Security and Privacy for Emerging Augmented Reality Technologies

Kiron Lebeck

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Franziska Roesner, Chair

Tadayoshi Kohno, Chair

Brian Curless

Program Authorized to Offer Degree:
Paul G. Allen School of Computer Science & Engineering

University of Washington

Abstract

Security and Privacy for Emerging
Augmented Reality Technologies

Kiron Lebeck

Co-Chairs of the Supervisory Committee:
Professor Franziska Roesner
Paul G. Allen School of Computer Science & Engineering

Professor Tadayoshi Kohno
Paul G. Allen School of Computer Science & Engineering

Augmented reality (AR) has emerged as a powerful computing paradigm in recent years. These technologies enable users to interact with digital content in new and exciting ways by continuously capturing sensory input from a user's surroundings and overlaying digital feedback atop the user's perception of the physical world. With application domains ranging from entertainment and education to automotive assistance and countless others, AR has the potential to fundamentally change how we engage with technology as part of our daily lives. Unfortunately, AR technologies may also expose users to new security and privacy risks that stem from the unique capabilities that make these technologies so powerful, and we currently lack a deep understanding of these risks or how to defend against them.

This dissertation identifies and addresses several key gaps in the AR security and privacy landscape, which represent critical impediments to realizing the full potential of these emerging technologies. First, it identifies the risks of visual output generated by immersive AR applications that may be malicious or buggy, and it describes the design of Arya — an AR platform that my collaborators and I created to constrain the output capabilities of AR applications while still supporting flexible application behaviors. Through our prototype

implementation and evaluation, we find that Arya provides a promising basis for securing the output of AR applications. Second, this dissertation presents a qualitative user study that my collaborators and I conducted to investigate the security and privacy concerns that users have surrounding emerging AR technologies, in the context of both single-user applications and shared, multi-user experiences. Our study uncovers a wide range of perspectives and concerns, as well as opportunities for further technical defenses. Finally, this dissertation explores the challenge of enabling multiple AR applications to augment a user's world simultaneously, identifies ways in which AR applications may conflict with each other as they attempt to display content, and proposes multiple design paths for AR platforms to better support multi-application ecosystems. By analyzing today's state-of-the-art consumer AR headsets, we discover a nascent multi-application landscape ripe for further exploration. Taken together, these thrusts of research lay a foundation for better understanding the security and privacy risks of emerging AR technologies, and for designing these technologies to better protect users from harm.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

ACKNOWLEDGMENTS

This dissertation represents the culmination of several years of work, and I owe a huge debt of gratitude to the many friends, colleagues, and mentors that have helped me along this journey.

First off, I want to extend a huge thank you to my advisors, Franziska Roesner and Tadayoshi Kohno. I have enjoyed these past 5 years at UW more than I ever imagined I would, due in large part to your guidance, support, and friendship. Thank you for teaching me so much and helping me grow in so many ways, both personally and professionally. It has been an absolute joy and privilege getting to work with you both — I couldn't have asked for better advisors, and I wouldn't be the person I am today without you.

I am also grateful to all of my fellow Security and Privacy lab members, who made our lab feel like a second home all these years: Christine Chen, Camille Cobb, Ivan Evtimov, Earlence Fernandes, Gennie Gebhart, Christine Geeng, Karl Koscher, Ada Lerner, Srirang Mare, Peter Ney, Temitope Oluwafemi, Lucy Simko, Anna Kornfeld Simpson, Ian Smith, Alex Takakuwa, Paul Vines, and Eric Zeng. From deep academic discussions to impromptu games of tiny-table ping pong in the lab and summertime hikes in the mountains, we've had a pretty great run — I'm going to miss you guys. I also want to thank Kimberly Ruth, my collaborator on much of the work that appears within this dissertation.

I want to thank the other members of my thesis committee, Brian Curless and Linda Ng Boyle, for lending me their valuable perspectives surrounding augmented reality. I also want to thank the wonderful staff of CSE that have helped make my time at UW so special: Elise deGoede Dorough, Tracy Erbeck, Melody Kadenko, Alexander Lefort, Lisa Merlin, Lindsay Michimoto, Sophie Ostlund, Andrei Stabrovski, Garrett Yoshitomi, and all the other folks

that help keep CSE running so smoothly. I am also grateful to Laura Dorsey and everyone at UW CoMotion for their help in navigating the research commercialization process.

Prior to graduate school, I had the good fortune to learn from some excellent teachers and mentors who helped shape my academic foundations. I particularly want to thank Landon P. Cox, who took me under his wing at Duke University and mentored me over the course of my years as an undergraduate student there. I also want to thank Eduardo Cuervo for his guidance both while I was at Duke and years later as my internship manager at Microsoft Research. I am also grateful to Ashwin Machanavajjhala for his mentorship during my senior year at Duke, and I thank Peter Gilbert, Valentin Pistol, Nisarg Raval, Ali Razeen, and Animesh Srivastava for showing me the ropes as an undergrad learning to do research. Looking back even further, I want to thank a few special teachers: Jay Wilson, who made AP Calculus not only manageable but incredibly enjoyable; Melinda Fitzgerald, whose 8th grade science class helped foster my own scientific curiosity; and Jeremy Bellion, my middle school band director, for having the tenacity to teach a bunch of rowdy kids to play music and for going the extra mile to run the jazz band club before school.

It has been said that the real PhD is the friends you make along the way, and I am so grateful to all the wonderful friends that have supported me throughout my time at UW. Thank you especially to Lucy Simko and Karl Weintraub, for all the delicious homemade pizza and bagels; to Eric Zeng, for keeping my meme game fresh and up to date; to Suchin Gururangan, for getting me back in the gym; to Maaz Ahmad and Antoine Bosselut, for the good times in office 510; to Eric Whitmire, for introducing me to the wonderful world of puzzle hunts; to Liang Luo, Elizabeth Clark, and everyone else in CSE-tennis, for all the great hitting sessions; to Aidan Ke-Lind, Peregrine Ke-Lind, and Shravan Prasad, for the hilarious JMH gaming nights; to Nihir Patel, Daniel Li, Lauren Caroll, Becky Leylek, and Ani Mohan, for making our days at Duke and after so special; and to everyone else who I am surely forgetting to mention, from UW and before. Thank you all for the great memories.

Finally, I want to thank my family. Thank you to my parents, Mitali and Alvin Lebeck, for always supporting me through the ups and downs of grad school and life, and for setting such excellent examples for me to live by; and thank you to my twin brother, Niel Lebeck, who has been my best friend since the beginning. I can't possibly capture all the inside jokes and life experiences we've shared all these years — thank you for always being there.

DEDICATION

To my parents, Mitali and Alvin Lebeck, for their encouragement, love, and support; and to my twin brother and best friend, Niel Lebeck, for putting up with me all these years.

Chapter 1

INTRODUCTION

Augmented reality, or AR, is a powerful computing paradigm that enables users to interact with digital content in new and exciting ways. AR technologies work in real time to understand the physical world by processing various sensor streams (e.g., video, audio, or depth information), and they overlay digital content such as visual, aural, or haptic feedback atop the user's perception of the world. With countless application domains, AR has the ability to touch many aspects of our lives: for example, video games that place digital content within our physical world [81]; automotive aids that empower drivers to better navigate the roads [82]; medical tools to help doctors more effectively diagnose and treat their patients [39]; and many more. AR is poised to revolutionize how we engage with technology as part of our daily lives, with the potential to deliver tremendous benefits.

Unfortunately, the same capabilities that make AR so powerful — e.g., the ability to continuously monitor a user's environment, and to directly influence how users perceive the physical world — also put AR technologies in uniquely privileged positions to negatively impact the security and privacy of users [25, 27, 95]. We currently lack a deep understanding of the ways in which AR may leave users vulnerable to harm, or how to best defend against such harms. This dissertation aims to help develop such an understanding, by identifying and addressing several key gaps in the AR security and privacy landscape, which are informed by recent trends in the evolution of these emerging technologies. I elaborate on these gaps and trends below, after which I summarize the contributions that this dissertation provides.

1.1 Motivating Trends in AR and Gaps in the Security and Privacy Landscape

Several recent trends in AR motivate this dissertation and give rise to important gaps in the AR security and privacy landscape, which I present below.

1.1.1 From 2D Screens to Immersive Worlds

Early-stage consumer AR devices, such as the XBox Kinect and Google Glass, were primarily characterized by their continuous sensing capabilities, which allowed applications to understand information about a user's surroundings in real time. However, in terms of the output that applications display to the user, these devices had relatively limited capabilities. For example, XBox applications display content on television screens, and the Google Glass has a small 2D display that lies in the corner of a user's vision. The full potential of AR extends far beyond the traditional displays that we are accustomed to, encompassing devices that can immerse users in new digital worlds that are more seamlessly blended into the real world. Indeed, we have seen examples of such technologies emerge in recent years, with immersive AR headsets such as the HoloLens [50] and Meta 2 [73] releasing in 2016.

Gap 1: The Risks of Immersive AR Output. As AR technologies gain the ability to influence how users perceive the world in more and more powerful ways, they may also leave users vulnerable to new types of harm stemming from malicious or buggy applications that abuse these sophisticated output capabilities. For example, an automotive AR application could intentionally or accidentally obscure critical content from the driver's view, such as oncoming vehicles, pedestrians, or road signs, causing harm to the user or bystanders.

1.1.2 From Individual to Shared Experiences

AR has the potential to serve not only as a computing paradigm for individual users to experience in isolation, but also as a tool for enabling new types of shared experiences between *multiple* users. This trend has begun to manifest more and more in recent years, ranging from applications such as the incredibly popular smartphone AR app Pokémon

Go [81] to collaborative AR tools for enterprise settings [119].

Gap 2: Security and Privacy for Shared AR Experiences. From a security and privacy perspective, the evolution of shared AR experiences creates a new dimension of risks and challenges to consider. Rather than focusing only on protecting individual devices or users, we must also consider how users *themselves* could negatively impact each other as they engage in more multi-user contexts. Early signs of these issues have already begun to emerge in the wild — for example, users reportedly vandalized shared AR artwork created in SnapChat [70], and there have been various instances of misbehavior in related domains such as shared virtual reality environments [3].

Gap 3: Understanding Users' Concerns. Immersive AR devices such as the HoloLens have only recently become available. As such, prior work predominately conjectures AR security and privacy risks that may arise in anticipation of these emerging technologies, but it does not directly study the security and privacy concerns of users themselves surrounding the recent wave of emerging AR technologies. Understanding the perspectives of users, in the contexts of both individual and shared, multi-user experiences, can reinforce technical directions being pursued in the research community and industry, or it can shape priorities and suggest new areas for further exploration.

1.1.3 From Single-App to Multi-App Ecosystems

The final trend that this dissertation identifies is a promising future direction for AR technologies. Specifically, AR users may benefit from the ability to engage with multiple immersive applications simultaneously, without having to exclusively choose between them. Unfortunately, most modern AR platforms do not provide multi-app support, and those that do still have significant limitations. Consider a user who wishes to engage with multiple apps while walking in a city, such as an AR navigation app [44], an AR game [81], and social apps that augment nearby people, e.g., by displaying their names above their heads or 3D masks over their faces. On a single-app platform, the user can only view and interact with one app

at a time. By contrast, a multi-app platform could allow the user to shift their attention between apps — for example, periodically glancing at directions without closing their game, while still seeing social overlays on nearby people.

Gap 4: Output Conflicts between Applications. Despite the potential benefits of multi-app AR, the ability of different apps to simultaneously augment a user's world raises critical questions: how might apps visually conflict with each other as they compete for space to display content (either intentionally or unintentionally), what ramifications might these conflicts have for users, and how can we design AR platforms to support rich behaviors while mediating conflicts that might occur? These challenges represent critical barriers to achieving robust multi-application support.

1.2 Contributions

Without a better understanding of the above gaps, as well as concerted efforts to address them, these challenges may severely impede the continued progress of AR technologies. This dissertation thus presents three thrusts of research that my collaborators and I conducted over the past several years to motivate and address these key issues. Specifically, this dissertation contributes the following:

1. Chapter 3 of this dissertation presents the first academic work, to our knowledge, to systematically explore and address the challenge of securing AR output in the face of malicious or buggy applications (gap 1). It describes the design, prototype implementation, and evaluation of Arya, an AR platform that my collaborators and I designed to manage the output of AR applications. Central to Arya is the *AR object* abstraction — new primitives that we designed to encapsulate properties of virtual content generated by apps (e.g., size and opacity), which give Arya fine-grained control over app outputs. Arya governs the output behaviors of apps by enforcing policies, which act as behavioral constraints on applications and their AR objects. In developing Arya, we identified numerous design trade-offs and challenges involved with balancing robust

output management with support for flexible application functionality.

2. To bridge gaps 2 and 3, Chapter 4 presents a qualitative lab study that my collaborators and I conducted, wherein we investigated the concerns of end users grounded in interactions with real AR technologies (specifically the Microsoft HoloLens) across both individual and shared experiences. Through semi-structured interviews, we explored participants' security, privacy, and other concerns, raising key findings. For example, we found that despite the HoloLens's limitations, participants were easily immersed, treating virtual objects as real (e.g., stepping around them for fear of tripping). We also uncovered numerous security, privacy, and safety concerns unique to AR (e.g., deceptive virtual objects misleading users about the real world), and a need for access control among users to manage shared physical spaces and virtual content embedded in those spaces. Based upon our findings, Chapter 4 identifies key challenges and lessons to inform the design of emerging AR technologies.
3. Finally, Chapter 5 addresses gap 4 by identifying means of visual conflict that may arise between simultaneously-running AR applications and proposing design strategies for AR platforms to mediate conflicts. It then analyzes some of today's state-of-the-art consumer AR platforms (the HoloLens, Meta 2, and the more recently-released Magic Leap One) to develop an understanding of their design choices and trade-offs. In doing so, Chapter 5 identifies unexplored gaps in the broader multi-application AR design space and reveals key guidelines to inform future multi-app AR efforts.

Chapter 2 provides additional context on augmented reality and prior research efforts that address additional related challenges surrounding AR. Chapters 3–5 present the above contributions in greater depth and provide a foundation for understanding and addressing critical security and privacy challenges raised by emerging AR technologies, while Chapter 6 summarizes and concludes.

Chapter 2
BACKGROUND AND RELATED WORK

In this chapter, I begin with background context on augmented reality and its evolution from early beginnings in the research community to the expansive industry of today. I then discuss prior work within the AR security and privacy space, followed by additional research efforts involving user experiences and display interfaces for AR that are related to this dissertation outside the scope of security and privacy.

2.1 The Rise of AR

The genesis of augmented reality can be traced back to the '60s, when Sutherland presented the first head-mounted display capable of displaying virtual, 3D objects atop a user's view of the physical world [103, 104], although the term "augmented reality" itself was not coined until the '90s [19]. Since the inception of AR, countless research efforts have sought to address a vast array of technical challenges, ranging from user input and display technologies to AR development tools, registration (properly aligning virtual content with the physical world), sensing techniques, and numerous other concepts comprehensively surveyed in prior work (e.g., [8, 9, 12, 87, 116, 121]). These works also survey an extensive body of research that explores the potential benefits of AR across a diverse spectrum of application domains, spanning medical assistance, manufacturing, robotics, entertainment, workplace collaboration, education, personal assistance, automotive assistance, and many more domains.

While AR research has steadily progressed, we have begun to see explosive growth in industry AR efforts over the past decade, initially catalyzed by the widespread proliferation of smartphones. For example, the smartphone AR app Pokemon Go [81] exemplified this growth, becoming the fastest smartphone game to reach one billion dollars in revenue [98].

With smartphone AR development platforms such as ARKit for iOS [6], ARCore for Android [5], and Facebook's AR Studio [7], creating mobile AR experiences has never been easier for developers.

2.2 Emerging AR Technologies

The full potential of AR extends far beyond the traditional mobile devices that we are accustomed to, and a new wave of emerging AR technologies has begun to harness this potential. Emerging AR technologies are characterized by two key aspects: first, they possess sophisticated technical capabilities that allow them to act as more natural extensions of a user's physical world, such as immersive displays that encompass a user's vision and can more seamlessly integrate digital content into the user's environment, powerful sensors that can reconstruct the 3D geometry of a user's physical surroundings and understand the semantic context of the user's world, and networked capabilities that allow multiple users to engage with each other in shared digital worlds embedded within the physical world. Prominent examples of AR technologies moving in this direction include today's state-of-the-art headsets such as the Microsoft HoloLens [50], Meta 2 [73], and Magic Leap One [69].

Second, another key aspect of emerging AR technologies is their increasing ubiquity across a diverse array of applications domains. For example, the HoloLens is being used by NASA's Jet Propulsion Laboratory to guide astronauts through complex tasks [80], by the Israeli military to manipulate terrain models and monitor troop positions [1], and across a wide range of other industries such as medicine [39]. Multiple organizations within the automotive industry have also begun exploring opportunities for AR to assist drivers [82]. Haeuslschmid et al. [47] describe a broad taxonomy of AR windshield applications grounded in existing literature, ranging from safety-oriented apps (e.g., highlighting lane markers to warn a driver of accidental lane drift) to navigation apps (e.g., path finding with 3D navigation arrows). Recent demos from Hyundai [53] (shown in Figure 3.1) and Continental [23] demonstrate the capabilities of early-stage AR windshields, and organizations such as BMW [10] and Honda Research [110] continue to push the boundaries of automotive AR.

AR has the potential to catalyze positive change across a wide variety of application domains, as foreshadowed by the vast array of research and industry efforts. Moreover, today's emerging AR technologies suggest important evolutionary trends in AR that have begun to manifest and that are likely to continue in coming years, as discussed in Chapter 1. Unfortunately, as the technical capabilities of AR devices continue to mature, and as these technologies become more ubiquitous components of our day-to-day lives, they may expose users and non-user bystanders to new security and privacy risks. This dissertation provides a foundation for proactively understanding and addressing such risks, motivated by the important evolutionary trends mentioned above. Below, I discuss prior thrusts of research that complement this dissertation.

2.3 AR Security and Privacy

Roesner et al. [95] and D'Antoni et al. [25] conceptually surveyed many risks that AR technologies may raise, and additional prior research efforts predominately focus on addressing a relatively limited set of security and privacy challenges related to AR (discussed below and surveyed in [27]). While these works provide conceptual bases for identifying potential risks and for technically addressing certain specific challenges, this dissertation is the first work (to our knowledge) to deeply and systematically investigate the security and privacy challenges that emerging AR technologies present from a broader perspective.

2.3.1 Sensor Privacy for Perceptual Computing

AR devices are examples of *perceptual computing* technologies, which have the capabilities to continuously monitor and interpret information about a user or their surroundings via sensors such as cameras or microphones. Left unchecked, applications with these capabilities can learn sensitive information that infringes on the privacy of the user or bystanders. For example, prior works have shown that a recording device with unfiltered camera access could reconstruct data such as: input typed on a bystander's smartphone, using reflections of the phone on nearby surfaces [88]; audio feeds, using high-speed videos of the vibrations of

inanimate objects placed near the audio sources [26]; or the contents of transparent near-eye displays worn by nearby users [60]. Others have shown that a malicious application on a user's own smartphone, with access to the camera, could reconstruct a rich three-dimensional model of the user's environment using surreptitiously-taken 2D photos [109]. A major thrust of prior work thus focuses on technical defenses to mitigate the privacy risks of continuous sensing capabilities on perceptual computing devices.

Filtering Sensor Feeds. Jana et al. [55, 56] observed that many legitimate perceptual computing applications do not actually require access to raw sensor feeds, but are instead interested in higher-level information embedded within these feeds (e.g., a facial recognition app may only be concerned with faces within a video stream). Prior efforts have thus focused on designing least privilege solutions for mediating applications' access to sensor data.

Surroundweb [117] is a least-privilege approach to allowing 3D web browsers to display web content within a physical room, by only exposing the dimensions and locations of flat surfaces to apps, and by allowing apps to declaratively place content relative to objects in a room (without revealing the presence of those objects) via sandboxed code. Relatedly, McPherson et al. [72] also analyzed the security of AR browsers, uncovering issues such as liberal permissions that allow various parties to access devices' sensor feeds, which can compromise the privacy of users and bystanders.

Taking another least-privilege approach, DARKLY [56] is a modification of the OpenCV computer vision library that applies privacy-preserving image transforms to a camera feed while allowing apps that utilize OpenCV to run unmodified. In [55], Jana et al. instead proposed the recognizer abstraction — trusted OS modules that process raw sensor streams and extract higher-level information (e.g., human faces), which multiple applications may subscribe to, providing a least-privilege approach to sensor access that scales to multiple applications. Roesner et al. expanded upon the recognizer abstraction with world-driven access control [96], a framework that allows a user's environment to communicate policies specifying access control decisions about real-world objects (e.g., to blur faces). As discussed

in Chapter 3, my collaborators and I leverage recognizers for our Arya system not only as a privacy-preserving tool, but also as a mechanism for Arya itself to develop an understanding of a user's surroundings.

Focusing specifically on depth streams, Figueiredo et al. [36] introduced Prepose, a domain specific language and runtime for writing gesture recognizers. Among other features, Prepose validates properties of gestures such as physical safety (i.e., ensuring the user does not have to overextend her arms), ensures custom gestures do not conflict with protected system gestures (e.g., the HoloLens bloom gesture), and allows multiple concurrently running applications to register gestures.

Leveraging Machine Learning. Others have explored least-privilege approaches for sensor data that involve a combination of machine learning and minimal user interaction. Templeman et al. proposed PlaceAvoider [108], a framework that allows users to blacklist sensitive spaces by photographing them (e.g., a bedroom or bathroom) and uses machine learning techniques to recognize images taken within these sensitive spaces, preventing untrusted apps from accessing these images. Similarly, Zarepour et al. [120] detect sensitive objects within images (e.g., license plates) and sanitize the data, e.g., by blurring the objects. Raval et al. [91] instead proposed privacy markers that consist of (a) a marking interface for users to whitelist physical objects they wish to reveal to applications, and (b) software that efficiently recognizes the marked objects. Additionally, Raval et al. [90] proposed a game-theoretic approach using neural networks to protect visual secrets, by formulating the problem of image perturbation as a game between an attacker seeking to identify sensitive information within an image, and an obfuscator seeking to protect that information.

Privacy Indicators. Rather than using computational techniques to filter information out of sensor streams, another avenue of work has explored ways to inform and empower users via privacy indicators. Egelman et al. [32] conducted a crowdsourcing study to investigate privacy indicators for ubiquitous computing applications (including wearable recording devices like AR headsets) that more effectively and transparently communicate the data they are

accessing, and that may help users make more informed privacy decisions. Privacy indicators have been well studied in other contexts such as the web [33, 111]; however, the effectiveness of these indicators has been called into question [24], with prior work showing that privacy indicators such as the MacBook webcam recording LED can be compromised [14].

2.3.2 Security and Privacy for Multi-User Interactions

As I discuss further in Chapter 4, AR can be used as a tool to foster collaboration between multiple users. However, such interactions may also leave users vulnerable to harm. A limited body of prior work has explored security and privacy challenges for multi-user digital interactions, primarily in the context of video conferencing and early-stage AR technologies.

Multi-User Privacy. Butz et al. [15, 16] introduced abstractions for virtual object privacy in a shared 3D AR environment — *privacy lamps*, which illuminate a region of space within which any object is considered private, and *vampire mirrors*, which reveal limited information about private objects to other users (e.g., by making the objects invisible or transparent).

In the context of video conferencing, Devincenzi et al. [30] introduced Kinected Conference, a framework that enables privacy-preserving video conferencing by, for example, leveraging a depth stream to selectively freeze video pixels at certain depths (e.g., to hide background information while the foreground video stream continues uninterrupted). Ens et al. [34] also proposed a new class of computing called *candid computing*, where a user's device provides other users with whom the user is interacting with information about the user's actions (e.g., by augmenting a user's image with coarse-grained information about the task they are currently engaging in without revealing details of the task).

The above works focus largely on technical defenses for a limited set of multi-user privacy issues. By contrast, this dissertation is the first work to our knowledge that focuses on broadly surfacing security and privacy concerns around modern emerging multi-user AR systems, based upon interviews with end users themselves in the context of hands-on interactions with today's state-of-the-art AR devices.

Secure AR Device Pairing. Limited prior works have investigated the problem of secure device pairing for emerging AR technologies like the HoloLens, to enable more secure multi-user interactions. Gaebel et al. [40] propose a pairing protocol that leverages wireless localization techniques combined with facial recognition to authenticate HoloLens users to each other. Sluganovic et al. [101] take a different approach to tackling the device pairing problem, instead allowing HoloLens users to authenticate each other using precisely positioned, shared virtual objects.

2.3.3 Understanding Privacy Perceptions

Another thrust of prior work aims to better understand the privacy concerns of users and bystanders of wearable technologies beyond AR specifically. For example, Denning et al. [29] conducted an in-situ study of bystander reactions to a mock-up AR device similar to Google Glass, and Hoyle et al. [51] conducted an in-situ study of users of lifelogging camera devices, including an exploration of the ways users manage the flow of personal information collected by their devices.

Lee et al. [66] surveyed over 1700 users of wearable technologies to understand their risk perceptions and the types of data collected by wearables that users would find most upsetting, including data such as photos or videos of the user unclothed or of sensitive financial information. Motti et al. [78] also uncovered a diverse set of themes by studying users' privacy concerns about wearables through a qualitative study of online comments posted by wearable device users — for example, fears of surveillance, surreptitious video recording, and facial recognition identifiability.

The above works highlight the importance that wearable device users place on their privacy, and the ways in which they fear their privacy might be compromised. However, no prior work has yet applied a similar lens to emerging AR technologies, which provide novel capabilities such as sophisticated visual output in addition to continuous sensing capabilities. Chapter 4 of this dissertation thus seeks to uncover end user security and privacy concerns of emerging AR technologies, grounded in users' hands-on experiences with these technologies.

2.3.4 Policy and Design

Calo et al. [17] and Roesner et al. [94] explored legal and policy challenges for AR grounded in the novel technical capabilities of these systems. For example, the authors discussed how existing regulations may or may not map to issues engendered by both the collection of information (e.g., the reasonable expectation of privacy doctrine) as well as the display of information (e.g., digital assault).

From a design perspective, Greenberg et al. [45] identified dark patterns for ubiquitous computing technologies — user interfaces that are designed to trick users into performing actions that may not be in their best interests. The melding of physical and digital information that AR technologies provide may leave users vulnerable to such manipulation. To prevent the abuse or manipulation of users, Friedman et al. [38] proposed applying principles of value sensitive design — designing technology that accounts for human values — to AR technologies. For example, they consider the human values of psychological and physical well-being, the need for privacy, and minimizing the risk of deception.

