
[122] L. Guenveur. "Android Share Growth Slows After Historic Gains Last Period," Accessed 21st June 2016; http://www.kantarworldpanel.com/global/News/Android-Share-Growth-Slows-After-Historic-Gains-Last-Period-.
[123] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks,* vol. 54, no. 15, pp. 2787-2805, 2010.
[124] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of Cyber-Physical Systems," in Proceedings of the 2011 International Conference on Wireless Communications and Signal Processing, 2011, pp. 1-6.
[125] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, "Challenges for Securing Cyber Physical Systems," in Workshop on Future Firections in Cyber-Physical Systems Security, 2009, pp. 1-7.
[126] F. Mattern, and C. Floerkemeier, "From the Internet of Computers to the Internet of Things," *From Active Data Management to Event-Based Systems and More*, K. Sachs, I. Petrov and P. Guerrero, eds., pp. 242-259, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
[127] W. Zhu, J. Yu, and T. Wang, "A Security and Privacy Model for Mobile RFID Systems in the Internet of Things," in Proceedings of the 14th International Conference on Communication Technology, 2012, pp. 726-732.
[128] M. L. Das, "Strong Security and Privacy of RFID System for Internet of Things Infrastructure," *Security, Privacy, and Applied Cryptography Engineering*, B. Gierlichs, S. Guilley and D. Mukhopadhyay, eds., pp. 56-69, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
[129] M. L. Das, "Privacy and Security Challenges in Internet of Things," *Distributed Computing and Internet Technology: 11th International Conference, ICDCIT 2015, Bhubaneswar, India, February 5-8, 2015. Proceedings*, R. Natarajan, G. Barua and M. R. Patra, eds., pp. 33-48, Cham: Springer International Publishing, 2015.
[130] M. Asadpour, and M. T. Dashti, "Scalable, privacy preserving radio-frequency identification protocol for the internet of things," *Concurrency and Computation: Practice and Experience,* vol. 27, no. 8, pp. 1932-1950, 2015.
[131] E. Rekleitis, P. Rizomiliotis, and S. Gritzalis, "How to protect security and privacy in the IoT: a policy-based RFID tag management protocol," *Security and Communication Networks,* vol. 7, no. 12, pp. 2669-2683, 2014.
[132] B. R. Ray, M. U. Chowdhury, and J. H. Abawajy, "Secure Object Tracking Protocol for the Internet of Things," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 544-553, 2016.
[133] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "SpecGuard: Spectrum Misuse Detection in Dynamic Spectrum Access Systems," *IEEE Transactions on Mobile Computing*, pp. 1-9, 2018.
[134] A. Dmitrienko, and C. Plappert, "Secure Free-Floating Car Sharing for Offline Cars," in Proceedings of the 7th ACM on Conference on Data and Application Security and Privacy, Scottsdale, Arizona, USA, 2017, pp. 349-360.

[135] T. Nam, and T. A. Pardo, “Conceptualizing Smart City with Dimensions of Technology, People, and Institutions,” in Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times, College Park, Maryland, USA, 2011, pp. 282-291.
[136] F. V. Meca, J. H. Ziegeldorf, P. M. Sanchez, O. G. Morchon, S. S. Kumar, and S. L. Keoh, “HIP Security Architecture for the IP-Based Internet of Things,” in Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops, 2013, pp. 1331-1336.
[137] O. Garcia-Morchon, S. L. Keoh, S. Kumar, P. Moreno-Sanchez, F. Vidal-Meca, and J. H. Ziegeldorf, “Securing the IP-based internet of things with HIP and DTLS,” in Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Budapest, Hungary, 2013, pp. 119-124.
[138] Y. Hou, M. Li, and J. D. Guttman, “Chorus: Scalable In-band Trust Establishment for Multiple Constrained Devices over the Insecure Wireless Channel,” in Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Budapest, Hungary, 2013, pp. 167-178.
[139] T. Kovačević, T. Perković, and M. Čagalj, “Flashing displays: user-friendly solution for bootstrapping secure associations between multiple constrained wireless devices,” *Security and Communication Networks,* vol. 9, no. 10, pp. 1050-1071, 2016.
[140] G. M. Køien, "Privacy Enhanced Device Access," *Security and Privacy in Mobile Information and Communication Systems: Third International ICST Conference, MobiSec 2011, Aalborg, Denmark, May 17-19, 2011, Revised Selected Papers*, R. Prasad, K. Farkas, A. U. Schmidt, A. Lioy, G. Russello and F. L. Luccio, eds., pp. 76-87, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
[141] Q. Do, B. Martini, and K.-K. R. Choo, “Is the data on your wearable device secure? An Android Wear smartwatch case study,” *Software: Practice and Experience,* vol. 47, no. 3, pp. 391-403, 2017.
[142] I. Muslukhov, S.-T. Sun, P. Wijesekera, Y. Boshmaf, and K. Beznosov, “Decoupling data-at-rest encryption and smartphone locking with wearable devices,” *Pervasive and Mobile Computing*, pp. 1-9, 2016.
[143] G. Alpár, L. Batina, L. Batten, V. Moonsamy, A. Krasnova, A. Guellier, and I. Natgunanathan, “New directions in IoT privacy using attribute-based authentication,” in Proceedings of the 2016 ACM International Conference on Computing Frontiers, Como, Italy, 2016, pp. 461-466.
[144] A. Alcaide, E. Palomar, J. Montero-Castillo, and A. Ribagorda, “Anonymous authentication for privacy-preserving IoT target-driven applications,” *Computers & Security,* vol. 37, pp. 111-123, 9//, 2013.
[145] X.-J. Lin, L. Sun, and H. Qu, “Insecurity of an anonymous authentication for privacy-preserving IoT target-driven applications,” *Computers & Security,* vol. 48, pp. 142-149, 2//, 2015.
[146] D. Banerjee, B. Dong, M. Taghizadeh, and S. Biswas, “Privacy-Preserving Channel Access for Internet of Things,” *IEEE Internet of Things Journal,* vol. 1, no. 5, pp. 430-445, 2014.
[147] A. Ukil, S. Bandyopadhyay, and A. Pal, “Privacy for IoT: Involuntary Privacy Enablement for Smart Energy Systems,” in Proceedings of the 2015 IEEE International Conference on Communications, 2015, pp. 536-541.
[148] G. Ács, and C. Castelluccia, "I Have a DREAM! (DiffeRentially privatE smArt Metering)," *Information Hiding*, Lecture Notes in Computer Science T. Filler, T. Pevný, S. Craver and A. Ker, eds., pp. 118-132: Springer Berlin Heidelberg, 2011.
[149] R. Roman, J. Zhou, and J. Lopez, “On the features and challenges of security and privacy in distributed internet of things,” *Computer Networks,* vol. 57, no. 10, pp. 2266-2279, 7/5/, 2013.
[150] G. Pék, L. Buttyán, and B. Bencsáth, “A Survey of Security Issues in Hardware Virtualization,” *ACM Computing Surveys,* vol. 45, no. 3, pp. 1-34, 2013.
[151] M. Abomhara, and G. M. Køien, “Security and Privacy in the Internet of Things: Current Status and Open Issues,” in Proceedings of the 2014 International Conference on Privacy and Security in Mobile Systems, 2014, pp. 1-8.