To our knowledge, this dissertation is the first to systematically characterize the risks that users themselves perceive of emerging AR technologies (Chapter 4), and to propose technical defenses that enable AR to better support the human values mentioned above by mitigating threats that may infringe upon these values (Chapters 3 and 5).

2.4 User Experiences and Display Interfaces for AR

Below, I discuss additional prior research efforts surrounding user experiences and display interfaces for AR that are related to portions of this dissertation.

2.4.1 User Experiences of Mobile AR

Prior work studied user experiences of AR, focusing primarily on early-stage mobile AR technology. In 2005, Swan et al. [105] conducted a survey of user-based experimentation of AR, finding that user-based AR research was progressing in three primary directions — studying

low-level tasks to understand how human perception functions in AR contexts, examining user task performance within specific application domains, and studying interactions between collaborating users.

Since then, Olsson et al. have explored the space of AR user experiences through focus groups [84] and semi-structured interviews [85], identifying aspects of AR experiences that people would find either positive (e.g., support for goal-oriented tasks or stimulation of positive emotions) or negative (e.g., overwhelming amounts of content or asocial experiences). Olsson further supports these findings by characterizing satisfying and unsatisfying experiences of former mobile AR app users [86].

Rauschnabel et al. [89] identified factors that may drive the adoption of AR headsets, such as ease of use and social norms. Focusing specifically on smartphone-based AR, Irshad et al. [54] conducted a lab study wherein they introduced participants to mobile AR applications and subsequently conducted a post-activity survey to study the users' experiences.

While these studies provide a valuable conceptual foundation, they focus on early-stage mobile AR, and they lack a focus on security and privacy concerns. This dissertation comes at a different time in the evolution of AR technologies, allowing my collaborators and I to study users directly interacting with immersive AR devices and to explore security and privacy concerns around these technologies more deeply.

2.4.2 Multi-User Experiences

While only limited prior work has explored the security and privacy challenges raised in multi-user AR environments, multi-user AR more broadly remains a rich area of work. Billinghurst et al. [13] surveyed research efforts in the space of collaborative augmented reality interfaces, both for face-to-face and remote interactions. For example, Studierstube [107] is one of the earliest general-purpose architectures for multi-user AR that allows physically co-located users to view shared 3D content through head-mounted displays (HMDs). Regenbrecht et al. [92] presented another system wherein users with HMDs can view and interact with shared 3D content tethered to a turntable-shaped device atop a table. Additionally, Billinghurst

et al. [13] developed two prototype collaborative AR interfaces: WearCom, an interface for remote multi-party conferencing that displays virtual avatars in 3D space, and Collaborative Web Space, an interface that allows physically co-located users to collaboratively browse web pages.

For more domain-specific uses, Kaufmann et al. [58, 59] explored the use of collaborative AR for education, while Ohshima et al. [83] explored the challenges of creating an AR air hockey system. Reitmayr et al. [93] developed a collaborative navigation tool for tourists, where HMD users can set waypoints in physical space that other users can visualize and follow. In [49], Henrysson et al. presented the first face to face collaborative AR app based on mobile phones rather than HMDs. Finally, looking instead at remote AR rather than physically co-located users, Kato et al. [57] described an AR conferencing system that allows remote collaborators to collaboratively view and edit virtual objects on shared virtual whiteboards. This diversity of efforts highlights the vast potential of collaborative AR, and further motivates a need to understand and defend against the potential security and privacy issues that users might face within these ecosystems.

Further afield from AR, digital interactions between physically co-located users have been studied in the context of interactive tabletop interfaces, including the challenges of governing personal territory [100] and preventing conflicts between users [77]. This dissertation identifies related challenges in multi-user AR settings.

2.4.3 Multi-Application Display Interfaces

Chapter 5 focuses specifically on enabling multi-application AR ecosystems, which involves careful consideration of the display interfaces that AR platforms may provide to users. Researchers have previously proposed AR systems that support multiple simultaneously-running applications in limited capacities. For example, Argon [68] instantiates multi-application support by allowing apps to run within overlapping, full-screen, transparent windows. By contrast, Studierstube [107] confines app outputs to bounded 3D windows controlled by the user. Earlier non-AR efforts also considered secure windowing (e.g., [35]).

While these prior works represent individual approaches to providing multi-application support, they did not rigorously explore the design space more broadly or reason about conflicts that may arise between the output of different applications, some of which may be malicious or buggy. As discussed further in Chapter 5, different points in the multi-application design space present different trade-offs in terms of balancing application functionality with the ability to mediate conflicts that may arise between the output of different apps.

2.5 Summary

Stepping back, the AR space remains a rich area of study, and prior efforts have pursued many important research directions related to various aspects of these technologies. However, emerging AR technologies raise critical new security and privacy challenges that have yet to be fully understood or addressed. This dissertation comes at a formative time to shape our understanding of these challenges, and it provides an initial blueprint for addressing them.

Chapter 3

SECURING AUGMENTED REALITY OUTPUT

This chapter focuses on the first key challenge that Chapter 1 describes — namely, addressing the security risks of AR *output*, or the risks that arise from the ability of buggy or malicious AR applications to modify how users perceive the physical world. This chapter characterizes these risks, identifies important challenges involved with preventing such risks, and describes the design and prototype implementation of Arya — an AR platform that my collaborators and I created to constrain the output capabilities of applications according to various policies, which dictate how apps can behave. Components of this work originally appeared in the 17^{th} Workshop on Mobile Computing Systems and Applications [61] and the 38^{th} IEEE Symposium on Security and Privacy [63], and this work was also invited for publication in IEEE Security & Privacy Volume 16, Issue 1 [64].

3.1 Overview

Output from malicious or buggy AR applications may expose users to serious forms of harm, particularly on immersive AR systems, such as head-mounted displays (HMDs) and car windshields, where users cannot easily disengage from their devices if output security issues arise. To illustrate these risks, imagine driving a car with an AR-enabled windshield. The intended benefits of this technology may include the ability to visibly highlight lane markers to prevent accidental lane drift, to display turn-by-turn driving directions visually overlaid on the road, and to visibly warn the driver of impending collisions — examples already showcased by industry, e.g., [53] (see also Figure 3.1). These tasks might run as multiple components of a single application, or as multiple, distinct applications. Without appropriate safeguards, however, the benefits of these applications can be overshadowed by risks. A malicious or

Figure 3.1: **Example AR Scenario.** This screenshot from Hyundai's CES demo [53] shows an AR warning overlaid on a car and the car's current speed.

buggy AR application could potentially obscure real-world pedestrians, overlay misleading information on real-world road signs, or occlude the virtual content of other AR applications, such as collision warnings or other important safety alerts. Similar issues could arise with HMDs for a user on foot. Consider, for example, an HMD application that accidentally or intentionally blocks the user's view of a tripping hazard or an oncoming car. The ability of AR content to obscure real-world objects is not hypothetical, as Figure 3.2 shows.

To our knowledge, no existing industry or research AR platforms are designed to mitigate the above types of output security risks. Today, it is the responsibility of the applications themselves to safely generate output and to adhere to guidelines, such as those suggested for HoloLens developers [74]. For instance, these guidelines suggest that applications should not create AR content that covers too much of the user's view of the world, but the HoloLens itself does not enforce this. Placing this responsibility on application developers, who may generate buggy, vulnerable, or malicious code, is problematic. Furthermore, the fact that today's AR platforms cannot exert any control over the output of individual applications means they also cannot handle conflicts between the output of multiple applications (a problem that I discuss in greater depth in Chapter 5).

Our Work: Designing for Secure AR Output. We seek to change the above situation.

Figure 3.2: **Real-World Occlusion.** This photo was taken by a smartphone through a HoloLens display (resulting in some reflective camera artifacts). It shows that virtual content displayed by HoloLens (here, a cat) can visually obscure real-world objects (also a cat).

Specifically, we design, implement, and evaluate a prototype AR platform with output security as an explicit, first-class goal. We refer to our design as Arya. In our threat model, Arya is trusted, but the AR applications running on Arya are untrusted. With Arya's security mechanisms enabled, applications still have significant flexibility to create immersive AR experiences, but their visual content is constrained by the platform based on *policies*, such as ensuring that windshield applications cannot obscure real-world road signs or pedestrians while the car is moving. This work both identifies and overcomes numerous challenges towards designing AR systems to mitigate output security risks.

Our core design builds upon the designs of prior AR systems and includes *sensors*, such as cameras and microphones; *recognizers* [55] to detect objects, such as cars and people, from the sensed input; and an *input policy module* [96] to determine which of the sensed objects should be passed to applications, possibly with modification. The central difference in Arya is the inclusion of an *output policy module* that sits between applications and the AR system's output drivers, and that enforces policy-based constraints on application outputs. We find that designing an output policy module is fundamentally challenging and requires identifying and answering key design questions, such as how to express desired output policies, how to

enforce those policies, and how to handle conflicts between different policies.

We identify and overcome these challenges through the iterative design, implementation, and evaluation of Arya and our Arya prototype. Arya instantiates the *AR object* abstraction — new primitives we designed to encapsulate properties of application content (e.g., the size, position, and opacity of individual virtual objects), which allow Arya to exert fine-grained control over the output of applications. We also develop a set of case study output policies based on existing policies drawn from several sources, including the HoloLens developer guidelines [74] and guidelines for the visibility of road signs [21]. For example, we use a guideline that real-world trees should not block road signs to inspire a policy that AR objects should not block real-world road signs. To support such policies, we design an *AR output policy specification framework* that allows policy writers to specify both (1) a condition under which the policy is violated (e.g., when an AR object blocks a real-world person) and (2) an action to take (e.g., make the offending AR object partially transparent). We carefully constrain this policy framework to support composable policies and to limit the potential performance or other impacts of buggy or malicious policies. We do not specify where policies come from in this work — they may come from the device manufacturer itself or other sources.

We develop our prototype atop the Unity game engine [112], an environment for creating interactive 3D content. To evaluate the output management portion of Arya through controlled experiments that simulate different real-world contexts, we develop virtual Unity scenes rather than using real-world sensor input. Our scenes represent HMD and car windshield AR scenarios, and we develop a set of case study applications that run within these scenarios. We demonstrate that our prototype can support the policies we identify and prevent corresponding undesirable situations in our case studies. We conduct a performance evaluation consisting of both microbenchmarks and a full system evaluation, and we find that the performance impact of policy enforcement in even our unoptimized prototype is acceptable. Our prototype played a central role in iteratively driving the design of Arya, and our design choices and evaluation findings provide lessons for future AR system designers.

Contributions. In summary, we contribute the following:

1. *AR Output Security*: We address the fundamental challenge of securing AR output for the first time, by designing Arya — an AR platform that can exert fine-grained control over the visual output of applications by enforcing output policies that govern how apps can behave.
2. *AR Output Policies*: We develop a policy specification framework for defining output policies that is designed to provide desirable properties (e.g., to limit performance impact and support composable policies). Through our design process, we uncover and overcome fundamental challenges in realizing the above vision, including how to specify and enforce policies, and how to handle conflicting policies. Despite its restrictions, we demonstrate that our framework can support real policies from multiple sources, such as the HoloLens developer guidelines and U.S. Department of Transportation guidelines for in-vehicle electronic devices.
3. *Prototype, Evaluation, and Lessons*: We prototype Arya on top of the Unity game engine and develop case study applications and policies for both HMD and automotive AR scenarios. We conduct benchmark and full system evaluations, finding the performance of policy enforcement acceptable. From our experiences, we surface lessons and recommendations for future AR systems.

3.2 Motivation and Threat Model

As discussed in Chapter 1, emerging AR platforms support fundamentally new types of applications that can respond contextually to input from a user's ever-changing environment, and that can directly alter the user's perception of his or her world with visual, auditory, or haptic output. Since today's AR devices primarily rely on immersive visual feedback, we focus most of our concrete discussions on visual output, though we note that similar issues may apply to other output modalities (e.g., audio or haptic).

Though emerging AR platforms and applications hold great promise, these technologies are still young and under active development. In particular, along with their novel opportu-

nities, AR applications have a unique ability to impact users' perceptions of the real world in undesirable or harmful ways. To understand these risks, consider the popular mobile AR app Pokémon Go. While this game is a relatively simple smartphone app today, it provides a taste of how emerging platforms like HoloLens will be able to capture the attention of users [97]. In contrast to smartphones, HMDs provide continuous, fully immersive experiences by enveloping a user's entire field of view. With these emerging HMD platforms, we envision that a user may also wish to multitask while playing a game like Pokémon Go — for example, by simultaneously using another app that overlays walking directions to nearby restaurants, or by using a labelling app to recognize and point out nearby social media contacts (a topic discussed more fully in Chapter 5). To reap the full benefits of these apps, the user will need to use them while actively moving about and interacting with the real world.

The interaction of AR apps with each other and with the user's view of the real world raises risks. If one of the apps were malicious or buggy, it could (a) annoy or distract the user with spurious content (e.g., poorly-placed ads), (b) endanger the user by occluding critical information in the real world (e.g., by obscuring oncoming vehicles), or (c) perform a denial of service attack on another application by occluding that application's output (e.g., a Pokémon creature that prevents the user from seeing navigation directions). Indeed, a recent concept video sketches out a possible future in which AR technologies fail to address these types of threats, as shown in Figure 3.3. While we describe these risks in terms of an HMD platform here, we stress that they extend across platforms and domains, such as AR-enabled windshields, which — like HMDs — are fully immersive.

Thus, the high-level challenge we address in this work is how an AR platform should constrain the output behaviors of potentially buggy, malicious, or compromised applications. We argue that emerging and future AR platforms *must* address these questions if they wish to support rich, untrusted applications that can be safely used while the user interacts with the physical world (e.g., while walking or driving, not only while sitting at a desk). We observe that undesirable output is not a new concern in and of itself: recall the early days of the web, when web applications frequently opened popups and used blink tags. Browser

Figure 3.3: **AR Concept Image.** This concept image of an AR user on a bus could represent a possible future in which AR output remains unregulated, leaving users unable to control the intrusiveness of AR applications. Full video available at `http://www.theverge.com/2016/5/20/11719244/hyper-reality-augmented-short-film`

vendors eventually constrained these undesirable behaviors by enabling popup blocking by default [79] and by obsoleting the blink tag. Unlike misbehaving apps on the early web, the effects of problematic AR output can range from minor annoyance to direct physical harm.

Threat Model. The above risks inform our threat model and security goals. Specifically, we consider one or more malicious, buggy, or compromised applications that create AR content, which may intentionally or accidentally:

- *Obscure important real-world content*, such as traffic signs, cars, or people.
- *Disrupt the user physiologically*, such as by startling them (e.g., by suddenly creating or quickly repositioning virtual objects).
- *Obscure another application's virtual content*, in order to hide or modify its meaning. While this chapter does consider visual conflicts between apps, Chapter 5 explores this particular challenge in much greater depth.

To combat these threats, we design Arya, an AR platform with a centralized, trusted *output policy module* that enforces policies on AR content. These policies aim to mitigate the above classes of threats, e.g., by preventing applications from blocking important real-

world information, such as people, with AR content. Arya handles policies that can constrain when and where applications display content; it does not support policies that constrain *what* content is displayed (e.g., a 3D animal versus a 3D rock).

We assume that Arya's operating system, drivers, and platform hardware are trusted. However, applications are not trusted by the system. Specifically, we assume that applications may be intentionally malicious, unintentionally buggy, or compromised, potentially leading to undesirable AR output. For example, an adversary might attempt to sneak an intentionally malicious application onto an open platform's app store (like the HoloLens app store), or different trusted development teams within a closed AR platform (e.g., a closed automotive AR platform) might produce applications that interact with each other unexpectedly in undesirable ways.

We also assume that Arya's operating system employs traditional, standard security best practices, e.g., application isolation. In this work, we focus only on threats between applications as they relate to the interaction of their AR output. Additionally, we do not address the question of how the AR output policies that Arya enforces are distributed. We assume that these policies may (for example) be pre-loaded by the device's manufacturer, introduced by third-party sources, or set based on user preferences. We assume that policies *may* be buggy or malicious, and we do not require Arya to trust the sources of these policies. Thus, our design must consider the possibility of malicious or buggy policies.

Finally, we focus specifically on visual AR content, and we consider issues related to non-visual output (e.g., haptic, audio) to be out of scope. However, the lessons we surface through this work may apply to other output modalities as well.

3.3 Design: Arya

We now present the design of Arya, an AR platform architecture with output security as a first-class goal. In designing Arya, we identify and address fundamental new design challenges that future AR platforms must consider if they wish to constrain AR application output. We begin with a high-level overview of Arya in Section 3.3.1, summarized in Figure 3.4, before

describing its constituent components and the technical challenges they address in greater depth.

3.3.1 System Overview

AR applications fundamentally require the ability to continuously capture and process sensor inputs, and to superimpose virtual output on the user's view of the world. Consider the collision warning application in Figure 3.1. This application must know when the user moves too close to another car so that it can display a warning whenever the user is at risk for a collision. However, the user's view of the real world is constantly in flux — the user may change lanes, or other cars may move in front of the user. Furthermore, applications may need to dynamically generate and update visual content in response to these changes — e.g., to display a warning when a collision is imminent. When this content is generated, Arya may also need to modify it to ensure that the warning does not occlude any pedestrians that stumble into the road, or impede the driver's view of the car that he or she is about to hit.

Arya consists of the following core modules, shown in Figure 3.4, that it employs to both support and constrain application behaviors in the face of a dynamically changing environment:

- *System Sensors and Recognizers*, to gather and interpret sensor data from the real world.
- The *Input Policy Module*, to filter and dispatch these data to applications that require access.
- The *Output Policy Module*, to process any new application requests to create or modify virtual content, and, if applicable, modify this virtual content based on the types of policies we introduce in this chapter.
- *Display Drivers*, to display updated virtual state.

These modules are used to support applications running on Arya that may call APIs to query information about the real world and create or modify virtual objects. Arya steps

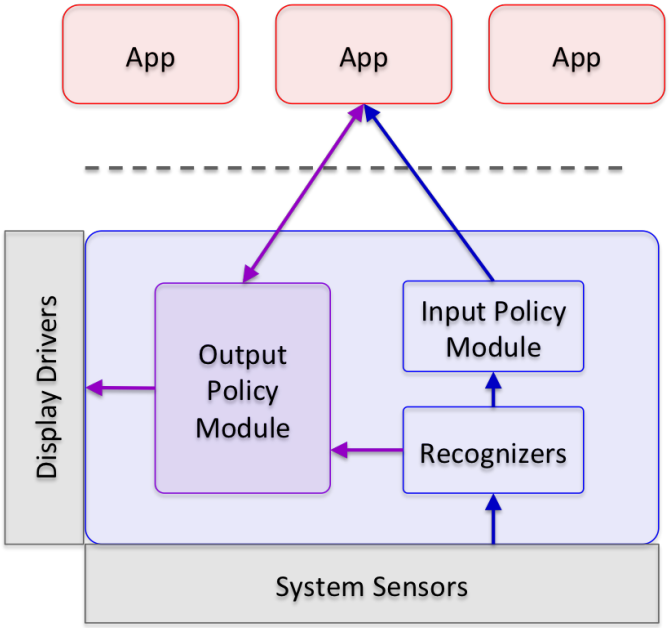


Figure 3.4: **Overview of Arya's Architecture.** We design Arya, an AR platform that consists of (1) system sensors, recognizers, and an input policy module that filters input from the real world, based on prior work (e.g., [55, 91, 96, 108]) and (2) an output policy module that constrains application output. The design of the output policy module is the primary contribution of this work.

through a core workflow to process application requests and produce every output video frame displayed to the user. We first discuss how Arya incorporates prior work to handle input in Section 3.3.2, before turning to our primary contribution — output management — in Section 3.3.3.

3.3.2 Input

Consider again the collision warning application from Figure 3.1. This application must be able to detect nearby vehicles, identify where those vehicles are in relation to the user's view, and determine if a collision is imminent. One way a system might support this capability is to expose the full camera sensor feed to the application, allowing it to perform vehicle detection. However, as prior works note (e.g., [55, 91, 96, 108]), applications that can access raw, unfiltered input from the real world raise serious privacy concerns. Additionally, if

multiple applications need to locate vehicles in the video feed, it would be inefficient for each to implement vehicle detection separately.

To address these privacy and performance issues, prior work [55] proposed *recognizers* for AR platforms: OS modules that process raw sensor streams, detect specific types of information within those streams (e.g., vehicles, people, faces, or planar surfaces), and expose these higher-level objects to applications. Recognizers enable a least-privilege model in which applications can be given access to only those recognized objects that they need. For example, a Pokémon game may not need a full video feed, but rather only information about planar surfaces in the user's view, to sensibly place Pokémon on horizontal surfaces.

In this work, we find that recognizers provide an additional benefit beyond their original purpose of enabling input privacy. Recognizers give Arya itself — and thereby Arya's output policy module — information about the user's real-world surroundings. For example, to support a policy that prevents applications from occluding people, Arya must know whether and where there are people in the user's view. Recognizers provide this information and allow Arya to enforce output policies that depend on the real world.

3.3.3 Output

Recall that our goal in designing Arya is to allow the OS to control the visual output of AR applications. At a high level, we do so by incorporating into the OS an *output policy module*, which controls and modifies AR application outputs according to policies. Before describing these policies and their enforcement in detail in upcoming sections, we describe here the visual output abstractions that Arya exposes to applications.

Foundation: Displaying and Constraining Visual Output. To enable Arya to exert fine-grained control over the output of applications, we introduce a new abstraction that we call *AR objects*. Conceptually, AR objects are OS primitives that encapsulate virtual content that applications wish to overlay on a user's view of the real world. For example, a single Pokémon creature would be an AR object in Arya, and a single application may contain many

such objects. An AR object has a visual representation and associated characteristics, such as size and opacity. AR applications require the ability to create and transform these objects (e.g., by moving, rotating, or resizing them), and Arya supports these common operations.

Additionally, rather than requiring that applications manually update the locations of their objects as the user moves throughout the physical world, Arya allows applications to create "world-locked" objects that are attached to real-world locations or objects, and Arya automatically updates where they are rendered in the user's display. For example, if an AR application attaches a virtual object to a real-world table, Arya can maintain this mapping, not requiring that the application explicitly update how the object is displayed as the user moves. Applications can also create "head-locked" objects that appear at a fixed location in the user's display.[1]

Note that the AR object model differs from the "window" display abstraction traditionally provided to applications, in which applications have full control over a contiguous rectangular screen area. A key benefit of AR objects is that they allow Arya to reason about application output and enforce policies at the granularity of individual objects. For example, if one Pokémon creature obscures a real-world person, Arya can take action against that one object (e.g., to make it transparent) without affecting the rest of the Pokémon application's output. We discuss AR objects in greater depth in [61].

We now turn to the remainder of our design. We present our key design questions, describe the challenges involved in creating an output policy module that constrains AR application output, and surface key design decisions made along the way.

3.3.3.1 Specifying AR Output Policies

Output policies broadly serve to protect AR users from deceptive, discomforting, or harmful content. While AR technologies are still quite young, concretely exploring the policy design

[1]HoloLens similarly supports world-locked and head-locked objects [75]. The key distinction is that Arya supports these features within the OS as part of its output management, while HoloLens does so at the application layer.

Identifier	Description	Applies To	Source
P1	Avoid abrupt movement of AR objects.	Car, HMD	HoloLens Developer Guidelines [74]
P2	Place AR objects at a comfortable viewing distance from the user.	Car, HMD	HoloLens Developer Guidelines [74]
P3	Allow the user to see the real world in the background.	Car, HMD	HoloLens Developer Guidelines [74]
P4	Avoid content that is "head-locked" (fixed location in the display).	HMD	HoloLens Developer Guidelines [74]
P5	Don't display text messages or social media while driving.	Car	NHTSA Distraction Guidelines [115]
P6	Don't obscure pedestrians or road signs.	Car	Tree Visibility Guidelines [21]
P7	Don't obscure exit signs.	HMD	Occupational Safety & Health [114]
P8	Disable user input on transparent AR objects.	Car, HMD	Literature on clickjacking (e.g., [52])
P9	Only allow advertisements to be overlaid on real-world billboards.	Car, HMD	N/A (New)
P10	Don't allow AR objects to occlude other AR objects.	Car, HMD	N/A (New)

Table 3.1: **AR Output Policies.** This table contains a set of policies that we use to drive Arya's design. We identified existing policies from various sources (P1-P8) and, if necessary, modified them to apply to the AR context. We created two additional policies (P9 and P10) motivated by our threat model. Note that NHTSA (the source of P5) is the U.S. Department of Transportation's National Highway Traffic Safety Administration.

space grounded in today's technologies allows us to begin to identify key challenges for future AR systems and to surface initial solutions. Thus, given an output policy module that constrains virtual content in the form of AR objects, our first design question is the following:

Design Question: *How can we translate abstract guidelines into concrete policies that the output policy module can enforce in practice?* To help drive our design around this question, we developed sample output policies for both HMD and automotive AR scenarios. In addition to creating our own policies, we draw on existing sources of guidelines for the relevant scenarios, including the HoloLens developer guidelines (which are suggestions, not technically enforced constraints), the U.S. Department of Transportation guidelines for in-vehicle electronic devices, and guidelines regarding the visibility of street signs. These policies are summarized in Table 3.1.

The first observation we make based on our case study policies in Table 3.1 is that they tell us only what conditions should be *avoided*, not *what to do* when the conditions are met. For

example, we would like Arya's output policy module to prevent applications from creating objects that are too close to the user, take up too much of the user's field of view, block pedestrians, etc. However, existing guidelines do not specify what actions the output policy module should take if an application violates one of these policies. For example, possible actions to enforce policies may include removing, moving, or modifying (e.g., making more transparent) an app's AR objects. We consider these options further below.

Design Decision: Separate Policy Conditions and Mechanisms. The above observation raises an opportunity: the conditions under which policies apply (e.g., when an AR object blocks a real-world person or is drawn too close to the user) and the mechanisms used to enforce the policies (e.g., remove the AR object or make it transparent) can be specified independently and composed as desired.

Specifically, we define AR output policies to consist of two distinct components:

1. A *conditional predicate*, or a boolean expression that determines when a policy should be applied.
2. One or more *mechanisms*, or actions that the output policy module should take when the policy's conditional predicate evaluates to true.

The next design question we face is then the following:

Design Question: *How should policy conditions and mechanisms be expressed?* The most flexible approach would be to allow conditions and mechanisms to consist of arbitrary code, which would clearly support a wide range of policies. However, arbitrary policy code raises several concerns. The first is performance: in the worst case, an arbitrarily-defined policy could halt the system by performing unbounded computation. The second is unexpected results due to buggy or untrusted policies: if policy mechanisms can arbitrarily modify applications' AR objects, then buggy policies could pose the same risks as buggy apps themselves in the worst case.

Design Decision: Restrict Policies. Due to the challenges raised by arbitrary policies, we instead develop an *explicitly restricted policy framework* that requires policies to combine

options from a well-defined set of parameterized conditions and mechanisms supported by Arya. Though this construction is limited by design, we find that it is flexible enough to express the set of desirable policies we developed ourselves and drew from other sources (see Table 3.1).

Policy Conditions. We develop a finite set of building blocks that policies can use to construct conditional predicates. Specifically, we allow policies to refer to *attributes* of objects. We define attributes to be either (1) visual properties of AR objects, such as size, transparency, and speed, or (2) relationships between AR objects and other virtual or real-world objects. For example, relational attributes include `DistanceFromUser()` or `IsOccluding(type)`, where "type" refers to a class of objects against which to check for occlusion (virtual objects or specific types of real-world objects detected by Arya's recognizers, such as people). For non-boolean attributes, a policy's condition is then formed by comparing one or more attributes of an AR object to parameter values specified by the policy — for example, "`if DistanceFromUser() < 10 meters`".

Finally, we allow policy conditions to depend not only on the attributes of AR objects, but also on global contextual information. For example, a policy may depend on properties of the user's platform (e.g., if a user's car is in motion) or other contextual information (e.g., time of day).

Policy Mechanisms. Policy mechanisms are more challenging to design, because they involve not just deriving boolean results, but modifying application behaviors. As mentioned above, possible mechanisms that Arya might support include deleting applications' AR objects (or not allowing them to be created in the first place), modifying them (e.g., to change their transparency or size), or moving them (e.g., away from blocking another object). In experimenting with different possible mechanisms, we identified the following challenge:

Challenge: Conflicting Policies. Since multiple policies may be triggered at once, certain combinations of policy mechanisms may conflict with each other or create a cycle. For example, consider one policy that moves an object away from blocking a person, but causes

it to block a road sign, thereby triggering another policy. Or consider a policy that reduces an object's transparency at the same time as another policy attempts to increase its transparency.

We can address this challenge in one of two ways. First, we could design a method to handle policy conflicts when they arise. However, this raises many additional challenges — for example, what should be done if the conflict cannot be resolved, whether conflict resolution can be performed quickly enough, and how non-conflicting but cyclic policies should be handled. Though there may well be solutions to these challenges (as we elaborate in Section 3.6), in this work we take another approach: we design policy mechanisms such that they *cannot conflict* in the first place.

Design Decision: Composable Policy Mechanisms. It is not immediately obvious how to design policy mechanisms that are composable yet sufficiently flexible to express meaningful policies. However, we observe the following: the goal of our AR output policies is ultimately to ensure that AR applications cannot modify the user's view of the world in dangerous or undesirable ways. Thus, policies should constrain application output to be *less intrusive*, so that the result is closer to an unmodified view of the real world. Based on this observation, we choose to support only policy mechanisms that move AR objects towards a less intrusive state — for example, mechanisms that make objects smaller, slower, or more transparent, or that remove them or deny their creation entirely.

Designing policy mechanisms in this way gives us our desired property of composability. For example, consider a case in which one policy wishes to set an object's opacity to 50%, and another to 30% (more transparent). As stated, we cannot satisfy both policies at once — the object cannot have both 50% and 30% opacity. However, if we return to the notion that the goal of a policy is to modify attributes to be less intrusive — in this case, more transparent — we can consider these policies as specifying thresholds. That is, the first policy wishes to enforce a *maximum* of 50% opacity, and the second a *maximum* of 30%. Formulated this way, these policies compose: setting the object's opacity to 30% satisfies both policies. Thus, given

some set of thresholds set by different policies, Arya takes the most restrictive intersection (i.e., the attribute values that result in the least intrusive state) and enforces these thresholds on AR objects.

In addition to supporting composable policies, this design also ensures that we can no longer encounter a situation in which policies flip-flop, with one making an object more transparent and the other making the object less. In the above example, the subsequent activation of a third policy specifying a higher maximum opacity (e.g., 60%) would not change the most restrictive active threshold (30%).

This design decision intentionally disallows mechanisms that might result in cyclic policy violations or lead to complex constraint solving, but that may sometimes be desirable (e.g., automatically repositioning AR objects). We discuss possible approaches that future work must explore to support such policies in Section 3.6.

Finally, we note that malicious or buggy policies *can* still result in applications being able to display less content, thus impacting application functionality. However, due to the composable properties of our polices, they cannot, by definition, result in *more* intrusive output. That is, Arya is fail-safe in the face of malicious or buggy policies.

3.3.3.2 Enforcing AR Output Policies

Now that we have determined how policies are specified, we turn our attention to how they are enforced by Arya's output policy module. The algorithms in Figure 3.5 detail policy condition checking and mechanism enforcement at different points within Arya, as we will introduce below.

Although we have thus far discussed policies as though they always apply to all applications and objects, we note that they can be enforced more granularly. For example, policies can be enforced selectively on the objects of specific applications or categories of applications (e.g., entertainment or safety-oriented apps). However, we do not focus on this granularity for the below discussion, instead assuming a more general situation in which policies do apply.

Algorithm 1 Example policy checked on API

```
1: procedure CREATE(AR OBJECT a, AR OBJECT SET 𝒜)
2:     for each On-Create Policy p do
3:         deny ← p.Evaluate(a)
4:         if deny then DenyCreation(a) ; return
5:     Create a ; 𝒜 ← 𝒜 ∪ a
```

Algorithm 2 Per-frame policy enforcement

```
1: procedure UPDATE
2:     Update mapping of real world
3:     for each AR Object a ∈ 𝒜 do
4:         for each Per-Frame Policy p do
5:             p.Evaluate(a)
6:         PolicyModule.EnforceThresholds(a)
7:     ℳ ← Incoming API requests
8:     for each m in ℳ do
9:         ProcessRequest(m)
10:    ℰ ← Pending callback events
11:    for each e in ℰ do
12:        SendEvent(e, targetApp)
13:    finally: Render AR Objects
```

Algorithm 3 Example attribute-modifying API call

```
1: procedure SETALPHA(AR OBJECT a, VALUE alpha)
2:     thresh ← a.AlphaThreshold
3:     if thresh < alpha then alpha ← thresh
4:     a.alphaValue = alpha
```

Figure 3.5: **Policy Enforcement.** These algorithms give pseudocode for how Arya checks and enforces policies (1) on API calls and (2) during the per-frame update loop. The thresholds set when a policy is enforced are respected (3) when object attributes are modified. Policy enforcement is detailed in Section 3.3.3.2.

Design Question: *At what points in its workflow should Arya evaluate policies?* The first natural place to check and enforce policies is when applications attempt to create, move, or modify their AR objects. For example, consider a policy with a condition such as "`if obj.size > X`" and a mechanism such as "`obj.SetAlpha(0.2)`" (i.e., a policy that makes large objects semi-transparent). This policy's condition can be checked, and its mechanism enforced, when the application calls `CreateObject()` or `ResizeObject()`. Similarly, a policy that prevents head-locked objects (in a fixed position of the user's display) can be evaluated and enforced on the call to `CreateObject()`. Algorithm 1 presents example pseudocode for policy evaluation on the `CreateObject()` API call; Arya handles other APIs similarly.

Challenge: Handling Relational Policies. Through our implementation experience with different policies, we find that only checking and enforcing policies on API calls is insufficient when those policies depend on relationships between objects, which may be virtual objects or detected real-world objects. Consider the example of a policy with the condition "if an AR object is occluding a real-world person" and the mechanism "set its opacity to 0.2" — or, in pseudocode, "`if obj.isOccluding(person) then obj.setAlpha(0.2)`". Clearly, this condition could be triggered when an application attempts to create or move its AR objects in a way that obscures a real-world person. However, even without explicit action by an application, changes in the real world (such as a person walking in front of the user) could result in a policy violation.

Now consider a related policy that refers only to virtual objects: "if an AR object is occluding another AR object, set its opacity to 0.2". At first glance, it seems that this policy *can* be enforced on API calls, i.e., when an application creates or moves virtual objects. However, suppose the user changes his or her viewing angle or moves around the physical world. In this case, Arya automatically updates the rendered locations of world-locked virtual objects without explicit API calls from the applications. As a result, objects that were previously not occluding each other may now be violating the policy.

Thus, as these two examples show, Arya needs to be able to enforce policies that de-

pend on relationships between objects *independently of actions taken by applications*. This observation leads to the following design decision:

Design Decision: Check Relational Policy Conditions at Regular Intervals. To account for changes in the real world that may affect policy decisions, such as the user's position and viewing angle, Arya cannot wait for applications to explicitly change their objects. Instead, it must continuously monitor policy conditions that relate to real-world objects (e.g., on a per-frame basis[2]). Thus, on every frame, Arya gathers information from its input recognizers (e.g., to determine if and where there are people in the current view of the real world) and notes the current state of all AR objects. This information is then used to evaluate policies such as the examples above. Once all per-frame policy conditions have been evaluated on an object, Arya enforces the respective policy mechanisms by finding the most restrictive intersection of attribute thresholds and applying them. In the above examples, Arya would set the opacity of the violating object to 0.2. Algorithm 2 details Arya's per-frame policy enforcement workflow. However, we must now consider the following:

Design Question: *How do relational policies that influence specific attributes (e.g., opacity) interact with API calls that modify the same attributes?* For example, consider again the policy which reduces the opacity of AR objects that occlude real-world people to 0.2. What happens if, after this policy is enforced, the application calls `SetAlpha(1.0)` to make that object opaque? If Arya naively evaluates the policy on the current frame before processing the API call, the object will — at least temporarily, for one frame — violate the policy. Such a temporary violation, particularly if the application calls `SetAlpha(1.0)` repeatedly, could nevertheless be disruptive to the user. On the other hand, if Arya processes the API call before enforcing the per-frame policy, it creates additional overhead by needing to roll back the results of the API call.

Design Decision: Decouple Threshold Setting and Enforcing. To avoid both of

[2]Our design considers per-frame checking for relational policies, but it generalizes to other regular intervals. For example, Card et al. [18] suggest that a 100ms interval may be sufficient.

the above problems, we decouple setting a threshold value for an attribute from enforcing that threshold. In the above example, the policy sets an opacity threshold of 0.2 when it is evaluated per-frame. That threshold is immediately enforced, i.e., the object's opacity is reduced. However, to avoid temporary violations, those thresholds are *also* enforced on any API calls processed in the same frame. That is, when Arya handles the `SetAlpha(1.0)` API call, it respects the current opacity threshold for that object, not exceeding 0.2. This process is detailed in Algorithm 3, which shows an example for the `SetAlpha()` API; other attribute-modifying API calls are handled similarly.

3.3.3.3 When Policy Violations Cease

Having considered how policies are specified and how they are enforced, we turn to a final question:

Design Question: *What should Arya do when a previously-enforced policy is no longer violated?* That is, when an AR object that was modified due to a policy ceases to violate said policy, how should those modifications be reverted?

An initially appealing approach is to have Arya itself manage the reversal of policy enforcement. For example, if Arya reduced an AR object's opacity to 0.2 in response to a policy, Arya should also return that object's opacity back to normal when the policy condition is no longer violated (e.g., when the object no longer occludes a real-world person). A benefit of this approach is the loose coupling between AR objects and policies, allowing applications to operate oblivious of any active policies. However, this design raises the following challenge:

Challenge: Policy Impact on Application State. When considering an object attribute, what constitutes a "normal" value is unclear — is it the value of that attribute at the time the policy was first violated? That state may no longer be valid when the policy violation ceases. Is it the application's current expected value of that attribute, supposing it has continued to update what it would be without any policy effects? That may work in many cases, but in other cases, the application may have made different decisions if it had known about the

policy violation. For example, an application whose objects are made transparent due to a policy may wish to remove the objects in question. These considerations illuminate a key tradeoff between application flexibility and more straightforward, policy-oblivious behavior.

Design Decision: Inform Applications About Policies. We choose to inform applications when their objects start or stop violating policies, so they can react appropriately. Under this model, if an app whose object is modified by a policy wishes to, for example, remove that object or display an error message to the user, it can do so. Similarly, this design allows applications flexibility in determining appropriate attribute values after an object stops violating a policy, rather than having Arya revert object attributes oblivious to application semantics.

In choosing to deliver information to apps about when their objects violate policies, we uncover an additional challenge:

Challenge: Privacy Risks. Sharing too much information about policy violations with applications can compromise privacy. Recall that, for privacy reasons (and building on prior work [55]), an application may not have access to a full video feed but rather limited recognizer inputs, e.g., planar surfaces. Now suppose, for example, that when an application's object is made transparent because it overlapped a real-world pedestrian, Arya triggered a callback to the application informing it not only how its AR object was affected but also which policy was violated. While sharing the details of the violated policy could be useful (e.g., allowing the application to move it object to stop violating the policy), it also raises privacy concerns. Specifically, it can reveal information to applications about real-world objects (e.g., that a pedestrian is present) or about other applications' AR objects.

Design Decision: Provide Limited Feedback to Applications. To mitigate this privacy risk, Arya does not share the full details of policy violations with applications. Instead, it informs applications only when attribute thresholds on its objects change (e.g., when an object is made transparent, or when the maximum allowable alpha value increases when a policy is no longer violated), so that it can react appropriately. However, Arya does not

provide any details about the policy condition that triggered the threshold change.

3.3.3.4 Design Summary

In summary, we identified key design questions regarding how to specify AR object policies and avoid conflicts between policies (Section 3.3.3.1), how to enforce policies (Section 3.3.3.2), and what to do when objects cease to violate policies (Section 3.3.3.3). To address these questions and the challenges they raise, we developed an output policy specification framework in which policies consist of restricted, composable conditions and enforcement mechanisms, with privacy-conscious feedback to applications when violations occur or cease.

We consider the design questions and challenges that we uncovered through this process to be contributions in and of themselves. While our proposed solutions meet our security goals, future AR system designers may wish to make different design choices. Our work surfaces a number of challenges and tradeoffs that must be considered, which we hope will help guide potential alternate design paths.

3.4 Implementation

We now describe our prototype implementation of Arya. Developing our prototype gives us the opportunity to deeply explore and evaluate Arya's AR output policy module, and iteratively feeds back into our design process. Our prototype consists of several parts: an AR simulator and virtual scenes to represent the real world, the Arya core implementation (including the output policy module and infrastructure to support multiple applications), standalone applications that run on Arya, and AR output policies that are enforced by Arya. We detail these components in turn.

AR Simulator. In practice, a full-fledged AR system has many moving parts — crucially, it continuously senses and processes real-world input, which feeds into applications as well as, in our design, the output policy module itself. However, real-world input is by its nature noisy and variable, as we discuss in Section 3.6. Even if we had perfect sensor hardware

Identifier	Conditions	Mechanisms
P1	If an AR object's speed exceeds X	Set the object's speed to X
P2	If an AR object is within X feet of the user	Set the object's alpha value to 0
P3	If an AR object occupies more than X percent of the display	Set the object's alpha value to 0
P4	If an application attempts to create a head-locked object	Deny the creation request
P5	If a user's vehicle is in motion	Set the alpha value of certain AR objects to 0
P6	If an AR object is occluding pedestrians or road signs	Set the object's alpha value to 0
P7	If an AR object is occluding exit signs	Set the object's alpha value to 0
P8	If an AR object's alpha value is less than X	Disable user interactions with the object
P9	If an AR object is not bounded by a real-world billboard	Set the object's alpha value to 0
P10	If an AR object is occluding another application's AR object	Set the object's alpha value to 0

Table 3.2: **Implemented Policies.** This table details the conditions under which our prototype policies are violated and the mechanisms Arya uses to enforce them. This list matches the policies in Table 3.1. X represents a parameterized value specified by individual policies. We note that policies may be selectively applied to specific applications or groups of applications — for example, P9 may only apply to an advertising app.

and sensor data processing algorithms, we would still like to evaluate in controlled, possibly hard-to-stage scenarios (e.g., while driving).

Since the focus of our work is not on improving or evaluating AR input processing (a topic of other research efforts, e.g., [31, 67, 76]), and to support controlled experiments, we abstract away the input handling part of Arya for our prototype. Instead, we create an *AR simulator*, which consists of a virtual reality (VR) backend to represent the real world. This approach is similar to driving simulators commonly used in other research, e.g., [113].

Specifically, rather than outfitting our prototype with real hardware sensors, we build on the Unity game engine, using Unity virtual environments, or "scenes", to represent the real world. This technique allows us to isolate the output management portion of the system and reliably "detect" our simulated real-world objects. AR applications running on Arya can create virtual objects to place into these scenes, and Arya's output policy module can regulate those objects given information about the fully-specified underlying VR world.

Virtual Scenes Representing the Physical World. A benefit of our AR simulator

is that it easily allows us to test output policies in different Unity scenes that represent various real-world scenarios. Specifically, we developed three scenes to represent HMD and automotive scenarios: an "in-home" scene,[3] an "AR-windshield" scene, and an "office" scene. These scenes are shown in Figure 3.6; the bare scenes, without AR applications running, are shown in the left column of that figure. We emphasize that these scenes represent the real world, and that no virtual content created by AR applications is shown in the bare scenes.

Arya Core. Up to this point, we have described only our prototyping infrastructure for representing a model of the real world. We now turn to Arya itself. We build Arya's core also on top of Unity, written in 3767 lines of C# code[4]. Loading this core into a new scene requires only a few user interface actions within the Unity editor. While Arya interfaces with our virtual scenes, it is largely modularized.

The Arya core includes infrastructure for running multiple AR applications on top of it, including handling multiple application threads and managing communication over local sockets. Arya exposes APIs to those applications for querying the real-world scene as well as for creating and modifying AR objects (such as `Object.Move()` and `CreateObject()`).

We implement recognizers in our prototype by labeling specific "real-world" objects in our virtual scenes as objects of interest, e.g., people, billboards, and signs. This information about the real world, as well as the state Arya keeps about applications' AR objects created and modified through its APIs, feeds into Arya's output policy module. This module enforces policies on application output, as detailed in Section 3.3.3.2.

Application Interface. Our prototype supports multiple standalone applications running atop the Arya core, which can simultaneously create and interact with AR objects and augment the same "real-world" scene. Applications are isolated by running as separate OS processes, such that their only interaction is implicitly by augmenting the same "reality."

---

[3]We augmented a pre-built scene, "Brian's House", purchased from the Unity Asset Store: `https://www.assetstore.unity3d.com/en/#!/content/44784`

[4]We used the CLOC tool for calculating lines of code: `https://github.com/AlDanial/cloc/releases/tag/v1.70`

Arya applications are written in C# and extend our base class `ARApplication`. This base class contains 889 lines of C# code and provides the infrastructure for communicating with the Arya core over local sockets to make API calls (e.g., to create or modify objects). We describe case study applications that we implemented for our evaluation in Section 3.5.

Prototype Policies. Finally, we prototype an AR output policy framework. Policies are written as standalone C# modules that extend our `ARPolicy` base class and are programmatically instantiated by the Arya core. As described in Section 3.3, policies follow a well-defined structure consisting of a condition and a mechanism. The Arya core provides a fixed set of AR object attributes (used in conditions) and enforcement mechanisms that policies can employ. Table 3.2 details the specific case study policies we implemented. We stress that the conditions and mechanisms we chose to implement are not the only possible options that Arya can support. Additional attributes could be defined, as could additional mechanisms that meet our composability criteria (moving objects towards "less intrusive" states). For example, our most complex attribute (determining if one AR object occludes another object) consists of only 49 lines of code, suggesting that developing new attributes could be easily done.

3.5 Evaluation

Our evaluation goals are two-fold. First, we seek to evaluate Arya's ability to support and enforce a variety of policies from different sources. Second, since policy enforcement is on the critical path for rendering output, we measure the performance overhead introduced by our prototype's output policy module. Our results suggest that Arya is a promising approach for constraining AR output — not only does it successfully address, for the first time, many output security issues, but it also does so with reasonable performance. We use these results to surface additional lessons and recommendations for future AR platform developers.

Figure 3.6: **Case Studies.** These screenshots show our case study scenarios: HMD in the home (top), car windshield (center), and HMD in the office (bottom). The left column shows the bare scenes in our Unity-based AR simulator, representing the real world without any apps running. From our prototype's perspective, everything in the bare scene is part of the real world. The center column shows our case study apps running, exhibiting both desirable and undesirable AR output behaviors. The right column shows the result of policy enforcement, leaving only desirable AR output. Note that Unity's alpha adjustment mechanism leaves transparency artifacts to outline where violating AR objects would be.

3.5.1 Case Studies: Policy Expressiveness and Effectiveness

We evaluate the efficacy of Arya's output policy module through case study applications that run within our three virtual scenes, described in Section 3.4: a home, a driving scene, and an office. We design our case study applications to exhibit both (a) acceptable or desirable behaviors, as well as (b) behaviors that violate one or more of our prototype policies detailed in Table 3.2. Figure 3.6 shows screenshots of our applications running in these scenes both without (center column) and with (right column) policy enforcement active. The left column shows the bare scenes, with no applications running.

Case-Study Applications. We developed two applications per scene that test our various policies. Our focus is to exercise our output policies, and thus we did not implement complex application-level logic. Nevertheless, these applications are inspired by real applications that might (or already do) exist for these emerging platforms.

HMD in the Home. For the home scene (top row of Figure 3.6), we created a "Virtual Pet" app, which displays a world-locked virtual cat that can move independently in the user's environment. However, the application moves the cat at a distractingly fast speed through the user's view, and it displays a head-locked spider that the user cannot look away from. Additionally, we built a tabletop game[5] in which the user increases their score by hitting coins with a ball. However, the application pops up in-game purchase notifications that block the output of other applications and may annoy the user.

AR Windshields. For the driving scene (center row of Figure 3.6), we created an advertising application that displays targeted ads over real-world blank billboards. However, the application also displays ads throughout the rest of the user's view, potentially creating a driving hazard. Additionally, we implemented a "notification" application that displays dummy text message, calendar, and email alerts. Unfortunately, it continues to generate distracting alerts while the car is in motion.

HMD in the Workplace. For the office scene (bottom row of Figure 3.6), we imagine a group of engineers using AR to design a new automobile.[6] We built an application that allows users to view their car models from different angles simultaneously. Additionally, we created an application that displays information to users about their colleagues, such as their names and roles in the company. While both of these applications do not exhibit intentionally malicious behavior, their outputs sometimes obscure the user's view by taking up too much of the screen, appearing too close to the user's face, or blocking out important information in the real world such as exit signs.

---

[5]Inspired by `https://unity3d.com/learn/tutorials/projects/roll-ball-tutorial`.

[6]Inspired by an application for HoloLens: `https://www.youtube.com/watch?v=yADhOKEbZ5Q`.

Security Discussion. As illustrated in Figure 3.6, Arya successfully allows multiple case study applications to concurrently display content while simultaneously enforcing our prototype policies to prevent malicious or undesirable output behaviors. Specifically, referring to policies by their identifiers in Table 3.2:

- In the home scene, P4 prevents the head-locked spider from being created. Additionally, P10 prevents the in-app purchase dialog from occluding the cat (a virtual object from another application), and P1 prevents the cat from moving too fast.
- In the driving scene, P6 prevents virtual ads from obscuring real-world pedestrians, and P9 constrains them to appearing *only* over real-world billboards. P5 prevents notifications from popping up while the car is in motion.
- In the office scene, P7 prevents the modeling application from blocking real-world exit signs. Meanwhile, P2 and P3 make objects that get too close to the user or take up too much space partially transparent.

These case studies exercise all but one of the policies we implemented (Table 3.2). The exception is P8, which disables user input on obscured AR objects. Though we implemented this policy, we cannot exercise it, because our prototype is designed to focus on generating output and hence lacks meaningful user input for application interactions.

Through these case studies, we confirm the ability of our policy framework to support policies that constrain a range of behaviors in different contexts. Our case studies also highlight, for completeness, an output safety risk that our current policies cannot mitigate: risks with unsafe or frightening *content*, such as spiders. Our policies — just like conventional web browsers, desktops, and mobile devices — do not prevent applications from displaying specific undesirable objects. This issue presents a potential avenue for future work.

3.5.2 Performance Evaluation

Arya's output policy module directly mediates content that applications wish to display and thus lies on the critical path for rendering. As such, the output policy module should incur

minimal overhead. While our prototype implementation is not optimized or representative of a full-fledged AR system, analyzing its performance can nevertheless shed light on possible output bottlenecks and other considerations that must go into implementing an output policy module in a production system.

Our case-study applications successfully exercise our prototype policies, but they contain relatively few AR objects. To identify potential bottlenecks, we next analyze the performance of the output policy module under heavier workloads, i.e., when there are many objects present. We first profile the performance of our output policy module in the absence of our application communication infrastructure to isolate the performance impact of our policies. We then analyze our communication infrastructure and conduct a full-system evaluation.

3.5.2.1 Profiling the Output Policy Module

We begin by profiling our prototype's output policy module without the overhead of application communication. To isolate the impact of the output policy module, we create a simple evaluation scene containing several objects (a "person", a "billboard", and an "exit sign"). Rather than having a separate application process create and update AR objects, we instead programmatically trigger API calls directly in Arya's core on a per-frame basis. From the output policy module's perspective, these requests appear to come from an actual application. This setup simulates application behaviors but eliminates any performance impact of the communication infrastructure and allows us to focus on the output policy module itself. This methodology also allows us to ensure the same amount of work occurs each frame, enabling repeatable experiments.

Our primary performance metric for profiling the output policy module is the frame rate, or average frames-per-second (FPS), of Arya's Unity backend. Since Arya's core functions (handling API calls and enforcing policies) operate on a per-frame basis, extra overhead introduced by the output policy module directly decreases the frame rate, making FPS a meaningful metric. For each data point in our measurements, we calculated the average FPS over a 30 second interval (after an initial 10 second warm-up period), repeating each

trial 5 times. We conduct two evaluations with this experimental setup: first, we compare the individual performance of the policies we implemented, and then we investigate policy performance as we scale the number of virtual objects in the scene.

Individual Policy Performance. We begin by trying to understand the performance impact of our individual policies relative to a baseline scene without any policy enforcement. These results are shown in Tables 3.3 and 3.4.

In designing this experiment, our goal is to fully tax the system, such that differences between policies become visible. To do so, we simulate the following application behaviors: we create N overlapping objects directly in front of the user, and move each object a small amount every frame. For these experiments, we chose N objects such that the baseline would be under load — i.e., less than 60 FPS, which is considered a standard for smooth gameplay in many PC video games [42] — so that we could see the effects of policies. We experimentally determined that $N = 500$ objects would give us a baseline frame rate of less than 60 FPS.

We designed the scene such that every frame, each virtual object violates each policy we implemented (see Table 3.2), though we only activate and evaluate one policy at a time. Two of our policies required slightly different experimental setups to trigger violations: P4 requires that the baseline setup repeatedly attempt to create objects each frame, and P9 requires the baseline setup to contain objects that are locked to real-world objects (in this case, billboards). The results for these two policies are in Table 3.4, and the caption further details the specific experimental setups.

Tables 3.3 and 3.4 show the results of these experiments. We observe a range of performance impacts across our different policies. For example, P1 (which limits the speed at which objects can move) and P2 (which makes objects too close to the user transparent) incur virtually no additional overhead over the baseline. On the other hand, P10 (which makes virtual objects that obscure other virtual objects transparent) incurs an almost 20 FPS hit.

	Baseline	**P1**	**P2**	**P3**	**P5**	**P6**	**P7**	**P10**
Avg FPS	51.4	51.3	48.0	39.2	49.0	43.7	43.8	32.3
Std Dev	1.2	1.3	1.1	1.5	0.4	1.6	1.1	1.8

Table 3.3: **Profiling Policy Performance (1).** As described in Section 3.5.2.1, we calculate the average frame rate of the Arya core with different active policies, compared to a baseline with no active policies. Policy identifiers in this table match those in Tables 3.1 and 3.2. In our experimental scenes, we load the system by having 500 objects that each move once per frame, and each tested policy is violated on every frame. Results are averaged over five 30-second trials.

	Baseline	**P4**
Avg FPS	4.6	57.7
StdDev	1.0	2.0

	Baseline	**P9**
Avg FPS	32.6	30.7
StdDev	1.0	1.2

Table 3.4: **Profiling Policy Performance (2).** For two policies, we use a different experimental setup, with different baseline measurements, than used in Table 3.3. For P4, which acts on the `CreateObject()` API, we create and delete objects every frame rather than moving them. For P9, we create virtual objects locked to a real-world billboard. Since the object-locking functionality itself incurs overhead (independently of policies), we generate a separate baseline. As in Table 3.3, results are averaged over five 30-second trials.

A key observation is that *the complexity of object attributes directly influences policy performance*. For example, P1 simply sets a threshold on objects' movement speeds, which is easily checked and enforced when an application calls `object.Move()` with a speed parameter. On the other hand, P10 incurs more overhead because it must detect virtual objects that occlude others in every frame, requiring costly raycasting operations. This lesson suggests that optimizing attribute computations and intelligently caching information will be critical for such a scheme to work in practice.

This lesson is further supported by our experience applying preliminary optimizations to P10. Initially, P10 incurred significant overhead due to redundant raycasting operations

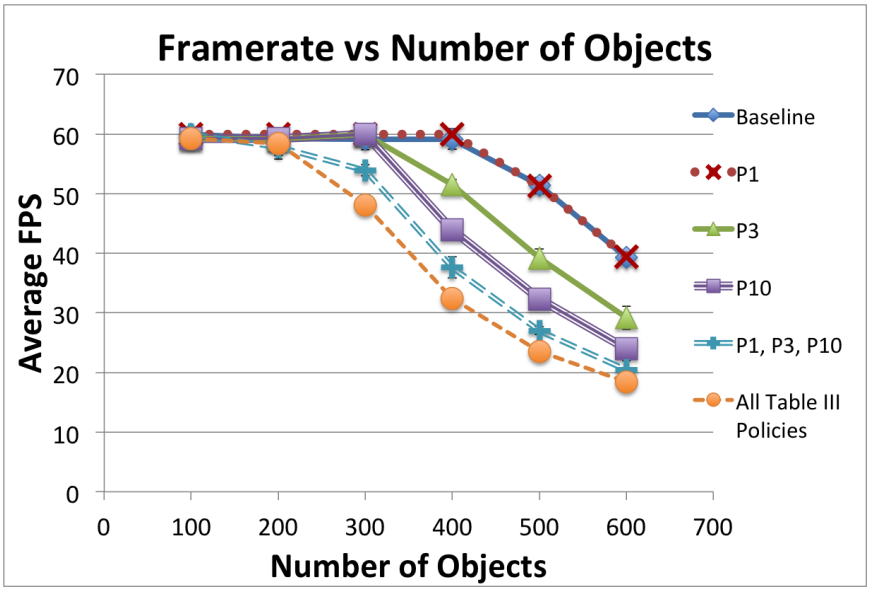


Figure 3.7: **Performance with Multiple Policies and Scaling AR Objects.** We investigate the performance impact of combining multiple policies and how that impact scales with increasing numbers of AR objects in the scene. We find that the performance overhead of multiple policies is less than the sum of the overhead from those policies individually, and that the performance hit of adding AR objects (unrelated to policies) dominates the impact of policy enforcement.

between overlapping objects, resulting in an average frame rate under 2 FPS. However, by optimizing P10 to not repeat computation on AR objects that the policy has already acted upon, we significantly improved its performance. This suggests that pursuing policy optimizations can have a great impact.

Finally, we note that P4, a policy that denies certain `OnCreate()` calls, actually *improved* performance over the baseline. This is a result of the baseline scene repeatedly creating and deleting headlocked AR objects, in contrast to P4 simply denying the requests. Thus, we observe that policies that deny object creation could also be used as a denial-of-service protection against applications attempting to create many objects.

Policy Performance Scaling with AR Objects. The above benchmark provides a single snapshot of how our policies compare, with a fixed number of virtual objects (500). However, we also wish to understand (1) how policy performance scales as the number

of active AR objects that violate them increases, and (2) how performance is affected by multiple simultaneously running policies.

Using the same experimental setup from Table 3.3, we compare the baseline scene to several policies, as well as combinations of policies, as we vary the number of active AR objects present. We select the policies for this experiment based on the results in Table 3.3, choosing our best performing policy (P1) and two worst-performing policies (P3 and P10). Figure 3.7 shows the results of this experiment. Note that we cap the maximum FPS at 60 using Unity's `Application.targetFrameRate` feature.

Our results reveal several interesting lessons. First, *policy overhead is not additive*. The performance hit incurred by several policies combined, even those that leverage different attributes, is less than the sum of their overheads individually. This finding is promising, since in practice, multiple policies may indeed be active at once. Even if the list of policies increases, we expect overlapping work between policies. For example, the cost of loading objects in memory could be amortized across multiple policies, and multiple policies may require similar computations about objects.

Second, we observe that *the performance impact of additional virtual objects dominates the impact of policies*. That is, as the number of AR objects increases, the frame rate of the baseline with no policies drops below 60 FPS, scaling with the number of objects. Although the frame rate with multiple active policies drops below 60 FPS more quickly, the impact of multiple policies scales with number of AR objects similarly to the baseline, after the initial performance hit of activating any policies. This is perhaps not surprising: more complex applications will run more slowly. However, the fact that the performance impact of policy enforcement does not become increasingly worse with more AR objects is promising.

3.5.2.2 Full System Evaluation

Our above experiments isolate the performance impact of the output policy module and evaluate it with respect to varying numbers of AR objects and policies. However, we also wish to understand the impact of the output policy module in the face of multiple pro-

	1 App	2 Apps	3 Apps	4 Apps
Avg Msgs/App/Second	1808	1020	646	508
Std Dev	221	115	251	141

Table 3.5: **Arya Message Throughput.** To inform our choice of parameters for a full system evaluation (shown in Figure 3.8), we first characterize the performance of our unoptimized application communication infrastructure, by which applications use local sockets to communicate with the Arya core to make API calls. The results are averaged over five 30-second trials.

totype applications simultaneously running on Arya. Since our primary focus was on the output policy module, other elements of the system — specifically, its handling of multiple application threads and local socket communications — are unoptimized. To isolate the performance impacts of these unoptimized components, we first conduct a microbenchmark evaluation to profile Arya's application communication infrastructure. Using the results of this microbenchmark, we choose parameters for a meaningful full system evaluation such that we do not hit bottlenecks due to communication and accidentally mask the impact of the output policy module.

Communication Microbenchmark. We first measure the throughput of Arya's message processing infrastructure. We connect application processes to Arya over local sockets, after which the applications saturate the connections with messages, which Arya then processes as fast as it can. Table 3.5 summarizes the message throughput of Arya with increasing numbers of concurrently running applications, where one message corresponds to one API call. As we increase the number of applications, the number of messages Arya can process per application decreases. This result is expected, since each application runs as a separate process, and communication between Arya and each app run on separate threads.

Putting It All Together. Finally, we evaluate our full prototype. We compare the average FPS under workloads with different numbers of applications communicating over sockets, and with many active policies. As before, we designed a scene in which there are multiple virtual

objects, each moving once per frame, and we calculate the average FPS over a 30 second interval.

We use the results of our socket microbenchmark to determine a realistic workload — i.e., a total number of AR objects — that will avoid communication bottlenecks. We fix the total number of AR objects for this experiment at 48, evenly split across the number of running applications (1-4). Each application calls the `object.Move()` API on each of its objects approximately 30 times per second. We arrive at 48 objects based on the results from Table 3.5: Arya can support up to about 1800 messages per second, and $48 \times 30 < 1800$, and it is evenly divided by 1, 2, 3, and 4 (number of apps we test). While 48 objects is much less than the 500 we used in our profiling experiments above, those experiments were specifically designed to tax Arya, whereas 48 represents a more reasonable workload for applications. For example, our case study apps consisted of only a handful of objects each. Additionally, in practice, apps may not call APIs on each of their objects continuously, though we do so in our experiments.

We compared this workload, with all seven policies from Table 3.3 active and continuously violated, to the baseline. Our results are shown in Figure 3.8. The error bars represent the standard deviation of 5 trials. The result is promising: we find that *under this realistic, 48-object workload, the performance impact of policy enforcement is negligible over the baseline*. Whereas our earlier profiling of the output policy module highlights bottlenecks (e.g., attributes that are expensive to compute) under load, our full system evaluation suggests that even our unoptimized prototype can handle multiple applications and multiple policies under a realistic workload.

3.6 Discussion

Designing a full-fledged operating system for AR platforms that supports strong security, privacy, and safety properties while enabling rich application functionality is challenging. Prior work addresses many input privacy challenges for AR, and in this work, we make significant strides towards securely handling visual output. However, many challenges remain.

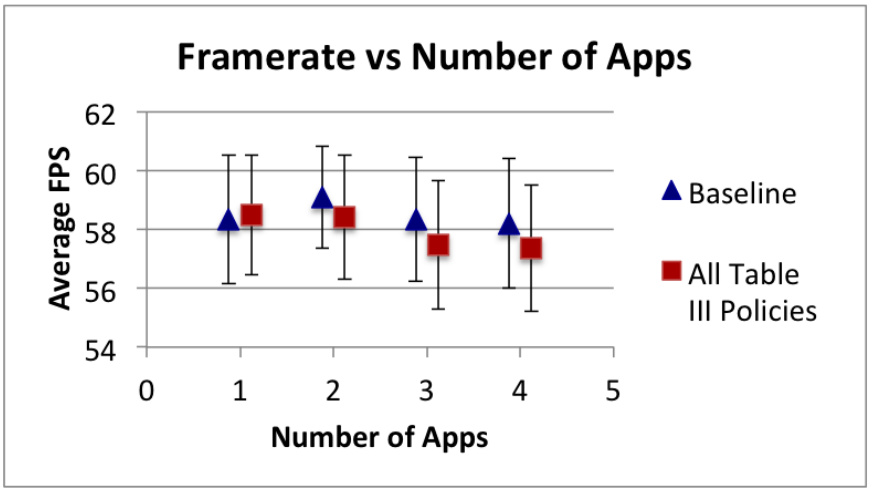


Figure 3.8: **Full System Evaluation.** This graph shows the results, in terms of Arya's core frame rate, of running 1-4 applications with 7 active policies, compared to a baseline with no active policies. As described in Section 3.5.2.2, the total number of objects is fixed at 48, split evenly across the number of applications in a given trial. Note that this graph's y-axis does not start at 0, so that the small differences in performance are visible. We find that under this reasonable workload, the performance impact of policy enforcement is minimal.

We step back and reflect on these challenges, and we make recommendations for designing future secure AR systems.

Handling Noisy Input Sensing. While our prototype used simulated AR environments to enable controlled output-related experiments, real AR systems will need to handle potentially noisy sensor inputs. Input noise may confound output policy management (e.g., if a recognizer fails to detect a person). Thus, future work must explore how to mitigate risks from noisy input — e.g., considering how to deal with ambiguity and probabilities, and how to determine appropriate defaults. For example, recognizers may need to output confidence values — e.g., confidence that there is a person in the video feed — and the output policy module may need to use confidence values across multiple frames to make determinations.

Constraint-Solving Policy Framework. By supporting policy mechanisms that compose by design, Arya avoids challenges raised by potentially conflicting or flip-flopping policies. However, this design choice excludes some policy mechanisms, particularly those that move

AR objects (since they might move objects to locations where they violate other policies). Some systems may wish to support such policies: for example, automatically repositioning a safety dialog on an AR windshield to ensure that it remains visible but does not obscure pedestrians. Future work should consider whether it is possible to design a more complex policy framework that supports policies that may conflict. One approach may be to allow applications to express AR object attributes as constraints rather than fixed values (e.g., specifying several acceptable locations where an AR objects may be displayed), giving the output policy module the responsibility of solving those constraints in the face of all active policies. However, such a system would still need to answer the question of what to do when a given set of constraints cannot be solved. Prior work has considered similar constraint-solving approaches for laying out UIs in more traditional platforms (e.g., tablets or phones) [41]. Techniques from this work may be applicable here, though the AR context also raises new challenges (e.g., the potential for constant constraint solving due to rapid changes in the real world).

Application Prioritization. With many applications potentially competing to display output that is subject to a variety of policies, we argue that Arya could benefit from a prioritization scheme that favors certain applications over others. While not the focus of this study, we observe, for example, that a safety-critical application might receive priority over a game if their outputs conflict or if the user encounters a dangerous situation.

API Extensibility. Arya hides low-level data from untrusted applications, providing high-level abstractions for applications to receive input (recognizers [55]) and to display output (AR objects). While this model effectively restricts the capabilities of malicious or buggy applications, it may also present flexibility challenges for honest applications (similar to the input flexibility challenges faced in [55]). A key question is thus how Arya should expose mechanisms for adding additional functionality without compromising the security of the system. While also not the focus of our study, we observe that an extensibility model analogous to OS device drivers, with modules developed by reputable third parties, could

facilitate more flexible options for application developers.

Non-Visual AR Output. Arya focuses on managing visual output, but as AR systems continue to evolve, we will likely see increased richness in non-visual output, such as auditory or haptic. Thus, future work should explore how the design choices and lessons presented in this chapter can be applied to other types of AR output. We expect that some challenges and design choices will be similar (e.g., a condition/mechanism-based policy framework) while others will differ. For example, beyond blocking certain audio output entirely, are there other, less strict mechanisms that may be viable (similar to partial transparency of visual content)?

Low-Level Support for AR Objects. Arya relies on the AR object abstraction, by which an application's visual output consists of multiple non-rectangular regions of pixels, rather than a single rectangular window. The traditional window abstraction is deeply embedded in today's operating systems and their interactions with graphics and display hardware. In our prototype, these issues were below the abstraction level of our implementation, which was built atop the Unity game engine. However, future work — and certainly non-prototype AR systems interfacing more directly with hardware — will need to consider how the AR object abstraction can and/or should be incorporated into lower-level design choices.

Robust Multi-Application Conflict Mediation. Finally, although Arya provides output mediation capabilities for conflicts between multiple apps, it does so in a relatively rigid fashion. The runtime policy method of conflict mediation that Arya employs may prove sufficient for multi-application workloads, but we require a deeper understanding of its trade-offs, as well as what the alternative approaches for handling conflicts between apps may be. This dissertation considers the question of multi-application conflict mediation in significantly greater depth in Chapter 5.

3.7 Conclusions

Immersive augmented reality technologies, such as head-mounted displays like Microsoft's HoloLens or automotive windshields, are becoming a commercial reality. Though the computer security research community has begun to address input-related risks with emerging AR platforms, little has been done to address *output* security challenges. Modifying the user's view of the world is a key feature of AR applications, and left unconstrained, this ability can raise serious risks. Our work considers these risks — for example, buggy or malicious applications that create virtual content that obscures the user's view of the real world in undesirable or unsafe ways.

To address these risks, we design, implement, and evaluate Arya, an AR platform that supports multiple applications simultaneously augmenting the user's view of the world. Arya's primary contribution is the design of an *output policy module* that constrains AR application output according to policies (e.g., preventing virtual content from obscuring a real-world person). We identify and overcome numerous challenges in designing an AR output policy specification and evaluation framework that supports composable, effective, and efficient policies. We evaluate our prototype implementation of Arya with prototype policies drawn from various sources. We find that Arya prevents undesirable behavior in case study applications, and that the performance overhead of policy enforcement is acceptable even in our unoptimized prototype. The design challenges we raise in this work, and the solutions we propose through Arya, represent a promising step towards secure AR output.

Chapter 4

TOWARDS SECURITY AND PRIVACY FOR MULTI-USER AUGMENTED REALITY: FOUNDATIONS WITH END USERS

When my collaborators and I began the Arya project described in Chapter 3, immersive AR platforms were not yet available for commercial use. However, by the time we completed that project, devices such as the HoloLens and Meta 2 had been released. For the first time, this new generation of AR was being put in the hands of real users. These developments presented us with a unique opportunity to explore how users *themselves* perceive and interact with this new generation of immersive AR technologies, while these technologies were still young and not widely deployed. Specifically, this chapter aims to develop a better understanding of the security and privacy concerns that users have around emerging AR technologies, in the context of both individual experiences and experiences shared between multiple users. This chapter presents a qualitative user study that my collaborators and I conducted to develop such an understanding, and this work originally appeared in the 39^{th} IEEE Symposium on Security and Privacy [65].

4.1 Motivation and Overview

As this dissertation previously discussed, prior efforts within the computer security and privacy community have made significant progress towards anticipating and addressing security and privacy challenges raised by AR technologies [25, 27, 95]. For example, these works have sought to defend against buggy or malicious apps on a user's device that may record privacy-sensitive information from the user's surroundings [36, 55, 91, 108, 117] or disrupt the user's view of the world (e.g., by occluding oncoming vehicles or pedestrians in the road) [61, 63], as well as the risks that a user's AR device might pose to bystanders [29, 96].

While valuable for the problems that they do tackle, we observe two critical gaps in prior works. First, they consider primarily *individual* AR users and their devices. However, emerging AR technologies will not be used only by individual users in isolation, but also by multiple users, each with their own AR device — including users who share the same physical space and may interact with shared virtual content embedded in this space. Indeed, existing AR research efforts (e.g., [59, 107, 118]), as well as already deployed AR apps such as Pokémon Go [81], rely on interactions between multiple, often physically co-located, users. We refer to AR systems that support these interactions as *multi-user AR systems*, and we argue that considering the risks that might arise for users of such systems is critical to the success of future AR technologies. Precursors of such risks have already begun to appear in the wild today, e.g., recent "vandalism" of augmented reality art in Snapchat [70].

Second, we observe that immersive AR technologies such as Microsoft's HoloLens [50] have only recently become available. Thus, even in the context of individual users or AR devices, prior works have focused on *conjectured* security, privacy, and safety concerns that arise in anticipation of emerging AR technologies, but that are not necessarily grounded in users' experiences with the technologies themselves.

Our Goals and Approach. We aim to bridge the above gaps by investigating the concerns of end users grounded in their experiences with real AR technologies, in both single- *and* multi-user contexts. That is, we strive to uncover a broad spectrum of risks that AR users may face — which may stem from buggy or malicious apps *or* other misbehaving users — and to identify challenges that must be addressed to support rich single- and multi-user experiences. Since immersive AR systems are only just emerging, we cannot fully predict users' expectations of or interactions with these technologies, nor their interpersonal interactions while using them. Thus, we directly study end users engaging with real AR technology, and with each other, through an in-lab partner study using the Microsoft HoloLens, an immersive AR headset (see Figure 4.1). In studying multi-user AR, we also focus on physically co-located users, rather than remote AR interactions such as telepresence. While we return

to a discussion of remote interactions in Section 4.4, we observe that physically co-located interactions exercise a fundamentally unique property of AR, compared to traditional digital interactions: the ability to support simultaneous views of shared physical *and* virtual worlds. Ultimately, we strive to provide a broad foundation for understanding and addressing the computer security and privacy challenges that emerging AR technologies will present.

Research Questions. In support of our above goals, we designed our study to investigate the following research questions:

1. *RQ1:* What expectations and behaviors arise for users engaging with a real, immersive AR technology, and what interpersonal interactions arise *between* these users?
2. *RQ2:* What concerns arise for users in practice — involving both single- and multi-user experiences — given the opportunity to interact with other users and applications on an immersive AR device?

Finally, since prior work has considered technical challenges with security primarily for single-user AR systems, we ask:

3. *RQ3:* What new system design challenges and opportunities arise for security and privacy in *multi-user* AR?

Methodology and Findings Highlights. We conducted an in-lab, qualitative user study with the HoloLens. We recruited pairs of participants (22 individuals in 11 pairs), combining hands-on HoloLens activities with semi-structured interview questions. Following accepted methods for qualitative research [20, 43, 46], we focused in depth on a small number of participants until we reached saturation of themes.

Among other findings detailed in Section 4.3, we find (to our surprise) that the HoloLens, despite its technical limitations, provided an immersive experience that shaped participants' expectations of and interactions with virtual content (Section 4.3.1). Notably, participants often assumed that virtual objects behave like physical objects — for instance, instinctively stepping around virtual objects or assuming (sometimes incorrectly) that both they and their physically co-located partner could see the same virtual objects. As we discuss, such

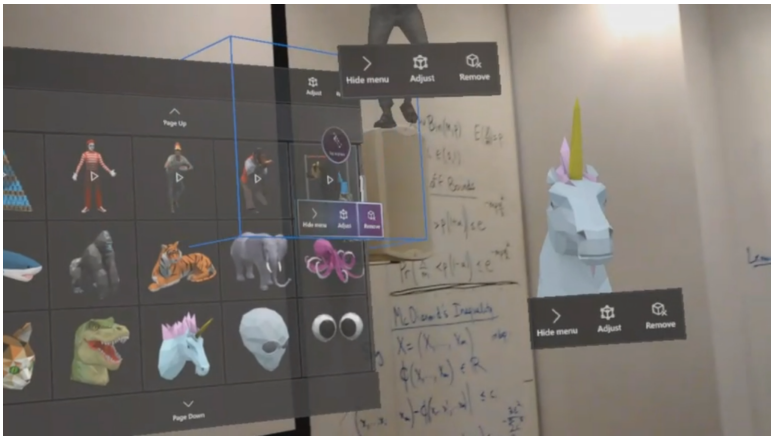


Figure 4.1: **Holograms.** A first-person view of virtual objects, or "holograms", as seen through the HoloLens head mounted display, including 2D menus and 3D objects.

expectations can be leveraged adversarially. Further, participants' interpersonal interactions (Section 4.3.2) — though lighthearted in the context of the study — hinted at potential conflicts and challenges. For example, some participants placed virtual objects in each others' faces or attempted to steal control of objects from each other.

Once participants had the opportunity to experience immersive AR technology firsthand, we asked them to consider specific adversarial scenarios, involving both other users and untrusted applications. In response, participants raised a rich variety of concerns about risks that might arise from these scenarios in both single- and multi-user contexts (Section 4.3.3). These concerns both corroborate and enrich those considered in prior work (e.g., the risk of deceiving someone about the physical world) and raise new issues around interpersonal interactions (e.g., concerns about other AR users destroying or manipulating one's virtual objects).

Finally, whereas prior technical work focused on securing single-user AR experiences, our results raise new design challenges for securely supporting multi-user AR interactions. For example, participants' interactions highlighted tensions around ownership and access control of virtual objects (Section 4.3.4).

Contributions. In summary, we contribute the following:

1. *Problem Identification:* We identify the fundamental need — largely unaddressed in prior work — to consider security, privacy, and safety for emerging single- *and* multi-user AR technologies, grounded in the experiences and interactions of end users.
2. *Study of End Users with Real AR Technology:* Through a user study with pairs of participants using the HoloLens, we identify and investigate critical research questions in support of the above goal.
3. *Foundation for Secure AR Systems:* Our work provides a foundation for addressing the security, privacy, and safety risks that will imminently arise for both single- and multi-user AR scenarios, and we raise key research and design challenges to inform future defensive directions.

4.2 Methodology

4.2.1 Methodology Overview

We designed a user study to investigate our research questions described in Section 4.1, in support of our above goals. Before presenting the full details of our methodology, we highlight several key decisions we made in designing our study.

Qualitative, In-Lab Partner Study. Since this research space remains largely unexplored, we designed an *exploratory, qualitative* study. Compared to a quantitative methodology, a qualitative study allowed us to explore a broad spectrum of expectations, interactions, and concerns, with limited need for preconceived notions of what we might find. Furthermore, immersive AR devices are not yet widely deployed amongst consumers, so we conducted our study *in-lab*. We brought in participants to use the Microsoft HoloLens, one of the most sophisticated, immersive AR devices commercially available today; we provide further details on it in Section 4.2.3.

Given our goal of studying multi-user AR systems, we conducted our study with *pairs of participants*. In an effort to ensure that participants felt comfortable enough with each other

to explore, converse, and potentially push boundaries while interacting during the study, we recruited pairs with pre-existing relationships. Additionally, because we hoped to observe participants' natural expectations and behaviors before they were shaped by the actual affordances of the HoloLens, we sought participants with *no prior HoloLens experience*.

Two Study Phases: HoloLens Activities and Interviews. We divided our study into two main phases: an activity phase in which we observed participants interacting with several HoloLens apps, and a semi-structured interview phase.

The activity-based phase allowed us to observe participants in real time as they interacted with applications and each other, thereby organically surfacing their expectations, reactions, and potential conflicts. We carefully selected HoloLens apps (and in one case, created one ourselves) that would provide participants with both single- and multi-user AR experiences — we detail the specific apps we used in Section 4.2.4 below.

By providing participants with hands-on HoloLens experiences, we sought to enable them to think more concretely about their potential concerns of immersive AR technologies in both single- and multi-user contexts. We designed the second, interview-based phase of the study to surface these concerns. Though we found that our partner study design naturally encouraged participants to think adversarially, we did not prime them to consider any specific threats. Rather, we asked open-ended questions about their potential concerns in AR scenarios involving different stakeholders (including other AR users, apps installed on their devices, and bystanders).

4.2.2 Recruitment, Screening, and Ethics

We recruited participants by advertising our study on mailing lists, on a local neighborhood Facebook group, and by asking personal contacts to forward our study information to additional mailing lists. Candidates completed our screening survey indicating any AR devices they had used, demographics (age, gender, profession) and contact information (name, email address), and their relationship with their potential partner (e.g., friends, co-workers,

spouses). We selected pairs who reported no prior experience using the HoloLens or similar AR devices. Participants who completed the interview were each compensated with a $15 Amazon gift card.

This study was approved by our University's IRB. We did not ask participants to reveal sensitive information, or to perform dangerous tasks while using a HoloLens. Each participant provided informed consent to participate in the study and to be audio/video recorded. We stored all recordings on password-protected drives, removing any personally identifying information from notes and transcripts. We also informed participants that the HoloLens may cause discomfort (such as eye strain or nausea) for certain individuals, and that they could stop the study at any time if they felt discomfort. We also informed participants of Microsoft's own health and safety information for the HoloLens, providing it upon request.

4.2.3 Setup and Hardware

We describe below our study setup and hardware, beginning with details about the Microsoft HoloLens.

HoloLens Details. The HoloLens [50] is an untethered head-mounted display available in a "Developer Edition" for $3,000. Users see virtual objects, or *holograms*, overlaid on a semi-transparent display through which they can also see the physical world, though the field of view within which holograms appear is small (˜30°x17.5°). The HoloLens has multiple sensors [48] that enable *spatial mapping* — the ability to interpret the geometry of a user's environment and overlay holograms in 3D. For example, a user can place a hologram on a table and view it from different angles as if it were physically present. The HoloLens supports third-party applications installed from an app store and can run a single 3D app at a time. User input is given via a tap gesture with the index finger, voice commands, or a single-button clicker.

Study Setup. We conducted the study in a large conference room of our University building. Participants used HoloLens apps (described below), as well as a Microsoft Surface Pro 3.

We used two Windows 10 laptops and HoloLens's "Mixed Reality Capture" functionality to record point-of-view footage. This footage includes a first-person view of the real world, the holograms a user sees, and audio from both the real world and any active application. We also recorded participants from a third-person perspective using a Canon HD camcorder.

4.2.4 Study Procedure

Below, we detail the HoloLens apps and interview questions that comprised our study. We developed our procedure in an effort to avoid participant response bias. For the activities, we acted as observers, only engaging with participants if they explicitly asked us questions. We also emphasized that we were not evaluating the apps themselves, to promote more honest opinions. For the interviews, our questions were broad in scope, allowing participants to focus on the themes that stood out to them the most. We did not press participants for responses on topics where they did not have strong opinions.

At a high level, each study involved an activity-based phase and a semi-structured interview. We conducted two pilot studies (with two pairs) and modified our interview questions in response to the pilot results and feedback, to reduce ambiguity and better meet our research goals. (Our results do not include data from the pilots.) We describe our study procedure below, providing additional details (including our concrete semi-structured interview questions) in the appendix.

1) Interview: Prior AR Exposure. As a baseline, we asked participants to discuss prior AR exposure, including devices or apps that they had used or observed others using, as well as depictions of AR in literature or film that they had seen.

2) Activity: Introduction to the HoloLens. Participants next used a HoloLens tutorial app (Figure 4.2a) to learn gestures and voice commands. They then spent a few minutes exploring the "shell", a single-user app similar to a desktop, from which other apps can be launched and which allows holograms to be placed, mapped to a physical space (Figure 4.2b). For each participant in a pair, we pre-populated the room with one of two sets of holograms

that had some overlap (identical objects placed in the same location), and some differences, to let us observe participants' initial expectations of shared content.

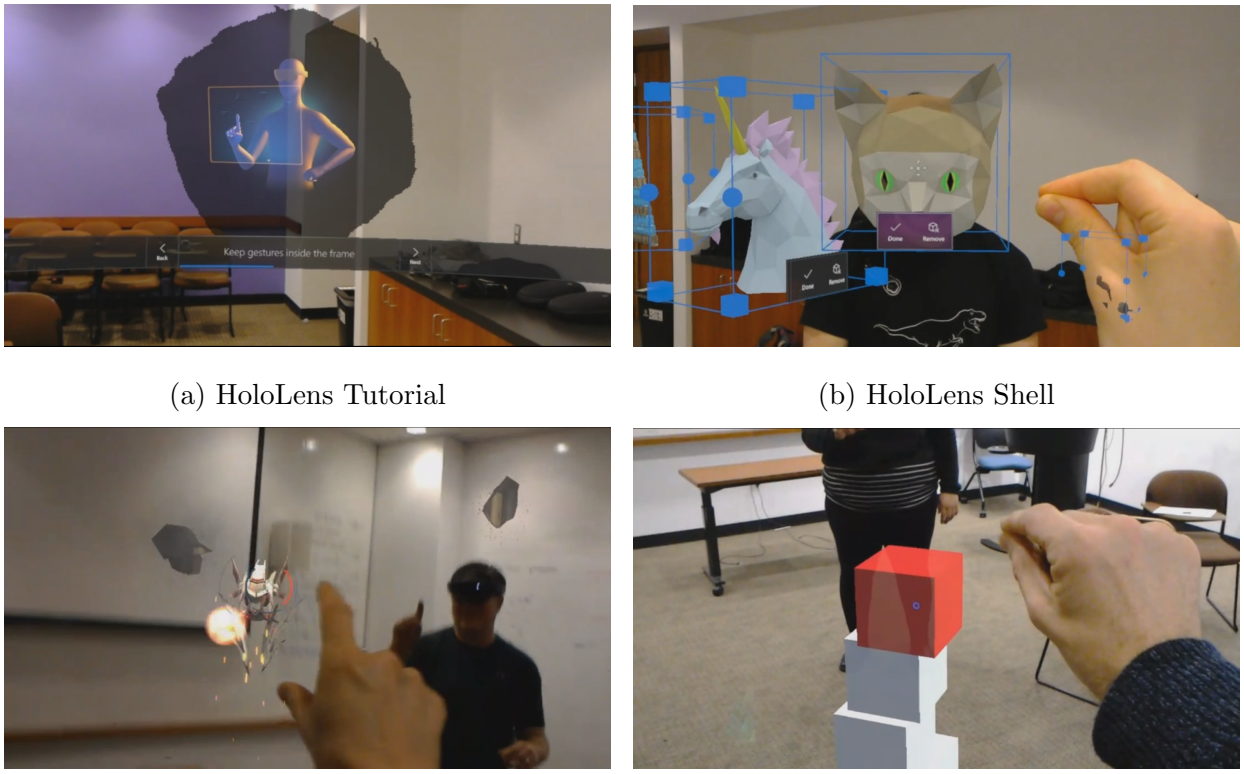
3) Interview: Initial Experience and Brainstorming. After this brief HoloLens exposure, we asked participants to describe their initial impressions of the HoloLens. We then asked them to spend a few minutes brainstorming potential use cases for AR. Though a goal of our study was not to identify concrete use cases, we found through our pilot studies that having participants brainstorm helped them think about AR more concretely and led to more grounded discussions later.

4) Activity: HoloLens Applications. We next asked participants to use each of three apps for five to ten minutes apiece: RoboRaid (Figure 4.2c, a single-player first-person shooter game), Shared Blocks (Figure 4.2d, a multi-player app we built that allows users to create and move blocks in a shared space), and Skype for HoloLens. We chose these apps, in addition to the shell, because they cover different aspects of an AR experience that AR users might encounter. Specifically:

- The shell is a single-user app that allows users to freely interact with multiple 3D holograms.
- RoboRaid is a single-user game that is more immersive and active than the shell. However, its procedural gameplay provides less freedom to experiment than the shell.
- Shared Blocks[1] is a multi-user app that we created to allow multiple HoloLens users to interact in a shared virtual space. Users can create blocks that obey physical properties (e.g., gravity), and either user can move or change the color of any existing blocks. To avoid biasing participants [28], we did not reveal that we built this app.
- Skype[2] is a multi-user app involving one HoloLens user and one tablet user. The

[1]Due to technical difficulties, one pair (P1) instead used Tower Blocks, a shared Jenga-like app available on the HoloLens app store, which is similar to but provides less flexibility than Shared Blocks (e.g., enforcing turns).

[2]Two pairs were not able to use Skype (P1, for whom Skype failed completely) and P7 (for whom the drawing feature on the tablet failed).



(a) HoloLens Tutorial

(b) HoloLens Shell

(c) RoboRaid

(d) Shared Blocks

Figure 4.2: **HoloLens Activities.** First-person views of four of our HoloLens activities (Skype is omitted because it does not work simultaneously with screen capture).

HoloLens user can draw lines in their view, and can see a window with the tablet user's video; the tablet user sees the HoloLens user's first-person view (including their drawings) and can also draw on the HoloLens user's view of the world. Though Skype involves only one user with a HoloLens, given the lack of available multi-user apps at the time of our study, we included Skype for its free-form interaction capabilities.

We uniformly randomized the order in which each pair used the above apps, in an effort to surface as many ideas from participants as possible; a fixed app ordering would have risked missing themes that might arise from alternate orderings.

5) Interview: Reactions, Concerns, and Multi-User Experiences. Upon the conclusion of all HoloLens activities, we interviewed participants, focusing on the following themes.

General Experience. We began by asking participants to describe their general experience,

as well what aspects they found enjoyable, frustrating, confusing, or surprising.

Security, Privacy, Safety, and Other Concerns. We next gave participants the opportunity to raise concerns about AR. Specifically, we asked them to discuss three concrete scenarios, to avoid asking them to think abstractly about AR:

- *Abuse of AR Technology* — how they might harass or disrupt another AR user, or what they might worry about another AR user doing to them.
- *Untrusted Applications* — any concerns they had surrounding applications downloaded from the Internet.
- *Bystanders* — any concerns they would have while acting as a bystander to an AR user that is either a stranger or friend, in either a private or public space.

We emphasize again that while we prompted participants to think about the above concrete scenarios, we designed our questions explicitly to avoid priming participants with *specific* concerns — that is, we did not mention any specific concerns ourselves.

Multi-User Experience. Finally, we asked participants to reflect on their experiences engaging with single- and multi-user apps. We asked if they preferred one setting over the other, and where they might imagine each being useful.

4.2.5 Data Analysis

To analyze data from the study, we used a qualitative, inductive (or "bottom-up") process in which we iteratively developed a set of themes, or codes, from the interview transcripts. First, all researchers independently read a subset of the transcripts and developed an initial set of codes; we then met in person to consolidate these codes into a common codebook. Two researchers then independently coded each interview according to that codebook, iteratively modifying the codebook and recoding previously coded interviews as necessary. Because our goal is to surface a breadth of themes that may arise for emerging AR technologies, we chose to identify the presence of each code in each interview, not distinguishing which of the two participants raised the theme. As a result, a single interview could be coded with two

conflicting codes (e.g., if each participant assumes their shell environments are shared for different reasons — see Table 4.2).

One primary coder coded all interviews, and two other coders independently coded about half of the interviews each. Our final codebook contains 108 codes. After coding all interviews, we met in person to resolve disagreements where possible, resulting in an average inter-coder agreement of 0.98, measured by Cohen's kappa [22]. Fleiss rates agreement over 0.75 as excellent and 0.40 to 0.75 as intermediate to good agreement [37]. Throughout this chapter, we report raw numbers based on the primary coder's values in the cases where disagreements remained due to ambiguity in the interviews.

4.3 Results

We now turn to our results. As a foundation for uncovering the security and privacy risks of emerging AR systems grounded in the experiences of real users, we begin with a discussion of our participants' concrete expectations and interactions (RQ1) in Sections 4.3.1 and 4.3.2. We then explore their concerns around multiple actors (RQ2) in Section 4.3.3, focusing on novel challenges for multi-user AR systems that emerge from these concerns (RQ3) in Section 4.3.4. While we focus on security- and privacy-related themes in this work, we initially coded a broader set of additional themes to capture as many of our participants' reactions as possible. However, we found some of those themes less relevant to understanding the security and privacy risks of emerging AR systems, and thus we do not report on those codes. Furthermore, all numbers and major themes reported are directly drawn from our codes, or from direct participant quotes where appropriate. From the themes drawn from our data, we derive more reflective discussions surrounding the potential implications that our participants' expectations, behaviors, and concerns may have for the security and privacy of emerging AR systems, beyond the sentiments directly expressed by participants.

Participants. 34 individuals completed our screening questionnaire, from which we selected 22 (comprising 11 pairs) to interview. We selected participants who reported not having used

ID	Gender	Age	Profession	Partner Relationship	Previous AR Experience
P1-A	Male	25-34	Entrepreneur	Friends / Coworkers	None
P1-B	Male	35-44	Business Owner and Consultant		Other (Unspecified)
P2-A	Female	25-34	Grant Manager	Coworkers	Smartphone-based AR
P2-B	Female	45-54	Fiscal Specialist		None
P3-A	Male	25-34	Software Engineer	Spouses / Significant Others	None
P3-B	Female	25-34	Attorney		None
P4-A	Male	18-24	Undergraduate Student	Friends	None
P4-B	Male	18-24	Undergraduate Student		None
P5-A	Male	25-34	Graduate Student	Friends	None
P5-B	Male	18-24	Graduate Student		Smartphone-based AR
P6-A	Female	35-44	Middle School Teacher	Coworkers	None
P6-B	Male	35-44	Middle School Teacher		Smartphone-based AR*
P7-A	Male	45-54	Author	Spouses / Significant Others	Smartphone-based AR
P7-B	Female	45-54	Attorney		None
P8-A	Female	18-24	Undergraduate Student	Spouses / Significant Others	Smartphone-based AR
P8-B	Male	18-24	Undergraduate Student		Smartphone-based AR
P9-A	Male	25-34	Commissioned Officer, U.S. Air Force	Coworkers	Google Glass
P9-B	Male	35-44	Non-Commissioned Officer, U.S. Air Force		None
P10-A	Male	18-24	Undergraduate Student	Spouses / Significant Others	Smartphone-based AR
P10-B	Male	18-24	Undergraduate Student		Smartphone-based AR
P11-A	Male	25-34	Law Student	Friends	None*
P11-B	Female	25-34	Law Student		Smartphone-based AR

Table 4.1: **Participant Summary.** The 22 study participants (11 pairs), including their demographic information, relationships, and prior AR use. Participants with asterisks (*) revealed during the interview (but not in the pre-screening survey) that they had 3-5 minutes of prior HoloLens experience, but we did not observe qualitative differences in those participants during the study. Participants with identifiers ending in "A" were the HoloLens users during Skype (while "B" used the tablet).

the HoloLens or a similar device and who were available at times when we conducted the study; we also attempted to maximize diversity among participants. Our participants are summarized in Table 4.1. We conducted interviews during April and May 2017, which lasted approximately 90 minutes each.

4.3.1 Expectations of Augmented Reality

Recall that we designed our study to first give participants experience with a few single- and multi-user HoloLens apps, before conducting a semi-structured interview to investigate their potential security and privacy concerns more directly. In this and the following section, we describe our observations from this initial phase of the study in support of our first research question (RQ1) — in this section, focusing on the *expectations* of AR revealed by our participants' interactions with the HoloLens, its apps, and each other.

In presenting these expectations, we also hypothesize ways adversaries (whether other users or malicious or buggy apps) might violate or exploit these expectations. Indeed, in Section 4.3.3, we will find that many of these concerns arose for our participants themselves after their own hands-on experiences — not just hypothetically for us, as researchers.

High-Level Expectation: AR as Physical. A common theme that seemed to underly a number of our participants' assumptions and behaviors was the treatment of AR content as *an extension of the physical world*, rather than isolated digital content. Indeed, nine pairs mentioned or exhibited the sense that holograms felt "real" or integrated into the real world (e.g., stepping around virtual objects as though they were really present in physical space).

> *"I'm kind of getting mixed up between the AR and the real life."* (P4-A)

The melding of digital and physical worlds is a core part of the vision for AR, and a key aspect that distinguishes AR from other technologies in terms of its positive and possible negative potential. However, the HoloLens as an instantiation of AR still has important limitations, noted sometimes by participants, such as frustrating user input controls (eleven pairs), a bulky form factor (two pairs), and a small field of view (eight pairs). We were thus

surprised at the degree to which our participants were immersed *despite* these limitations — that is, the degree to which participants projected physical assumptions onto virtual objects (also referred to as "holograms").

Concretely, the classes of assumptions and behaviors that we observed our participants making included the following:

Assumption: Virtual Objects are Shared. By conducting a partner study, we were able to observe not only participants' expectations of AR in isolation but also in conjunction with other AR users. Recall that participants first used the shell, a single-user app. Most notably, we found that participants often (nine pairs) initially assumed that *both* they and their partner could see the same holograms, for multiple reasons (Table 4.2). The most common explanation (six pairs) was that the *physical* world is shared. In other words, because both participants see the same physical world, they often expected the *virtual objects* integrated into that world to also be shared.

> *"I kept having the same feeling of ... 'oh come check this out' and then I was like 'oh yeah I only get to see this'. Because it's like through my eyes and I'm used to being a human, and someone else can literally stand next to me and see what I see."* (P6-B)

AR apps may exhibit different sharing behaviors, with some virtual content private and some public, and violations of a user's expectations about what is shared may expose the user to harm. For example, a user may interact with sensitive virtual content without realizing that other users can see it, or they may inadvertently (e.g., verbally) reveal private information that has been shared with them but not with others who are nearby. These risks raise unique challenges for multi-user AR systems, discussed further in Section 4.3.4 and Section 4.4.1.

Assumption: Virtual Objects Act Like Physical Objects. Our study also surfaced a number of assumptions and behaviors that arise even in single-user AR settings. One such assumption is that virtual objects have similar physical properties as physical objects — for

Assumptions of Shared Content (Shell Activity)	Number (of 11 pairs)
Assumed content was shared before or during shell	9
Assumption based on video games	1
Assumption based on partner study context	2
Assumption based on the physical world metaphor	6

Table 4.2: **Shell Expectations.** Participant expectations about whether the world would be shared in the shell activity, and why.

instance, that they will follow basic rules of physics (e.g., not fall through the floor) and that they continue to exist even while not seen, (i.e., object permanence). Indeed, two pairs exhibited a sense of permanence for virtual objects, discussing or treating them as if they were physically present even when the participants could not see them. For example, several minutes after removing the HoloLens, one participant described his experience with a virtual sloth.

> *"I keep trying [to reach out] as if it's still right there. . . . For me, the giant sloth is still filling that half of the room."* (P1-B)

While this sense of immersion enables exciting possibilities, it also raises potential risks. For example, the assumption that virtual objects behave like physical objects could be exploited by adversaries who intentionally violate the expectations of the victim — e.g., to have an object suddenly appear in or disappear from a victim's path, or move in unexpected ways. In fact, many of the concerns voiced directly by participants (Section 4.3.3) stemmed from this sense of immersion.

Assumption: The Real World Would Still Be Visible. We found that five pairs observed (sometimes with surprise) the HoloLens's ability to display nearly opaque holograms that can occlude a user's view — perhaps contributing to the fact that participants treated virtual objects like physical objects.

> *"And now I feel like the [physical] table is invisible. I feel like I can't see the other*

side of the table [that is occluded by a virtual block]. That's crazy." (P9-A)

As AR technologies advance, it will become even harder to identify certain properties of the real world when hidden by virtual objects, a fact that can be exploited adversarially. For example, an adversary could mislead a victim about the nature or even presence of an object in the physical world — e.g., occluding a dangerous physical object, such as a gun, with a benign virtual object. Indeed, this concern was echoed in different forms by our participants (Section 4.3.3).

Behavior: Avoiding Virtual Obstacles. Assuming that virtual objects act like physical objects also caused participants to adapt their own behaviors. For example, one participant attempted to physically avoid holograms, as they might with physical obstacles on the floor, for fear of tripping.

"I'm like worried I'm going to trip on the blocks." (P4-A)

That is, participants not only *assumed* that virtual objects acted a certain way, but this assumption also affected their own actions and reactions in the physical world. We observe that adversaries can take advantage of this effect, such as by placing holograms to cause a victim to perform physical actions that they might not otherwise perform (e.g., swerving quickly or jumping to avoid a perceived obstacle).

Behavior: Physically Manipulating Virtual Objects. Though the HoloLens supports only a simple "air tap" gesture, ten pairs nevertheless tried, or expressed a desire for, more physically-inspired gestures such as kicking, throwing, or grabbing; and indeed, other emerging AR platforms, such as the Meta 2 [73], support more natural gestures like grabbing virtual objects. Such gestures are desirable from a usability perspective but can also raise risks, including safety risks if an app causes a user to act in a way that is unsafe in their physical environment (e.g., causing them to lose balance), as well as privacy risks if other users or their devices can infer a victim's private interactions with a virtual object through their gestures (as an extension of the classic shoulder-surfing attack, but now from any angle).

4.3.2 Inter-Personal Interactions

Our partner study allowed us to observe not only individual participants' expectations and behaviors, but also their interactions with another, physically co-located AR user, continuing our investigation of RQ1 from Section 4.3.1. Though in the study's context these interactions were lighthearted, they nevertheless surface potential tensions between users that have not been deeply studied in prior work on security and privacy for AR. These interactions also directly informed participants' own concerns, as we discuss in Section 4.3.3.

Table 4.3 details ways in which participants interacted with each other during different activities in the study. We report on these interactions below, and going beyond our observations of participants' behaviors, we raise possible tensions or threats that may arise from them.

Visually Modifying Each Other. We observed that participants often attempted to modify the appearance of their partner (or the researchers) using virtual objects. For example, participants in seven pairs tried to draw on other individuals while using Skype (using either the HoloLens or the tablet), and participants in six pairs placed holograms on top of their partner or in front of their face (e.g., Figure 4.2b).

> *"P8-A: I put a cat on your head.*
> *P8-B: I put the world [a globe] on your head."* (P8)

Such interactions can be problematic either if the other user *can* see the hologram on them (e.g., blocking their vision) or if they *cannot* see it (e.g., if an adversary "*put like a digital sticky note on [the user's] back*" (P4-A)). As we discuss in Section 4.3.3, participants voiced concrete concerns along these lines during the semi-structured interview phase.

Shooting at Each Other. Echoing observations from Section 4.3.1 regarding participants' assumptions about shared virtual content, we observed participants target each other with virtual objects even when their partner could not see those objects. For example, while using the single-user app RoboRaid, participants in six pairs attempted to shoot each other (or the researchers, who were not wearing HoloLenses). This example raises the question

Multi-User Interaction	Number (of 11 pairs)
Shell: Put holograms on or in front of another person	6
Shared Blocks: Fought over control of a block	3
Shared Blocks: Put blocks on or in front of another person	5
Shared Blocks: Collaboratively built a structure	5
RoboRaid: Shot at another person	6
Skype: Drew on another person	7

Table 4.3: **Example Interactions.** What pairs of participants did to or with each other during different activities. (Note that the Shared Blocks numbers are out of 10, and the Skype number is out of 9, because those apps failed during some studies.)

of whether uninvolved bystanders will become unwilling participants to other users' AR experiences, and participants later voiced concerns rooted in not knowing what another user sees (Section 4.3.3.2).

Interfering with Others' Objects. When virtual objects *were* shared, as in the Shared Blocks app, participants sometimes attempted to interfere with their partner's objects. For example, participants in three pairs destroyed structures their partner had built, or stole control of blocks from each other.

> *"P4-B: He's messing with my blocks!*
>
> *P4-A: I stole his block and I'm like carrying it around."* (P4)

Though these interactions seemed largely experimental in the context of the study, they nevertheless represent potential tensions between people in multi-user AR settings. As we discuss in Sections 4.3.4 and 4.4.1, these tensions raise critical design challenges for multi-user AR systems and applications around object ownership, visibility, and control.

Using Virtual Objects as Physical Barriers. Building directly on an observation from Section 4.3.1 above, we noted that participants sometimes used the opacity of virtual objects to their own advantage. For example, one participant crawled behind a pile of virtual blocks

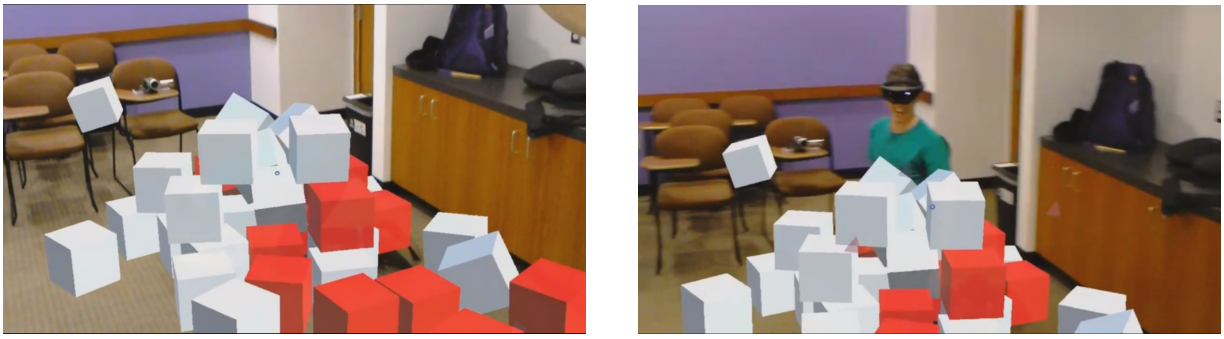


Figure 4.3: **Using Virtual Objects as Physical Barriers.** This participant leveraged the opacity of virtual objects in the Shared Blocks application to hide from his partner behind a pile of blocks (left) and pop out (right).

to hide from his partner and then popped out, as shown in Figure 4.3. Thus, AR enables new risks between multiple people interacting in the *physical* world, not just in the digital world.

Actions Triggered by Commands from Others. When multiple people use AR systems in close proximity, their commands may interfere with each other. Participants in two pairs experienced either gestures or voice commands from their partner (or the researchers) triggering actions on their own device. For example, when a researcher instructed P7-B to say the voice command "next", the participant remarked that the instruction actually triggered the command. In another case, P6-A observed her HoloLens react to a hand gesture from her partner. Although these interactions happened accidentally during the study, they could also be exploited adversarially by people in close proximity to an AR user.

Collaboration. Finally, we emphasize that although in this section we focused on tensions or threats between AR users or physically proximate people, multi-user AR interactions can also enable cooperation, as discussed in Chapter 2. Indeed, participants sometimes worked collaboratively in our study. For example, five pairs worked together to build structures such as towers or forts in Shared Blocks. Thus, a challenge for multi-user AR platforms is to enable these types of collaborative interactions between benign users, while also protecting

users from potential threats from less cooperative users.

4.3.3 End User Concerns

In Sections 4.3.1 and 4.3.2 above, we observed participants' expectations of and behaviors with the HoloLens, and we hypothesized risks that might stem from these experiences. In this section, we shift focus to our second research question (RQ2), uncovering specific risks that our participants surfaced during semi-structured interviews, when presented with several adversarial scenarios and after having experienced a real AR technology. Recall from Section 4.2.45 that we asked participants to consider specific adversarial scenarios involving other users and untrusted applications. We did not, however, prompt them with any specific risks that might stem from these scenarios. Our goal was not to determine the set of adversaries that participants might be concerned about, but rather to identify the spectrum of specific risks that they believe could arise in emerging AR ecosystems.

By providing participants with several open-ended adversarial scenarios, we enabled them to think about concrete situations in which misuse or harm might arise, and allowed them to identify the potential outcomes of those situations that they would find most concerning. For example, while we prompted participants to consider harassment from other users (recall Section 4.2.4 bullet 5), we explicitly did not prompt them to consider specific outcomes such as physiological harm (as discussed in Section 4.3.3.1), and hence all mentions of such harms arose organically from participants.

We organize the rest of this section around the types of concerns that participants raised in response to our adversarial scenarios, rather than around the scenarios themselves, since many concerns arose in response to multiple scenarios. We note inline any situations in which a particular concern referred to a specific scenario. Table 4.4 lists our top-level hierarchical codes that capture these concerns, formed by clustering individual codes for similar, thematically-related concerns. Some of these concerns suggest novel risks and challenges for multi-user AR systems (as we further expand upon in Sections 4.3.4 and 4.4.1), while others validate and add richness to theoretical concerns raised by prior works considering security

Category of Concerns	Number (of 11 pairs)
(4.3.3.1) Physiological Attacks	11
(4.3.3.1) Deceptive Holograms	9
(4.3.3.2) Virtual Clutter	8
(4.3.3.2) Obstruction of Virtual Objects	2
(4.3.3.2) Inappropriate Content	6
(4.3.3.2) Advertisements	6
(4.3.3.3) Bystander Privacy	8
(4.3.3.3) Privacy from Invasive Applications	10
(4.3.3.4) Displaying Content on People	9
(4.3.3.4) Obscurity of Other Users' Actions	8

Table 4.4: **Participants' Concerns.** The concerns that participants raised during semi-structured interviews.

for single-user AR settings.

4.3.3.1 The Risks of Immersion

A unique property of emerging AR systems is the ability to provide immersive experiences that directly impact users' perceptions and actions within the physical world. Indeed, many of the assumptions and behaviors discussed in Section 4.3.1 stemmed from this sense of immersion, which — despite the HoloLens's technical limitations — raised concerns.

> *"This could go really wrong... much more realistic than I thought it would be. It's just an extension again of all the issues that people already have with first-person shooters and everything. There's definitely a line there that gets much more blurry, that like my parents were always worried about growing up, and that's when it was in a fantasy world, well-defined, on a screen in your house.*

> *When that world can mesh seamlessly with a normal place, that's odd... You're getting closer and closer to something that could be kind of — evil's not the right word, but that could just be a little socially uncomfortable."* (P5-B)

More concretely, our participants identified a few specific risks that might arise from immersive experiences gone wrong.

Physiological Attacks. Participants in all (eleven) pairs considered ways that AR content could physiologically harm users, e.g., by startling them or triggering epileptic attacks.[3] For example, one participant considered the possibility of a malicious user startling the driver of an AR-enabled car:

> *"If they're driving or something... throw a digital object at them, and I could imagine it'd go through the windshield."* (P4-B)

Deception. Nine pairs also expressed concern over the use of holograms to deceive users, likely informed by their observations of HoloLens apps convincingly occluding physical objects as discussed in Section 4.3.1. For example, P5-B suggested that a malicious app from one company might overlay their brand logo on physical objects from a competing company, as a form of subversive marketing. P9-A and P9-B also considered ways that one user might mislead another by projecting an alternate visual representation of their appearance, or avatar. Furthermore, P4-B discussed ways to hide physical objects with virtual ones:

> *"I'd probably put something like one of the holograms, something boring and innocuous, on top of something like their car keys or their wallet. I imagine it's kind of like... there's physical clutter, you just wouldn't look underneath it."* (P4-B)

Others considered physical consequences that might stem from deceptive holograms, likely informed by their own tendencies to treat virtual objects as extensions of the physical world, discussed in Section 4.3.1.

---

[3]Such concerns have already manifested even with non-AR technology, e.g., a recent case of a reporter targeted with a seizure-inducing tweet [4].

> *"P11-B: I think what's going to be really interesting is when we start getting to the point in animation in this when it's getting hard to distinguish real versus fake. Like if you're walking down the street and there's an open manhole cover in front of you —*
> *P11-A: Will you think it's the real thing?*
> *P11-B: Yeah exactly! And maybe it's already there, and you just see it in your periphery, maybe you do think it's open. Or maybe there's a real manhole cover in front of you and you think it's fake and you don't need to actually dodge it."* (P11)

The above risks are particularly unique to immersive AR environments, where virtual content can lead to serious physical discomfort or harm. These risks may arise in single-user contexts, e.g., from buggy or malicious apps, or they may arise in multi-user interactions (as we saw foreshadowed in the interactions between our participants). In single-user contexts, these risks further support existing efforts to prevent misbehaving AR apps from generating undesirable output, such as Chapter 3 of this dissertation; in multi-user contexts, these risks raise new defensive challenges, as we discuss further in Sections 4.3.4 and 4.4.1.

4.3.3.2 Unwanted Virtual Content

Whereas the aforementioned concerns largely stem from the immersive potential and physicality of AR, participants also expressed concern about unwanted virtual content more generally. Though such concerns about unwanted content (e.g., ads) may also arise with more traditional technologies (e.g., smartphones), the fact that such content might be overlaid continuously on a user's view of the physical world, rather than confined to a small screen, raises new challenges. As above, defenses must consider — and may differ between — both single- and multi-user AR contexts, as well as adversaries including malicious or buggy applications and other AR users.

Virtual Clutter. Eight pairs worried about becoming overwhelmed by virtual objects

(which some experienced directly while using the HoloLens), or popups. For example, combining the experiences of using blocks to obscure each other's view in Shared Blocks and seeing an animated virtual monkey eating pizza in the shell, P10-B raised the potential for spamming someone with annoying holograms:

> *"Then there's the situation where someone puts way too many holograms and keeps like placing pizza monkeys... that would be kind of annoying."*
> (P10-B)

Obstruction of Virtual Objects. Virtual objects can be used to obstruct not only physical objects, as described above, but also other virtual objects. Two pairs were concerned about this capability. For example, P5-A became annoyed with a virtual chirping bird in the shell, and considered how a malicious user might prevent someone from removing such an object by hiding it among other virtual objects.

> *"I thought of trying to hide [virtual] content from somebody... You put like an annoying little bird and hide him in blocks."* (P5-A)

Inappropriate Content. Participants in six pairs discussed unsolicited or inappropriate AR content, in some cases based upon capabilities showcased in the HoloLens apps used in the study, such as Skype's free-drawing feature.

> *"For example, graffiti... people would be drawing penises everywhere."* (P11-A)

Advertisements. Six pairs expressed concern over unwanted ads. Though this concern arose in the context of asking participants to consider risks with untrusted applications, we note that we did not prime participants to think about ads in particular (nor did any of the HoloLens activities include ads).

4.3.3.3 Privacy

Another general class of concerns arose around privacy — privacy from untrusted applications and other users, as well as privacy of both virtual and physical world information.

Privacy for Bystanders. Eight pairs raised concerns about privacy for bystanders of users with AR devices. Though these concerns echo prior work [29], we note that this prior work studied individuals who did not necessarily have personal AR experience, and who only observed nearby users wearing a mock-up AR device. In contrast, our study design allowed participants to raise concerns informed directly by their *own* experiences using a real, immersive AR device.

Indeed, participants voiced concerns about how an AR device could be used not only to sense information about them as a bystander ("*get their weight, their measurements, their eye color*" (P11-B)), but also to visually augment that sensor data with sensitive information drawn from elsewhere.

> *"If I felt like they had an application that was recognizing me and saying who I was and what my net worth was and where I lived and all that stuff, that would make me uncomfortable."* (P7-B)

P11-A noted that these concerns can arise even if one trusts the AR user, due to *"hackers"* or over-permissioned apps.

Four pairs mentioned ways in which they might change their own behaviors in response to the presence of nearby AR users, suggesting the risk of a "chilling effect" — for example, by becoming "*more conscious*" of what they said or trying to "*appear more composed*" (P4-B). Two pairs also suggested ways to mitigate their privacy concerns, by requiring that friends remove their devices in the participant's home or mandating manufacturer-enforced recording bans — echoing countermeasures explored in prior work (e.g., [96]).

Privacy for AR Users from Invasive Applications. While privacy concerns around AR have often been discussed in the context of bystanders (e.g., echoing early concerns with Google Glass [106]), significant privacy concerns also arise for AR users themselves. Indeed, ten pairs voiced concern about invasive apps compromising their physical-world privacy. These concerns involved AR applications' abilities to both capture visual information about the user's physical surroundings directly (e.g., seeing credit card numbers) as well

as behavioral information about the user (e.g., pulse and eye tracking enabling sensitive inferences).

> *"There's all kinds of really subtle things that an AR headset would be able to tell about you that in an advertisement sense would be really powerful. So to have a marketer have knowledge of like 'you have a crush on this person because you can't stop looking at them' is pretty scary."* (P5-B)

These concerns further support the need for solutions to restrict sensor data available to AR apps, already explored in prior work (e.g., [36, 55, 91, 96, 108]), to protect the privacy of both bystanders and AR users themselves.

Private Holograms. Finally, when we asked participants explicitly about scenarios in which shared or private AR experiences would be useful, they had concrete ideas for both use cases. For example, P6-A mentioned private use cases like "*porn*" or "*Skyping a friend*" as well as shared use cases like "*creating games and art together*". Participants often implied that their private use cases should be hidden from other users:

> *"If I were navigating somewhere, I'd want to be able to keep that sort of thing private."* (P4-B)

Though participants did not voice as an explicit "concern" the idea of someone else seeing their private holograms, their desires for private content within AR suggest that multi-user AR platforms must protect that content. We further discuss challenges with managing shared and private virtual content in multi-user AR interactions in Sections 4.3.4 and 4.4.1.

4.3.3.4 What Other AR Users See

Concerns arose for participants regarding not only virtual content on their own devices, but also the virtual content that *others* can see.

Displaying Content on People. Participants worried about the type of virtual content that other AR users might overlay on top of them or other nearby people. A common concern

(nine pairs) was the prospect of someone using AR to modify another person's appearance — a concern potentially informed by their attempts to visually modify each other while using the HoloLens (e.g., placing holograms on each other's heads), as described in Section 4.3.2.

> *"Can you do that with HoloLens, change somebody? Like you're looking at somebody and you can change what they look like? It's kind of like you would do with Snapchat... That gets kind of psychologically wee-ooh-aah... Can you imagine people married, and they imagine somebody else?"* (P6-A)

Further, participants in two pairs were concerned about the potential for AR users to display personal ratings around others, or to have *"social scores floating by them"* (P1-B). P1-B also considered the idea of displaying embarrassing facts above a person's head that nearby users could see.

Though augmenting people with virtual content is promising to explore (e.g., displaying the names and affiliations of people at an academic conference, or modifying people's appearances with permission during a costume party), our findings suggest that they should also be designed carefully to consider potential misuse or unexpected social consequences.

Obscurity of Other Users' Actions. When virtual content is not shared between multiple AR users, or when a non-AR user interacts with an AR user, multiple people may see different views of the same physical space. Particularly for emerging AR devices like the HoloLens, which provide a private heads-up display for a single user (unlike AR content displayed in a smartphone app), how — and even whether — these views differ can be hidden from other people.

Indeed, participants in eight pairs discussed the assumptions they might make, and the social challenges that might arise, if they could not tell what an AR user was doing.

> *"If they were just kind of staring off into space I'd assume they were checking their email or watching a YouTube video or something like that, but if they were staring at someone, or like staring at different people, maybe something more malicious."* (P4-A)

Both of the above classes of concerns (overlaying on people and the obscurity of an AR user's actions) may manifest for bystanders as well as other AR users seeing different virtual content. However, we observe that multi-user systems have an opportunity to help *mitigate* these concerns. For example, future work might explore mechanisms for AR users to provide some degree of transparency about their actions to other AR users, without leaking private information (e.g., the same way putting down a physical phone signals that one is paying attention). Additionally, our findings suggest that providing users with recourse over unwanted augmentations "attached" to them in some way may ease concerns.

4.3.3.5 Lack of Concern

As discussed in this section, participants raised many concerns surrounding AR technologies. However, we also observe that some participants were notably unconcerned about the potential for AR to be abused by other users or applications.

> *"I don't think I'm really that worried about things that people would do to me. AR wouldn't really be somewhere that I'd feel unsafe... especially because you can see the real world."* (P8-A)

Some users may not view risks of AR as impediments to adoption, and indeed there may be circumstances in which this lack of concern is warranted (e.g., when interacting with trustworthy users or well-vetted apps). Nevertheless, where there are disconnects between users' mental models of AR and what is technically possible, there may be an opportunity for researchers and developers to help shape users' expectations and take measures to protect users from abuse.

Further, from understanding *why* users might lack concern, we can develop an intuition for possible defensive measures. For example, the lack of concern in the above quote rests on the ability to "see the real world" — emphasizing the value, from a defensive perspective, of enabling users to reliably perceive the physical world (either at all times, on demand, or when a possible security situation arises).

4.3.4 Challenges for Multi-User AR

Above, we presented a rich variety of concerns our participants raised about risks that may arise in both single- and multi-user AR interactions. Indeed, prior works have identified some of these risks and explored defensive strategies to protect users and bystanders of single-user AR systems. However, our findings suggest that many of these risks can also arise due to other, adversarial AR users — and, as we discuss in Section 4.4.1, defensive techniques designed for single-user systems may not translate well to multi-user AR settings.

In this section, we thus return to our third and final research question (RQ3): what new challenges will arise when considering defensive strategies for *multi-user* AR systems? Although we explore this question in greater depth in Section 4.4.1, we found that our participants presented valuable perspectives to guide this discussion. In particular, we highlight key tensions that arose surrounding *ownership* and *access control*.

Ownership of Virtual Objects and Physical Spaces. By definition, multi-user AR systems allow multiple users to interact with shared virtual content. Determining the precise nature of this sharing raises questions such as: *what* content created by a given user is shared with *whom*, and *how* can those other users interact with this content?

In the least restrictive case, all users could create virtual objects and expose them to other users, and freely view and interact with the objects created by other users (as in our Shared Blocks app, for example). However, it is precisely the potential for unrestricted interactions that appears to form the foundation of many of our participants' concerns.

> *"It feels like the kind of experience where I'd feel powerless very quickly... if somebody started making all of my blocks or all of my things disappear, or started putting a bunch of windows in my face, I would feel so powerless about what to do."* (P5-A)

In particular, the above sentiment highlights an important desire expressed by many participants — a desire for *ownership* over their AR environments, including ownership over:

- *Virtual objects* perceived as belonging to the user. For example, recall from Sec-

tion 4.3.2 that one user stole a block created by his partner, and others destroyed block structures built by their partners. Participants' reactions often (seven pairs) suggested a sense of ownership over their own blocks. Recall also from Section 4.3.3.3 some participants' desires for private virtual content (e.g., while navigating somewhere).

- *Personal space.* The above quote suggests that users may desire not only control over their virtual objects but also control over their *physical personal space* (e.g., to prevent objects from appearing in their face). In AR, virtual objects may feel as though they are physically infringing on the user's personal space or may directly impact their perception of the physical world. When considering multi-user systems, a variety of concerns from Section 4.3.3, ranging from virtual clutter to socially uncomfortable overlays, are intimately tied to the ability of misbehaving users to place unwanted virtual objects in the environments of victim users.

Access Control. The above perspectives raise a key challenge: how can multi-user AR systems give users control over their virtual objects and physical spaces, to prevent undesirable interactions with other users? While we step back and discuss this question further in Section 4.4.1, many participants arrived at this question — and possible answers — on their own, as a result of their HoloLens experiences and general concerns.

Edit Permissions. Five pairs expressed a desire for edit permissions, i.e., mechanisms to prevent other users from freely creating, changing, or deleting objects in their view.

> *"If access to apps was not controlled, then anyone could introduce any app and just interrupt your environment at any time. So for example you're wearing this and you're trying to just navigate the streets without interruption, and someone decides to drop a dragon in the street in front of you."* (P11-A)

View Permissions. Participants in nine pairs also discussed a need for view permissions on virtual objects, to prevent other users from seeing their own private content. The perceived appropriateness of shared or private experiences was often highly contextual — for example, some individuals preferred primarily private content.

> *"Very case by case. Definitely would want an opt-in system, like 'I want to share this object', because I think there's a lot more stuff I'd rather keep [private]. Like 9 times out of 10 I'm not showing people stuff on my phone. Fewer cases where I share stuff. Definitely want, like, tap-to-share."* (P4-B)

Others preferred primarily shared experiences.

> *"I'd like to think that predominantly the reality was shared, and then you had the option to not share if you wanted to, but I would like to think that the default would be like 'hey we're all in the same reality'... And I think that it would actually further the adoption of the technology if people felt like it was a more communal experience as opposed to the haves and have-nots."* (P9-A)

Specific Access Control Mechanisms. In terms of *how* users should manage such edit and view permissions, some participants suggested concrete mechanisms to support explicit sharing decisions (e.g., "*tap-to-share*" from P4-B, above).

> *"It would be really interesting if... it's like 'anything you put on the purple wall is shared'. So then I could have my own environment over here and I could be working and I could be like 'hey check this out' and I throw it up on the purple wall, and then [P8-B] can see it."* (P8-A)

Participants also hypothesized visual aids to help them understand which of their objects are shared:

> *"The color of the window or the adjustable [object bounding boxes] or something — if it would be red for private ones that other people couldn't see and green for public ones that other people can see, instead of I think everything is blue right now, that would be super useful."* (P1-A)

While these mechanisms are by no means the only possible solutions, they provide starting points. Further, the fact that our participants came up with concrete access control mechanisms organically, without being asked to think about such mechanisms, suggests that

they valued access control as a design objective. As we discuss next, emerging multi-user AR apps and platforms *must* consider and address these questions.

4.4 Discussion

Our results — the exploration of user expectations, behaviors, and concerns with a real AR device — allow us to draw broader lessons and recommendations to inform the design of emerging AR technologies, which we present below. We also identify limitations of our study and avenues for future work.

4.4.1 Security and Privacy Design Challenges for Multi-User AR

Our findings provide a foundation for understanding and addressing security and privacy for multi-user AR systems — a space that has remained until now unexplored. Below, and continuing to answer RQ3, we identify key design challenges drawn from these findings.

Controlling Access to Personal Objects. Participants desired both view and edit permissions, to restrict others from seeing or modifying their personal holograms (4.3.4). While some considered how an AR system might support this control (e.g., "*tap-to-share*" from P4-B), determining appropriate mechanisms remains an open question. This challenge is further complicated by the fact that different users will place differing levels of importance on shared and private experiences (4.3.4).

Preventing Unwanted Content from Other Users. The ability for a user to prevent other users from sharing unwanted content is also critical. Many of our participants' concerns, such as virtual clutter and inappropriate content (4.3.3.2), were rooted in a lack of such control. However, as above, AR systems will need to determine appropriate mechanisms that account for diverse user sharing preferences.

Negotiating Access to Other Users' Content. Users will also require mechanisms to easily *initiate* sharing requests. If not carefully designed, such mechanisms could (for example) result in a user spamming a victim with requests or accidentally sharing content

with the wrong user. One approach may be to leverage a user's physical environment (e.g., "*anything you put on the purple wall is shared*" (P8-A)).

Navigating Partially Shared AR Environments. Users may make different choices in terms of what they share, with multiple interacting users seeing different (possibly overlapping) sets of virtual content. As we saw (4.3.1), incorrect expectations of sharing can leave users vulnerable to confusion or harm. AR systems thus have an opportunity to help users better understand what content is shared with whom.

Designing Access Control UIs. The above challenges will all require AR systems to instantiate access control mechanisms with careful UI design, to ensure that the mechanisms can appropriately assist users. While we can draw initial ideas from our participants, such as objects with different colored borders indicating whether they are shared or private (4.3.4), access control UIs for multi-user AR remain an open area of study.

Managing Personal Space in AR. While the above challenges involve control over virtual objects, recall from Section 4.3.4 the equally important need to provide users with control over their *physical* personal spaces. Addressing these concerns raises the fundamental challenge of defining personal space in AR, and determining how to best manage the personal spaces of multiple users who may cross paths.

Insufficiency of Single-User Defenses. The above challenges highlight a fundamental tension of multi-user AR systems between supporting flexible shared experiences and preventing unwanted interactions — challenges that may require novel defensive solutions where existing single-user defenses prove insufficient. For example, our prior work proposed mechanisms to prevent undesirable application output in single-user AR contexts (e.g., [63]), and others explored defenses to shield sensitive information from invasive applications (e.g., [36, 55, 91, 108]). However, if applied naively to multi-user systems, the above defenses may lead to unexpected conflicting views or inconsistent application states that impede desirable interactions between users.

4.4.2 Grounding Concerns in User Experiences

When considering emerging technologies that are just beginning to gain traction, it is critical to understand the expectations, desires, and potential interactions of end users. While others have conceptually explored the security and privacy challenges presented by emerging single-user AR systems, we find that our study of real users engaging with multi-user AR devices both illuminates new challenges (discussed above) and enriches concerns raised in prior works.

A wide variety of concerns emerged for our participants in response to even limited exposure to a sophisticated-yet-imperfect AR device, when prompted by our adversarial scenarios. Understanding how users envision risks might arise can inform defensive directions previously based only on conceptual risk assessments. We give two examples. First, prior work proposed a framework to enforce policies that constrain virtual content displayed by AR applications [63], deriving potential policies from several sources (e.g., the HoloLens developer guidelines). Our findings can help expand and enrich this set of policies based on the concerns of real users. As another example, while past work proposed techniques for bystanders to prevent nearby AR devices from recording them [96], our participants' concerns about unwanted holographic overlays on people suggest an opportunity to expand these techniques to also prevent nearby AR devices from *overlaying* on others.

4.4.3 How These Concerns Might Arise in Practice

Although immersive AR technologies are still quite young, we now reflect upon ways in which our participants' concerns might eventually arise in future AR ecosystems, given the variety of desirable AR use cases being explored. We briefly consider three scenarios in which these concerns could arise, in the absence of appropriate defensive measures.

AR-assisted Driving. Both industry (e.g., [71, 82]) and research efforts (e.g., [11, 102, 110]) continue to explore opportunities for AR-assisted driving (such as tools that overlay speed and braking information of nearby vehicles). Given the safety-critical nature of driving, malicious or buggy AR content could greatly endanger the driver or others nearby. For example

(as one participant noted), a digital object that appears as though it was thrown through the windshield could startle the driver. As another example, a deceptive application might misrepresent real-world information, e.g., by occluding pedestrians, changing the values on speed limit signs, or presenting false information about the speeds of nearby vehicles.

Shared AR Art. AR could enable a unique medium of artistic expression, where content creators can layer publicly view-able, digital art or graffiti atop the physical world without modifying the world itself. Ideally, different users within the same physical space could subscribe to their favorite artists for carefully-curated experiences. However, in the absence of an appropriate sharing protocol or access control capabilities, viewers may be subjected to visual spam or inappropriate content from misbehaving parties, with little recourse beyond simply shutting off their application. These concerns were held by many participants, and we have indeed begun to see precursors of such issues already with Snapchat [70].

AR in Schools. Prior work has explored AR as a tool for mathematical education [59], and in future classrooms of all ages, we might see AR used for other educational purposes. However, left unchecked, this technology could manifest as another vector for bullying and abuse among youth. For example, our participants grew concerned about digital content being overlaid on people. An AR application might be used to place a virtual "kick-me" sign or other malicious object on a victim, and without adequate control over his or her personal space, the victim may have no recourse to remove it or prevent others from seeing it.

4.4.4 These Concerns Manifest in Current-Generation AR

The concerns raised in Section 4.3 may seem like issues only for future-generation AR technologies, and indeed we began this work with that assumption. However, our findings suggest that these concerns are in fact imminent, even for today's imperfect AR technologies. That is, our participants' behaviors and interactions demonstrated that the HoloLens — despite its clear limitations — is *already* sufficiently immersive to blur the line between physical and digital experiences, and to elicit serious concerns. For these concerns to manifest as real

threats in the AR ecosystem, we need only see an increase in adoption by users and app developers, not a fundamental shift in the underlying technologies.

4.4.5 Limitations

Finally, we note several limitations of our study. First, our study was qualitative, and thus we cannot draw quantitative conclusions or generalize our results to a broader population. Instead, the goal of a qualitative study is to surface a broad set of themes — in this case, security and privacy issues around emerging AR technologies. We also do not evaluate how *likely* participants believe specific risks are, focusing instead on their breadth of concerns. Future work should consider studying these questions in a larger-scale quantitative study.

Our study is also likely influenced by our choice of AR technology, the HoloLens. We chose the HoloLens because it is one of the most immersive AR devices commercially available. Though some of our findings are thus HoloLens-specific (e.g., reactions to opaque holograms), they raise lessons that extend to AR technologies more generally. Additionally, recall that participants' expectations were often rooted in their treatment of AR as an extension of the physical world (4.3.1). While we center our discussion on physically co-located interactions, future work should also explore how these assumptions do (or do not) change for users engaged in remote interactions.

Though we aimed to recruit diverse participants, our participant pool was likely biased towards people who wanted to try the HoloLens, and who may be more tech-savvy, more likely to be early technology adopters, and more positively disposed towards the technology. Though future work may wish to consider other groups (e.g., people *disinclined* to use AR technology), our results highlight important security and privacy challenges that emerging AR technologies will raise.

Users may behave differently after extended experience with an AR device than during a ninety-minute session. In our work, we aimed to study users' *initial* expectations and experiences, unhampered by pre-existing knowledge that might constrain a more experienced user's perspective. However, these findings may not generalize to more experienced users,

and as these technologies become more widely used, future work should study longer-term users' experiences and concerns.

4.5 Conclusions

In this work, we identified the fundamental need to explore the security, privacy, and safety challenges of emerging single- and multi-user AR technologies, grounded in the experiences of end users. Through a qualitative lab study with 22 participants (11 pairs), combining hands-on activities with semi-structured interview questions, we studied the expectations, interactions, and concerns of users engaging with the Microsoft HoloLens, an immersive AR headset. We found that participants were easily immersed in HoloLens experiences, treating virtual objects as real despite nontrivial limitations of the current technology; that participants raised a variety of concerns around misuse by multiple actors, including other users and applications; and that multi-user interactions raised fundamental tensions around access control for virtual objects embedded into shared physical spaces. Our findings give us the opportunity to draw broader lessons and suggest key design challenges for future AR technologies, including previously unexplored multi-user issues. This chapter thus lays a foundation for understanding and addressing the security, privacy, and safety risks that emerging AR technologies will present.

Chapter 5

ENABLING MULTIPLE APPLICATIONS TO SIMULTANEOUSLY AUGMENT REALITY: CHALLENGES AND DIRECTIONS

This chapter shifts focus to the challenge of allowing multiple simultaneously-running AR applications to augment a user's world. My collaborators and I began considering this challenge while working on the Arya project described in Chapter 3; however, Arya focuses predominately on addressing output security risks from *individual* AR applications, with only a limited ability to handle output conflicts *between* apps. This chapter describes further research that my collaborators and I conducted to identify the means of visual conflict that may occur between AR apps, and it proposes design strategies for AR platforms to mediate such conflicts. This work originally appeared in the 20^{th} International Workshop on Mobile Computing Systems and Applications [62].

5.1 Overview

Today's AR platforms do not typically allow users to engage with more than one application a time, and those that do provide multi-app support have many limitations. However, users may benefit from the ability to engage with multiple applications at once, without having to exclusively choose between any single app. For example, an AR user might wish to use multiple apps as they travel through a city. These may include an AR navigation app to help the user find a destination [44], an AR game that the user interacts with in the proximity of specific real-world landmarks [81], and social media apps that help the user connect with people around them by displaying information such as names and common interests above people's heads. These applications all contribute to the user's total AR

experience in complementary ways. A multi-app AR platform would thus allow the user to shift their attention between apps as they wish, rather than only limiting users to viewing and interacting with one app at a time.

Realizing the vision of multi-app AR will require identifying and overcoming new challenges that stem from the unique capabilities of AR platforms. In particular, rather than sharing the blank canvas of a traditional computer screen and displaying content within isolated windows, the output of immersive AR apps will exist atop the backdrop of the user's ever-changing world. These apps may need to dynamically update their outputs in response to changes in the user's physical environment while simultaneously displaying content alongside each other, raising fundamental questions: how might immersive AR apps visually conflict with each other, and how can multi-app AR platforms allow different apps to simultaneously augment their shared world while mediating conflicts?

Prior AR-related efforts, including the work described in Chapter 3 of this dissertation, primarily focused on *individual* apps negatively influencing users' perceptions of the real world, rather than on visual conflicts between multiple apps [2, 61, 63, 95]. We currently lack a foundation for reasoning about these conflicts or understanding the design challenges involved with supporting multiple immersive apps. In this work, we provide such a foundation by conducting an investigation into the multi-app AR design space, deferring implementation and experimental evaluations to future work. Specifically, we contribute the following:

1. *Problem Identification*: We identify the need to view the design space of multi-app AR platforms with a critical eye towards visual conflicts that may occur between the output of different apps.
2. *Design Space Exploration*: We introduce a broad categorization of approaches for multi-app AR platforms to handle conflicts, and we uncover key trade-offs presented by different design strategies.
3. *AR Platform Analysis*: We analyze the multi-app capabilities of modern AR platforms to understand how they fit into the broader design space.
4. *Future Directions*: Through our exploration and analysis, we identify promising di-

rections for future work. For example, we encourage future work to implement and evaluate key concepts set forth in this chapter.

5.2 Motivation

We begin with case study scenarios that highlight the possibilities of multi-app AR, including risks that users may face from visual interactions between apps.

Tourism. Alice uses Tour Guide while on vacation, which displays floating icons above landmarks that she can select to read more information. Restaurant Assistant displays food safety and customer ratings above nearby restaurants, which Alice can select to read detailed reviews and menu options. Navigation guides Alice as she walks to a new destination by displaying directional arrows on the ground, and for entertainment, an immersive Pokémon game blends interactive 3D characters into Alice's physical environment.

Unfortunately, multiple Pokémon characters inadvertently stand atop Alice's Navigation arrows on the ground and prevent Alice from seeing her directions. At the same time, Tour Guide has an endorsement contract with a local café, and to discourage Alice from eating elsewhere, it displays fake negative ratings above other eateries that block the true ratings of Restaurant Assistant.

Social Gatherings. Bob is attending a festival with friends and wishes to connect with other attendees. Social Media AR recognizes nearby people in Bob's extended network and displays their names, mutual friends, and common interests above their heads. Since Bob is interested in romantic connections, he also uses AR Dating, which computes compatibility scores of other users, highlights them, and displays the scores above their heads. Finally, Bob and his friends use Immersive Snapchat Filters to modify each other's appearances in fun ways, such as overlaying humorous costumes.

Bob notices that a friend-of-a-friend is also identified as a potential romantic connection, with Social Media AR and AR Dating both displaying information above their head. However, content from both apps appears jumbled atop each other, and Bob cannot disam-

biguate content from either app. AR Dating also identifies other potential partners near Bob, but since Snapchat has already displayed full-body filters over them, AR Dating cannot highlight them.

The Workplace. Carol and her colleagues use AR to improve productivity at work, with Collaborative Workspace allowing them to interact with shared 3D models and virtual whiteboards both in the office and remotely. Colleague Assistant displays helpful reminders that float next to Carol's coworkers (such as upcoming meetings or recent emails), and AR Chat allows Carol to stay connected with her team by displaying real-time messages that float next to her. Finally, AR Art lets Carol easily personalize her workspace with virtual paintings, sculptures, and other artwork.

Carol finds Colleague Assistant helpful, but the app is compromised and intentionally positions its reminders to obscure AR Chat messages. While AR Art improves Carol's workplace ambiance, the app is buggy and creates 3D objects that interfere with Collaborative Workspace. Since there is no indication that AR Art created these objects, Carol believes Collaborative Workspace to be malfunctioning and disables it. Additionally, when an AR Chat message moves atop an AR Art piece on Carol's desk, the art is "knocked" to the ground.

A New Output Paradigm. The above scenarios raise a fundamental question: can a multi-app AR platform support the diverse needs of immersive apps while also mitigating negative interactions between them? As with apps on other computing platforms, immersive AR apps may compete for resources such as memory, CPU cycles, and network bandwidth. What sets these apps apart are their output needs.

Consider a traditional desktop app, such as a video game, text editor, or web browser. The outputs of these apps exist within independent windows, and the behavior of these apps does not depend upon the precise placement of their windows on the computer screen (i.e., the user could reposition any of the windows and the apps would behave the same). However, in AR, the behavior of an app may depend directly on how its outputs are positioned in the

context of the user's world. For example, the efficacy of Alice's NAVIGATION app depends upon the app's ability to precisely position directional arrows on the ground, and Bob's AR DATING and SOCIAL MEDIA AR apps must be able to place overlays above specific people's heads. Furthermore, on a traditional desktop display, all content shown on screen is controlled directly by either apps or the OS. By contrast, users will view AR apps atop the backdrop of the physical world rather than a blank screen. This external environment may change unpredictably, introducing variability that AR apps may need to contend with. For example, apps may need to dynamically update their outputs in response to changes in the user's world itself (e.g., AR DATING must update the locations of its overlays as people move throughout Bob's field of view), as well as changes in the user's own position within the world (e.g., NAVIGATION must appropriately place new arrows on the ground as Alice walks around and changes directions). AR presents a new output paradigm from traditional displays, creating new challenges that will require novel solutions.

Threat Model. In this work, we focus on the conflicts that stem from visual interactions between immersive AR apps, leaving a discussion of additional output modalities (e.g., audio) for Section 5.5. Furthermore, we focus on users' *perceptions* of AR content rather than their *interactions* with apps. Output conflicts may lead to harmful user interactions (e.g., AR "clickjacking"), but such issues depend on the specific input capabilities provided by an AR platform, which we consider out of scope.

Our threat model encompasses both apps that are malicious, as well as apps that are honest-but-buggy and do not intentionally seek conflict. We begin by considering a broad space of visual conflicts that may arise, including the following:

- *Occlusion.* The output of one app might block the user from seeing that of another. For example, Alice, Bob, and Carol all encounter occlusion above. We exclude situations where the user intentionally positions one app's content to occlude other apps, focusing on occlusion events that arise in the absence of user intent.
- *Placement Denial.* By occupying a particular space, one app might prevent another

from generating content. For example, Bob's SNAPCHAT app prevents AR DATING from highlighting certain individuals, by occupying the space around them with full-body filters.

- *Eviction.* By moving content into a space occupied by another app, an offending app might cause the victim's content to be removed or displaced, as Carol experiences when AR CHAT knocks an AR ART object to the ground.
- *Masquerading.* One app might generate content that is mistaken for that of another. For example, Carol mistakenly perceives buggy output from AR ART as output from COLLABORATIVE WORKSPACE.
- *Content Modification.* As we will see in Section 5.3.2, certain conflict mediation mechanisms may modify the visual properties of app outputs, e.g., by adjusting transparency. Such approaches raise an additional threat: one app may be able to induce changes in the visual properties of another app's content.

5.3 Design Space Exploration

We now turn to our design exploration of multi-app AR platforms, asking: how can these platforms mediate visual conflicts between apps, and what are the trade-offs associated with different design alternatives? We consider the ability of an AR platform to meet the following criteria while remaining resilient to the above-mentioned conflicts:

- *Support for Multiple Applications.* Does the platform allow multiple apps to run simultaneously?
- *Full Output Autonomy.* Does the platform give apps full control over the placement of their outputs in 3D space?
- *Some Output Autonomy.* Does the platform give apps at least *some* positional control over their outputs?
- *Limited User Burden.* Does the platform require limited or no user involvement in managing output?

			Output Conflicts					Functionality Goals				
			Occlusion	Placement Denial	Eviction	Masquer-ading	Content Modification	Multi-App Support	Full Output Autonomy	Some Output Autonomy	Limited User Burden	Limited Dev Burden
Display Abstractions	Single App		✔	✔	✔	✔	✔	N	Y	Y	Y	Y
	Windows		✔	✔	✔	✕	✔	Y	N	N	N	Y
	Shared World		✕	✔	✔	✕	✔	Y	Y	Y	Y	Y
Output Management in a Shared World	Runtime Policies	Ex1: Modify Occluder	✔	✔	✔	✕	✕	Y	N	Y	Y	N
		Ex2: LRU	★	✔	✕	✕	✔	Y	N	Y	Y	N
	Declarative Output	Ex1: Defined Layout	✔	✕	✔	✕	✔	Y	N	Y	Y	N
		Ex2: LRU	✔	✔	✕	✕	✔	Y	N	Y	Y	N
	Application Self-Management		★	✔	✔	✕	✔	Y	Y	Y	Y	N
	User-Managed Output		✔	✔	✔	✕	✔	Y	N	Y	N	N

Figure 5.1: **Potential Design Paths for Multi-app AR Platforms.** Check marks indicate that a design can prevent a conflict; stars indicate that the conflict is prevented when apps are trusted; and Xs indicate that a design cannot prevent the conflict.

- *Limited Developer Burden.* Does the platform limit the need for app developers to handle unexpected interactions with other apps?

Figure 5.1 summarizes key trade-offs that characterize the design paths we discuss throughout this section.

5.3.1 Display Abstractions

The interface that an AR platform provides to apps for displaying content determines the space of available output behaviors. Consider the following:

Single-App. Inter-app conflicts cannot occur if only one app can display content at a time. While this approach is at odds with our goal of supporting multiple apps, it is the only design in Figure 5.1 to meet every other goal and may suffice for individual apps requiring the user's undivided attention.

Windows. One method for preventing output conflicts is to confine apps to separate regions of space — a 3D analogue of the window abstraction used by desktop PCs. We consider a model where windows are controlled by the user and cannot be created or repositioned autonomously by apps. These properties allow windows to visually isolate apps from each other, but in doing so, they trade-off the ability for apps to dynamically generate content throughout the user's world. While our prior work argued for the insufficiency of windows in AR due to such flexibility limitations [61], we find that the viability of a window-like abstraction actually depends upon the needs of specific apps. For example, Carol's AR Chat, AR Art, and other apps naturally fit within bounded spaces, but Alice's Pokémon and Navigation apps require more dynamic output capabilities.

Shared World. The final model we consider is a shared world that allows multiple apps to simultaneously display content throughout the user's environment. This approach stands in contrast to windows, sacrificing visual isolation to give apps the flexibility to place AR content wherever they wish. As a result, one app may draw in the same space as another app or otherwise occlude that app's output. We explore strategies for addressing such conflicts below.

5.3.2 Managing Output in a Shared World

When considering how to manage output conflicts in a shared world, we must first determine *who* should shoulder this burden. Thus, we explore opportunities for the OS, apps themselves, or the user to take on this responsibility. While we present these design paths individually, we note that they may be combined to manage output in different ways.

5.3.2.1 OS-Enforced Conflict Mediation

As discussed above, giving apps the freedom to place content wherever they wish may lead to occlusion conflicts. We thus begin with two complementary design paths that enable the OS to prevent occlusion. These designs leverage the AR object abstraction proposed in our prior

work and discussed in Chapter 3 of this dissertation [61, 63]. AR objects are OS-managed primitives that encapsulate AR output — for example, a single POKÉMON character would be one AR object. The OS can modify the visual properties of AR objects (e.g., position or transparency) to prevent occlusion. Specifically, we introduce the following approaches:

1. *Runtime Policies.* The OS prevents occlusion by observing visual interactions between AR objects at runtime and enforcing policies that modify them in response. For example, the OS could observe when one of Alice's POKÉMON objects occludes a NAVIGATION arrow and turn the POKÉMON object partially or fully transparent to ensure that NAVIGATION's arrow remains visible.
2. *Declarative Output.* The OS provides apps with a language to abstractly indicate their output needs, but it controls *how* these needs are met to prevent occlusion. For example, Bob's AR DATING and SOCIAL MEDIA apps could request to display content above someone's head, and the OS would determine an appropriate layout. Similarly, Alice's RESTAURANT ASSISTANT app could place virtual signs in front of restaurants without controlling the precise 3D coordinates of these objects.

Trade-off: Intelligent Mediation vs. App Freedom. Runtime policies only allow the OS to identify occlusion after it has occurred, and they provide no contextual information about how the OS should respond to individual conflicts. By contrast, declarative output ensures that apps do not conflict in the first place, and by capturing the high-level needs of apps, it gives the OS the ability to intelligently respond to app requests. Consider AR DATING and SOCIAL MEDIA from above. If the OS understands that both apps are attempting to augment the same person's head, it could (for example) arrange content so that both apps are visible above the person's head, rather than making one app's objects invisible.

In providing more effective mediation capabilities, declarative output trades off the ability to support fine-grained object placement for apps. Declarative output naturally caters to apps that can specify their output needs in terms of high-level visual relationships to physical-world objects, such as AR DATING. However, this approach does not lend itself to apps such

as Alice's POKÉMON game, which needs to create and move characters at precise 3D locations in Alice's world. For apps such as POKÉMON that cannot operate under a declarative model, runtime policies provide the OS with a potential fallback mechanism for mediating conflicts.

Preventing Occlusion Can Enable New Conflicts. Preventing occlusion in a shared world fundamentally requires the OS to constrain the output behaviors of apps. In doing so, the OS may enable new forms of conflict. Recall the example runtime policy in which POKÉMON's object is made transparent when it occludes NAVIGATION's arrow. This policy allows NAVIGATION to *induce* visual modifications in POKÉMON's objects by placing arrows behind them. A declarative approach can also enable new conflicts — for example, the OS may deny an app's request to display content if it cannot determine an acceptable layout that would accommodate this request without causing occlusion.

As another cautionary example, consider a least-recently-used (LRU) mechanism that identifies overlapping objects and removes those that the user has interacted with least recently. When applied as a runtime policy or declarative output tool, an LRU mechanism enables even well-intentioned apps to inadvertently evict each other. Furthermore, a malicious app could leverage an LRU runtime policy to probe for the locations of other apps' objects by observing when its *own* objects are evicted, using this information to surround a victim app's objects and occlude them.

Limitation: Conflict Identification. A limitation of any OS-driven approach is that the OS may not be able to unilaterally decide which visual interactions are problematic. If the OS can determine a prioritization ordering for different apps, it can potentially decide which apps to act upon when mediating conflicts, whether it employs runtime policies, declarative output, or another strategy. However, the notion of what constitutes a conflict may not always be obvious, nor the decision of which app should receive priority. Note that we previously explored the idea of OS-enforced runtime policies in prior work, described in Chapter 3 of this dissertation [63]. However, that work focused primarily on visual conflicts between AR objects and real-world objects, where the real world was assumed to take priority, and

it did not deeply consider the viability of runtime policies for resolving multi-app conflicts.

5.3.2.2 Application Self-Management

We next consider the potential for apps to collaborate in avoiding conflicts by sharing information with one another and reacting to each other's requests. For example, if Alice's NAVIGATION app could provide the 3D locations of its directional arrows to POKÉMON and request that POKÉMON not occlude them, then POKÉMON could adjust its behavior while still providing the user with the same overall experience.

Application self-management allows apps to retain control over their outputs and respond to conflicts in predictable ways, in contrast to OS-enforced policies that impose external modifications on app content. The consequence of giving apps this level of control is that self-management is only viable under a threat model where apps are trusted to avoid interfering with each other given the information to do so (e.g., on a closed platform running well-vetted apps that are designed to cooperate). A malicious app could leverage any additional information given to it about other apps to attack them — for example, if POKÉMON was malicious and learned precisely where NAVIGATION's arrows were, it could strategically generate objects that occlude those arrows.

5.3.2.3 User-Managed Output

Ultimately, the user may be best positioned to determine which conflicts are detrimental to their own AR experience. Thus, the final design path we explore is one that leaves mediation to the user's discretion. An AR platform could provide the user with different tools for this task — for example, to demote problematic apps to more restrictive states (e.g., confining them to windows), to delete individual AR objects, or to provide apps with behavioral cues (e.g., to instruct an app to avoid displaying content in specific spaces).

The OS also has an opportunity to inform the user's actions by enabling the user to easily discern potential conflicts. Recall Carol's COLLABORATIVE WORKSPACE app — Carol believed this app to be misbehaving, but the OS could inform her that the problematic object

came from another app. Furthermore, the user may be unaware that certain conflicts have actually occurred. For example, unbeknownst to Alice, her Tour Guide app displayed fake restaurant ratings that hid the overlays of Restaurant Assistant. The OS could identify such visual interactions and provide Alice with this information so that she can act according to her wishes.

5.3.3 Summary

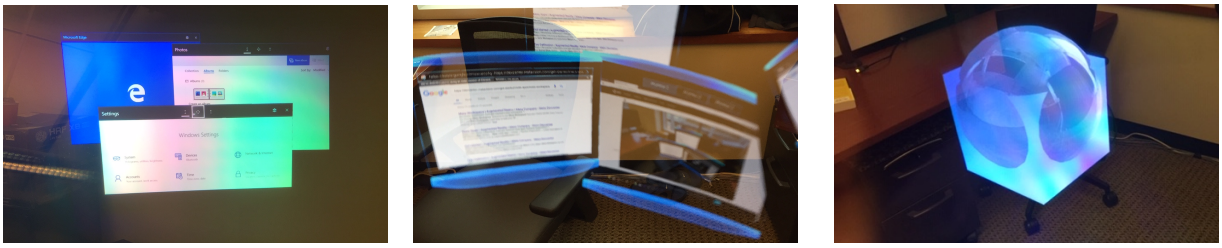
Identifying and mediating visual conflicts between AR apps is challenging, and different design strategies present varying trade-offs, as showcased in Figure 5.1. Our key insight is that any output mediation technique will infringe upon app functionality, and the precise nature of this infringement differs between design paths. Additionally, we observe that different techniques will be appropriate under different trust models, and our exploration highlights the potential for malicious apps to abuse well-intentioned capabilities.

5.4 AR Platform Analysis

In this section, we analyze the Microsoft HoloLens, Meta 2, and Magic Leap One AR headsets, asking: how do they fit into the broader design space above, and what unexplored directions may warrant further investigation? Each platform supports an immersive single-app mode that aligns with the first row of Figure 5.1, and we thus focus our analysis on the platforms' multi-app modes. Figure 5.2 depicts multi-app photos that we took through the lens of each device.

HoloLens. The HoloLens's multi-app mode supports Universal Windows Platform (UWP) apps, which run within 2D windows placed in 3D space by the user (Figure 5.2a). UWP apps run across different Microsoft platforms, providing a familiar interface for both users and developers. The window abstraction sacrifices support for immersive output to allow the HoloLens to enforce strong visual isolation between apps.

Meta 2. The Meta 2's multi-app mode is similar to that of the HoloLens, employing 2D



(a) Microsoft HoloLens (b) Meta 2 (c) Magic Leap One

Figure 5.2: **Multi-App AR Examples.** Multi-app photos from three AR headsets, taken with an iPhone 6 through the lens of each device.

windows placed in 3D space by the user (Figure 5.2b). The device tethers to a desktop PC and supports virtual "computer monitors" that enable the user to interact with their desktop's apps within AR windows.

Magic Leap One. By contrast, the Magic Leap One's multi-app mode supports multiple 3D apps at once. Apps may create "prisms" — bounded 3D regions in which they can display content. To probe the capabilities of prisms, we built multiple apps that display simple geometric shapes, and we ran two simultaneously. Figure 5.2c depicts two such apps: one displays a cube within a prism, and the other displays a sphere within a separate prism.

Prisms can be placed by the user, but we discovered that prisms from different apps are created atop each other by default. Apps can specify their prisms' sizes, but we could not determine if they can also control prism positions. If an app *can* control prism sizes and positions, then prisms act as a form of a shared world without conflict mediation mechanisms. As shown in Figure 5.2c, this design enables occlusion conflicts to occur. Furthermore, note that the cube and sphere are interleaved in 3D space, rather than one app receiving explicit rendering priority. Combining output from different apps in this way does not make intuitive sense from a user's perspective, suggesting that this occlusion is not intended behavior. We note that the Magic Leap developer guidelines suggest that prisms are *intended* to act as well-defined 3D windows, but this intention is not enforced by the platform.

5.5 Discussion

Our design exploration and analysis establish a foundation for understanding and addressing key multi-app AR challenges. Here, we identify promising avenues for future work.

Output Management Techniques. Our analysis reveals a nascent multi-app landscape among today's AR platforms. Critically, no platform provides a shared world abstraction endowed with additional conflict mediation capabilities. Of the mediation strategies captured in Figure 5.1, we believe that declarative output is the most compelling path for further exploration. A declarative approach can prevent output conflicts even with malicious apps, and it strikes a balance between app flexibility and conflict mediation. The OS can handle app requests in a more predictable manner than runtime policies allow, and apps can exercise more immersive behaviors than a windowed display abstraction supports. Furthermore, this approach does not impose the burden of output management on users. Even though declarative output cannot support apps that require arbitrary 3D placement, it is well-suited for apps tasked with augmenting specific real-world objects (e.g., Tour Guide and AR Dating).

Going forward, we propose that future work should validate the conceptual directions laid out in this work, by investigating the viability of declarative output (as well as the other above-mentioned output management techniques) in greater depth. One path would be to build a multi-application AR platform that supports different mediation strategies, and to evaluate these strategies along a number of axes — for example, the performance overheads that each technique imposes on applications; the ability of these techniques to effectively resolve output conflicts; the functionality limitations they place on application behaviors; and the burdens they place on both developers and users. Evaluating these criteria will better illuminate the trade-offs presented by different design paths, and will confirm (or contradict) our initial intuition regarding declarative output as the most promising path forward.

Non-Visual Output. While this work lays a foundation for addressing conflicts between

AR applications in terms of visual output, AR platforms may provide additional output modalities as well, such as aural or haptic feedback. Future work should investigate conflicts that may arise between AR apps in terms of non-visual output, determine if and where design strategies for preventing visual conflicts can be adapted to non-visual settings, and identify areas where new approaches will be required. Additionally, future work should consider opportunities for AR platforms to leverage combinations of multiple output modalities to mediate conflicts (e.g., by incorporating both aural and visual cues to help users contend with visual conflicts between apps).

Understanding User Perceptions of Conflict. Determining the visual interactions that users find problematic can inform defensive efforts, particularly for conflicts that cannot be fully prevented. For example, as suggested in Figure 5.1, no design can truly prevent masquerading, which depends upon users' perceptions of AR content. An AR platform can *attempt* to prevent masquerading, just as early windowing systems employed labeling techniques to indicate the origins of different windows (e.g., [35]). However, a user may still incorrectly perceive the origin of AR content. Future work is thus needed to identify design strategies that effectively engage the user and minimize the impacts of such conflicts.

5.6 Conclusions

Immersive multi-application AR platforms can enable users to interact with apps that simultaneously blend digital content into the physical world. However, AR apps may visually conflict with each other as they navigate the dynamically-changing environment of the user's world. In this work, we identify the challenges of mediating visual conflicts between apps without unduly infringing on their intended behaviors. We explore the design space of multi-app AR platforms and uncover key trade-offs presented by different design alternatives. We then analyze the design choices of current AR platforms and identify promising opportunities for future work. Our lessons lay a foundation to guide multi-application AR efforts, and we encourage future work to implement and evaluate the directions set forth in this chapter.

Chapter 6

CONCLUSION

Augmented reality is a powerful computing paradigm that is fundamentally changing how we interact with digital content in the context of the physical world. However, this exciting prospect comes with a cost: the capabilities that make AR so powerful also have the potential to expose users to new security and privacy risks. The preceding chapters of this dissertation identified important trends in the evolution of emerging AR technologies, deeply explored risks associated with these trends, and proposed technical solutions to protect users.

The first trend that I discussed was the evolution of immersive AR output capabilities. Rather than displaying content atop traditional computer screens, applications running on immersive AR devices like the HoloLens can more seamlessly integrate digital content into users' perceptions of the physical world. Consequently, immersive AR applications that are buggy, malicious, or compromised are in a unique position to harm users by modifying how they perceive the physical world. Chapter 3 of this dissertation characterized these types of output security risks, and it introduced Arya, a platform that my collaborators and I designed to mitigate such risks. Central to Arya is an output policy module that enforces behavioral constraints on the output of AR apps at the granularity of AR objects — new fine-grained primitives we designed to encapsulate properties of virtual content generated by apps. Through our prototype implementation and evaluation, we found that Arya balances flexible application behaviors with the ability to prevent undesirable output. In designing Arya, we identified numerous trade-offs involved with defining a policy specification framework and enforcing policies in practice. The conditional policy framework that we developed represents a promising first step, but there is significant room for further exploration. For example, other researchers have begun to explore more complex policy-specification strategies built

on Arya's foundation [2].

The second trend that this dissertation identified was a transition from purely single-user AR applications to multi-user experiences. The ability of users to interact with each other in shared AR worlds has the potential to enable new forms of collaboration, entertainment, and general engagement between people. Unfortunately, misbehaving or ill-intentioned users may attempt to abuse these capabilities to infringe upon the security and privacy of other users. Furthermore, since immersive AR technologies such as the HoloLens have only recently become available, we previously only had the ability to explore the security and privacy risks of these technologies from a conceptual standpoint. Through a qualitative user study, Chapter 4 of this dissertation presented an exploration of these risks grounded in the expectations, behaviors, and concerns of real AR users, with respect to both single- and multi-user experiences. Our findings shed light on numerous security and privacy concerns, such as deceptive AR content misleading users in the physical world, advertisers compromising users' privacy by continuously monitoring their environments, and a need for intuitive access controls that are tailored to shared AR environments to prevent unwanted interactions between users. Some of our findings reinforce defensive technical directions already being pursued by the AR security and privacy community, while others suggest opportunities for further work. For example, the findings described in Chapter 4 laid a foundation for my collaborators to develop novel content sharing techniques for multi-user AR [99].

The final trend, which I presented in Chapter 5, involves a shift from single-application AR platforms to platforms that allow users to engage with multiple, simultaneously-running apps. Despite the potential for multi-app AR platforms to empower users with rich multi-tasking capabilities, there remain numerous important challenges that stand in the way of realizing this vision. In particular, Chapter 5 discussed the challenges involved with preventing output conflicts between multiple apps that may interfere with each other (either intentionally or unintentionally) as they compete for visual real estate. We conducted a conceptual exploration into the design space of multi-app AR platforms, uncovering important trade-offs associated with different strategies. We then analyzed the multi-app capabilities

of several state-of-the-art AR headsets that are commercially available today, discovering a nascent multi-app landscape ripe for future exploration and experimentation.

Taken together, Chapters 3–5 identify and address multiple security and privacy challenges raised by emerging AR technologies. However, as AR technologies continue to evolve, they will inevitably raise new risks and create new opportunities for misuse. Thus, the aim of this dissertation is not just to identify and address the security and privacy challenges of today, but also to encourage future work to continue approaching AR technologies with a security- and privacy-driven mindset. This is in some ways an inherently reactionary process — while researchers and developers may attempt to anticipate and proactively address security and privacy issues that might arise (as this dissertation does), the emergence of new and possibly unforeseeable risks in the wild may alter any preconceived notions we previously held, which will require us to react accordingly.

While we cannot entirely predict the evolutionary trajectory of AR technologies or the security and privacy risks they may raise, this dissertation provides an informed basis for pursuing future avenues of work, and it discusses these opportunities throughout the preceding chapters. For example, Chapters 3 and 5 focused entirely on visual output, but future AR platforms may provide rich output through additional sensory pathways as well (e.g., aural or haptic feedback). Determining if output conflict mediation techniques tailored to visual feedback will suffice, or if we will need to develop new techniques specifically for non-visual output modalities, will be critical to keep users safe and secure in the face of evolving AR capabilities. Furthermore, we have barely begun to scratch the surface of multi-user interactions within shared AR experiences. Continuing to explore the perspectives of different user populations as AR technologies advance, and iterating on solutions for defining and constraining interactions, will be essential moving forward.

We stand today at a pivotal juncture with AR — these technologies are rapidly evolving, but it is still very early in their life cycles. Thus, now is the time to consider security and privacy for AR, while this computing paradigm is still young and designs are not yet set in stone. This dissertation lays a foundation for identifying security and privacy risks

that emerging AR technologies may raise, and for designing technical defenses to protect users from harm while balancing support for the rich application behaviors that make AR so powerful.

BIBLIOGRAPHY

[1] G. Ackerman and D. Bass. Israeli army prepares augmented reality for battlefield duty. https://www.bloomberg.com/news/articles/2016-08-15/microsoft-s-hololens-technology-adopted-by-israeli-military.

[2] S. Ahn, M. Gorlatova, P. Naghizadeh, M. Chiang, and P. Mittal. Adaptive fog-based output security for augmented reality. In *Morning Workshop on Virtual Reality and Augmented Reality Network*, 2018.

[3] J. Alexander. 'Ugandan Knuckles' is overtaking VRChat, 2018. https://www.polygon.com/2018/1/8/16863932/ugandan-knuckles-meme-vrchat.

[4] T. M. Andrews. Tweet that sent journalist Kurt Eichenwald into seizure considered 'deadly weapon' in indictment. Washington Post, Mar. 2017. https://www.washingtonpost.com/news/morning-mix/wp/2017/03/22/tweet-that-sent-journalist-kurt-eichenwald-into-seizure-considered-deadly-weapon-in-indictment/.

[5] https://developers.google.com/ar/.

[6] https://developer.apple.com/arkit/.

[7] https://developers.facebook.com/products/ar-studio/overview/.

[8] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE computer graphics and applications*, 21(6):34–47, 2001.

[9] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355–385, 1997.

[10] R. Baldwin. Mini's weird-looking AR goggles are actually useful, Apr. 2015. http://www.engadget.com/2015/04/22/bmw-mini-qualcomm-ar/.

[11] K. Bark, C. Tran, K. Fujimura, and V. Ng-Thow-Hing. Personal Navi: Benefits of an augmented reality navigational aid using a see-thru 3D volumetric HUD. In *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '14, New York, NY, USA, 2014. ACM.

[12] M. Billinghurst, A. Clark, and G. Lee. A survey of augmented reality. *Foundations and Trends in Human–Computer Interaction*, 8(2-3):73–272, 2015.

[13] M. Billinghurst and H. Kato. Collaborative mixed reality. In *Proceedings of the First International Symposium on Mixed Reality*, pages 261–284, 1999.

[14] M. Brocker and S. Checkoway. iSeeYou: Disabling the MacBook webcam indicator LED. In *USENIX Security Symposium*, pages 337–352, 2014.

[15] A. Butz, C. Beshers, and S. Feiner. Of vampire mirrors and privacy lamps: Privacy management in multi-user augmented environments. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*. ACM, 1998.

[16] A. Butz, T. Hollerer, S. Feiner, B. MacIntyre, and C. Beshers. Enveloping users and computers in a collaborative 3D augmented reality. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*. IEEE, 1999.

[17] R. Calo, T. Denning, B. Friedman, T. Kohno, L. Magassa, E. McReynolds, B. Newell, F. Roesner, and J. Woo. Augmented reality: A technology and policy primer. 2015.

[18] S. K. Card, T. P. Moran, and A. Newell. The model human processor- an engineering model of human performance. *Handbook of perception and human performance.*, 2:45–1, 1986.

[19] T. P. Caudell and D. W. Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on*, volume 2, pages 659–669. IEEE, 1992.

[20] K. Charmaz. *Constructing Grounded Theory*. Sage Publications Ltd, second edition, 2014.

[21] City of Portland. Trees & Visibility, Safety, & Clearance. `https://www.`

portlandoregon.gov/trees/article/424262.

[22] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 1960.

[23] Continental. Augmented reality head-up display. https://www.youtube.com/watch?v=3uuQSSnO7IE.

[24] L. F. Cranor. What do they indicate?: evaluating security and privacy indicators. *Interactions*, 13(3):45–47, 2006.

[25] L. D'Antoni, A. Dunn, S. Jana, T. Kohno, B. Livshits, D. Molnar, A. Moshchuk, E. Ofek, F. Roesner, S. Saponas, et al. Operating system support for augmented reality applications. *Hot Topics in Operating Systems (HotOS)*, 2013.

[26] A. Davis, M. Rubinstein, N. Wadhwa, G. J. Mysore, F. Durand, and W. T. Freeman. The visual microphone: passive recovery of sound from video. 2014.

[27] J. A. de Guzman, K. Thilakarathna, and A. Seneviratne. Security and privacy approaches in mixed reality: A literature survey. *arXiv preprint arXiv:1802.05797*, 2018.

[28] N. Dell, V. Vaidyanathan, I. Medhi, E. Cutrell, and W. Thies. Yours is better!: Participant response bias in HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012.

[29] T. Denning, Z. Dehlawi, and T. Kohno. In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, 2014.

[30] A. DeVincenzi, L. Yao, H. Ishii, and R. Raskar. Kinected conference: augmenting video imaging with calibrated depth and audio. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pages 621–624. ACM, 2011.

[31] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4), 2012.

[32] S. Egelman, R. Kannavara, and R. Chow. Is this thing on?: Crowdsourcing privacy indicators for ubiquitous sensing platforms. In *Proceedings of the 33rd Annual ACM*

Conference on Human Factors in Computing Systems, pages 1669–1678. ACM, 2015.

[33] S. Egelman, J. Tsai, L. F. Cranor, and A. Acquisti. Timing is everything?: the effects of timing and placement of online privacy indicators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 319–328. ACM, 2009.

[34] B. Ens, T. Grossman, F. Anderson, J. Matejka, and G. Fitzmaurice. Candid interaction: Revealing hidden mobile and wearable computing activities. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 467–476. ACM, 2015.

[35] J. Epstein, J. McHugh, R. Pascale, C. Martin, D. Rothnie, H. Orman, A. Marmor-Squires, M. Branstad, and B. Danner. Evolution of a trusted B3 window system prototype. In *IEEE Computer Society Symposium on Research in Security and Privacy*, 1992.

[36] L. S. Figueiredo, B. Livshits, D. Molnar, and M. Veanes. PrePose: Security and privacy for gesture-based programming. In *IEEE Symposium on Security and Privacy*, 2016.

[37] J. L. Fleiss, B. Levin, and M. C. Paik. *Statistical Methods for Rates and Proportions*. John Wiley & Sons, New York, 3 edition, 2003.

[38] B. Friedman and P. H. Kahn Jr. New directions: A value-sensitive design approach to augmented reality. In *Proceedings of DARE 2000 on designing augmented reality environments*, pages 163–164. ACM, 2000.

[39] T. M. Futurist. The Top 9 Augmented Reality Companies in Healthcare, 2017. `https://medicalfuturist.com/top-9-augmented-reality-companies-healthcare`.

[40] E. Gaebel, N. Zhang, W. Lou, and Y. T. Hou. Looks good to me: Authentication for augmented reality. In *Proceedings of the 6th International Workshop on Trustworthy Embedded Devices*. ACM, 2016.

[41] K. Gajos and D. S. Weld. SUPPLE: Automatically generating user interfaces. In *Proceedings of the 9th International Conference on Intelligent User Interface*, 2004.

[42] 60 FPS on Consoles. `http://www.giantbomb.com/60-fps-on-consoles/3015-3223/`.

[43] B. G. Glasser and A. L. Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Publishing Company, Chicago, 1967.

[44] https://www.theverge.com/2018/5/8/17332480/google-maps-augmented-reality-directions-walking-ar-street-view-personalized-recommendations-voting.

[45] S. Greenberg, S. Boring, J. Vermeulen, and J. Dostal. Dark Patterns in Proxemic Interactions: A Critical Perspective. In *Proceedings of the 2014 Conference on Designing Interactive Systems*, pages 523–532. ACM, 2014.

[46] G. Guest, A. Bunce, and L. Johnson. How many interviews are enough? An experiment with data saturation and variability. *Field Methods*, 18, 2006.

[47] R. Haeuslschmid, B. Pfleging, and F. Alt. A design space to support the development of windshield applications for the car. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5076–5091, 2016.

[48] HoloLens hardware details, 2017. https://developer.microsoft.com/en-us/windows/mixed-reality/hololens_hardware_details.

[49] A. Henrysson, M. Billinghurst, and M. Ollila. Face to face collaborative ar on mobile phones. In *Proceedings of the 4th IEEE/ACM international symposium on mixed and augmented reality*, pages 80–89. IEEE Computer Society, 2005.

[50] HoloLens, 2017. https://www.microsoft.com/microsoft-hololens/en-us.

[51] R. Hoyle, R. Templeman, S. Armes, D. Anthony, D. Crandall, and A. Kapadia. Privacy behaviors of lifeloggers using wearable cameras. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014.

[52] L.-S. Huang, A. Moshchuk, H. J. Wang, S. Schecter, and C. Jackson. Clickjacking: Attacks and defenses. In *21st USENIX Security Symposium*, 2012.

[53] Hyundai. Hyundai augmented reality demonstration - CES 2015. https://www.youtube.com/watch?v=iZg89ov75QQ.

[54] S. Irshad and D. R. A. Rambli. User experience evaluation of mobile AR services. In *Proceedings of the 12th International Conference on Advances in Mobile Computing*

and Multimedia, pages 119–126. ACM, 2014.

[55] S. Jana, D. Molnar, A. Moshchuk, A. M. Dunn, B. Livshits, H. J. Wang, and E. Ofek. Enabling fine-grained permissions for augmented reality applications with recognizers. In *USENIX Security*, 2013.

[56] S. Jana, A. Narayanan, and V. Shmatikov. A scanner darkly: Protecting user privacy from perceptual applications. In *IEEE Symposium on Security and Privacy*, 2013.

[57] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *2nd IEEE and ACM International Workshop on Augmented Reality, 1999 (IWAR'99)*, pages 85–94. IEEE, 1999.

[58] H. Kaufmann. Collaborative augmented reality in education. *Institute of Software Technology and Interactive Systems, Vienna University of Technology*, 2003.

[59] H. Kaufmann and D. Schmalstieg. Mathematics and geometry education with collaborative augmented reality. *Computers & Graphics*, 27(3), 2003.

[60] T. Kohno, J. Kollin, D. Molnar, and F. Roesner. Display leakage and transparent wearable displays: Investigation of risk, root causes, and defenses. Technical report, Microsoft Research, 2016.

[61] K. Lebeck, T. Kohno, and F. Roesner. How to safely augment reality: Challenges and directions. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, 2016.

[62] K. Lebeck, T. Kohno, and F. Roesner. Enabling multiple applications to simultaneously augment reality: Challenges and directions. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, 2019.

[63] K. Lebeck, K. Ruth, T. Kohno, and F. Roesner. Securing augmented reality output. In *IEEE Symposium on Security and Privacy*, 2017.

[64] K. Lebeck, K. Ruth, T. Kohno, and F. Roesner. Arya: Operating system support for securely augmenting reality. *IEEE Security & Privacy*, 16(1):44–53, 2018.

[65] K. Lebeck, K. Ruth, T. Kohno, and F. Roesner. Towards security and privacy for multi-user augmented reality: Foundations with end users. In *IEEE Symposium on*

Security and Privacy. IEEE, 2018.

[66] L. Lee, S. Egelman, J. H. Lee, and D. Wagner. Risk perceptions for wearable devices. *arXiv preprint arXiv:1504.05694*, 2015.

[67] X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li, and D. M. Gavrila. A new benchmark for vision-based cyclist detection. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016.

[68] B. MacIntyre, A. Hill, H. Rouzati, M. Gandy, and B. Davidson. The argon AR web browser and standards-based AR application environment. In *ISMAR*, 2011.

[69] `https://www.magicleap.com/magic-leap-one`.

[70] L. Matney. Jeff Koons' augmented reality Snapchat artwork gets 'vandalized', Oct. 2017. `https://techcrunch.com/2017/10/08/jeff-koons-augmented-reality-snapchat-artwork-gets-vandalized/`.

[71] M. May. Augmented reality in the car industry, Aug. 2015. `https://www.linkedin.com/pulse/augmented-reality-car-industry-melanie-may`.

[72] R. McPherson, S. Jana, and V. Shmatikov. No escape from reality: security and privacy of augmented reality browsers. In *Proceedings of the 24th International Conference on World Wide Web*, pages 743–753, 2015.

[73] Meta, 2017. `https://www.metavision.com/`.

[74] Microsoft. Designing for mixed reality. `https://developer.microsoft.com/en-us/windows/holographic/designing_for_mixed_reality`.

[75] Microsoft. HoloLens: Coordinate systems. `https://developer.microsoft.com/en-us/windows/holographic/coordinate_systems`.

[76] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *CoRR*, abs/1603.00831, 2016.

[77] M. R. Morris, A. Cassanego, A. Paepcke, T. Winograd, A. M. Piper, and A. Huang. Mediating group dynamics through tabletop interface design. *IEEE Computer Graphics and Applications*, 26(5), 2006.

[78] V. G. Motti and K. Caine. Users privacy concerns about wearables. In *International*

Conference on Financial Cryptography and Data Security, 2015.

[79] R. Naraine. Windows XP SP2 turns 'on' pop-up blocking, 2004. `http://www.internetnews.com/dev-news/article.php/3327991`.

[80] How virtual, augmented reality helps NASA explore space. `http://www.siliconvalley.com/2016/04/04/how-virtual-augmented-reality-helps-nasa-explore-space/`.

[81] Niantic. Pokemon Go. `https://www.pokemongo.com/`.

[82] P. Nowak. Heads-up: Driving is about to be revolutionized, 2017. `https://www.theglobeandmail.com/globe-drive/culture/technology/augmented-reality-merges-into-vehiclewindshields/article35096455/`.

[83] T. Ohshima, K. Satoh, H. Yamamoto, and H. Tamura. Ar2 hockey: A case study of collaborative augmented reality. vrais98: Proceedings of the virtual reality annual international symposium. *IEEE Computer Society, Washington, DC, USA*, (s 268), 1998.

[84] T. Olsson, P. Ihamäki, E. Lagerstam, L. Ventä-Olkkonen, and K. Väänänen-Vainio-Mattila. User expectations for mobile mixed reality services: an initial user study. In *European Conference on Cognitive Ergonomics: Designing beyond the Product—Understanding Activity and User Experience in Ubiquitous Environments*. VTT Technical Research Centre of Finland, 2009.

[85] T. Olsson, E. Lagerstam, T. Kärkkäinen, and K. Väänänen-Vainio-Mattila. Expected user experience of mobile augmented reality services: a user study in the context of shopping centres. *Personal and Ubiquitous Computing*, 17(2), 2013.

[86] T. Olsson and M. Salo. Narratives of satisfying and unsatisfying experiences of current mobile augmented reality applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012.

[87] G. Papagiannakis, G. Singh, and N. Magnenat-Thalmann. A survey of mobile and wireless technologies for augmented reality systems. *Computer Animation and Virtual Worlds*, 19(1):3–22, 2008.

[88] R. Raguram, A. M. White, D. Goswami, F. Monrose, and J.-M. Frahm. iSpy: automatic reconstruction of typed input from compromising reflections. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 527–536, 2011.

[89] P. A. Rauschnabel and Y. K. Ro. Augmented reality smart glasses: An investigation of technology acceptance drivers. *International Journal of Technology Marketing*, 11(2):123–148, 2016.

[90] N. Raval, A. Machanavajjhala, and L. P. Cox. Protecting visual secrets using adversarial nets. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1329–1332. IEEE, 2017.

[91] N. Raval, A. Srivastava, A. Razeen, K. Lebeck, A. Machanavajjhala, and L. P. Cox. What you mark is what apps see. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2016.

[92] H. T. Regenbrecht and M. T. Wagner. Interaction in a collaborative augmented reality environment. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*, pages 504–505. ACM, 2002.

[93] G. Reitmayr and D. Schmalstieg. *Collaborative augmented reality for outdoor navigation and information browsing*. na, 2004.

[94] F. Roesner, T. Denning, B. C. Newell, T. Kohno, and R. Calo. Augmented reality: hard problems of law and policy. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: adjunct publication*, pages 1283–1288. ACM, 2014.

[95] F. Roesner, T. Kohno, and D. Molnar. Security and privacy for augmented reality systems. *Communications of the ACM*, 57(4), 2014.

[96] F. Roesner, D. Molnar, A. Moshchuk, T. Kohno, and H. J. Wang. World-driven access control for continuous sensing. In *ACM Conference on Computer & Communications Security*, 2014.

[97] D. Rubino. Microsoft's Nadella weighs in on Pokémon Go, HoloLens, and the bright

future for AR. http://www.windowscentral.com/microsofts-nadella-weighs-pokemon-go-hololens.

[98] J. Russell. Pokemon Go has now crossed $1 billion in revenue, 2017. https://techcrunch.com/2017/02/01/report-pokemon-go-has-now-crossed-1-billion-in-revenue/.

[99] K. Ruth, T. Kohno, and F. Roesner. Secure multi-user content sharing for augmented reality applications. In *USENIX Security Symposium*, 2019.

[100] S. D. Scott, M. S. T. Carpendale, and K. M. Inkpen. Territoriality in collaborative tabletop workspaces. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*. ACM, 2004.

[101] I. Sluganovic, M. Serbec, A. Derek, and I. Martinovic. HoloPair: Securing shared augmented reality using microsoft hololens. In *Annual Computer Security Applications Conference (ACSAC) 2017*, 2017.

[102] S. Sridhar and V. Ng-Thow-Hing. Generation of virtual display surfaces for in-vehicle contextual augmented reality. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2012.

[103] I. E. Sutherland. The ultimate display. *Multimedia: From Wagner to virtual reality*, pages 506–508, 1965.

[104] I. E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968.

[105] J. E. Swan and J. L. Gabbard. Survey of user-based experimentation in augmented reality. In *Proceedings of 1st International Conference on Virtual Reality*, pages 1–9, 2005.

[106] J. Swearingen. How the camera doomed Google Glass, Jan. 2015. https://www.theatlantic.com/technology/archive/2015/01/how-the-camera-doomed-google-glass/384570/.

[107] Z. Szalavári, D. Schmalstieg, A. Fuhrmann, and M. Gervautz. Studierstube: An envi-

ronment for collaboration in augmented reality. *Virtual Reality*, 3(1), 1998.

[108] R. Templeman, M. Korayem, D. Crandall, and A. Kapadia. PlaceAvoider: Steering first-person cameras away from sensitive spaces. In *Network and Distributed System Security Symposium (NDSS)*, 2014.

[109] R. Templeman, Z. Rahman, D. Crandall, and A. Kapadia. Placeraider: Virtual theft in physical spaces with smartphones. *arXiv preprint arXiv:1209.5982*, 2012.

[110] C. Tran, K. Bark, and V. Ng-Thow-Hing. A left-turn driving aid using projected oncoming vehicle paths with augmented reality. In *5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 2013.

[111] J. Y. Tsai, S. Egelman, L. F. Cranor, and A. Acquisti. The impact of privacy indicators on search engine browsing patterns. In *SOUPS*. Citeseer, 2009.

[112] `https://unity3d.com/`.

[113] University of Iowa. The National Advanced Driving Simulator. `http://www.nads-sc.uiowa.edu/`.

[114] U.S. Department of Labor, Occupational Safety and Health Administration. Occupational Health and Safety Standards: Maintenance, safeguards, and operational features for exit routes. .

[115] U.S. Department of Transportation, National Highway Traffic Safety Administration. Visual-Manual NHTSA Driver Distraction Guidelines For In-Vehicle Electronic Devices (Docket No. NHTSA-2010-0053), 2010. `http://www.distraction.gov/downloads/pdfs/visual-manual-nhtsa-driver-distraction-guidelines-for-in-vehicle-electronic-devices.pdf`.

[116] D. Van Krevelen and R. Poelman. A survey of augmented reality technologies, applications and limitations. *International Journal of Virtual Reality*, 9(2):1, 2010.

[117] J. Vilk, A. Moshchuk, D. Molnar, B. Livshits, E. Ofek, C. Rossbach, H. J. Wang, and R. Gal. SurroundWeb: Mitigating privacy concerns in a 3D web browser. In *IEEE Symposium on Security and Privacy*, 2015.

[118] D. Wagner, T. Pintaric, F. Ledermann, and D. Schmalstieg. Towards massively multi-

user augmented reality on handheld devices. In *Pervasive*, volume 2005. Springer, 2005.

[119] K. Yeung. Microsoft partners with Autodesk to bring 3D product design to HoloLens, 2015. `https://venturebeat.com/2015/11/30/microsoft-partners-with-autodesk-to-bring-3d-product-design-to-hololens/`.

[120] E. Zarepour, M. Hosseini, S. S. Kanhere, and A. Sowmya. A context-based privacy preserving framework for wearable visual lifeloggers. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2016 IEEE International Conference on*, pages 1–4. IEEE, 2016.

[121] F. Zhou, H. B.-L. Duh, and M. Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 193–202. IEEE Computer Society, 2008.

Appendix A

USER STUDY PROTOCOL FROM CHAPTER 4

Below, I describe our protocol for the user study discussed in Chapter 4. I summarize each phase of the study in order and also provide our concrete semi-structured interview questions. We followed this protocol for every interview, only departing under two circumstances:

1. In some situations, if a participant said something particularly vague, we asked them to elaborate or unpack their thoughts more, before resuming with the script. We did not ask leading questions or guide participants towards specific opinions when asking for elaboration. Rather, we simply sought additional clarification on points that participants themselves had already raised.
2. If a participant began discussing thoughts related to an upcoming question in our interview protocol before being explicitly asked that question, we let them continue their train of thought rather than interrupting them. If appropriate in the flow of conversation, we would ask them the corresponding question from our script before resuming with the script as structured below. When this scenario occurred, it typically resulted in a minor re-ordering of General Experience questions within either phase 4 or phase 6. Occasionally, it resulted in the elevation of questions from phase 8 into phase 6.

We note that we specifically did not depart from the below protocol for phase 7 under any circumstances (Security, Privacy, and Other Concerns). To avoid prematurely priming participants to consider adversarial scenarios, we did not ask any questions from phase 7 until the conclusion of all previous phases, regardless of any previous thoughts participants may have discussed that were related to the topics covered in phase 7. Additionally, questions within phase 7 were presented in the same order for every participant pair.

Section 4.2.4 of this dissertation provides an overview of our study protocol. We provide additional details here, with the interview phases numbered below.

1. Overview Explanation. We began each interview by explaining the basics of augmented reality to participants (i.e., applications that overlay digital content directly on a user's perception of the physical world through some sort of device), explaining that they would be using an AR headset called the Microsoft HoloLens, and by providing an overview of our study (described below and also in Section 4.2.4), before providing participants with consent forms to sign if they wished to participate.

2. Interview: Initial Questions. We asked participants the following questions:

- What drew you to sign up for this study?
- Have you heard of AR before? If yes, what have you heard?
- Have you used any AR applications before?
  - Which ones?
  - On what devices?
- Have you seen other people using AR before? If so, where/when?
  - What about in fiction books, or in film?

3. Activity: HoloLens Tutorial + Shell. We next asked participants to go through the HoloLens tutorial, followed by using the HoloLens shell, as described in Section 4.2.4.

4. Interview: Initial Experience + Brainstorming. We asked the following questions, providing participants with a short period of time to gather their thoughts and take notes on paper after we asked each question, before verbally answering. For each interview phase, we ensured that both participants had an opportunity to speak.

- What do you generally think so far?
- What stood out to you the most?
- What did you like the most about what you've seen so far, or what seemed the "coolest"?

- What bothered you about your experience so far, or what did you find the most frustrating?
- Have you found anything particularly confusing or surprising so far?
- Did you expect that you were both seeing the same holograms?
  - Why or why not?
  - Would you have preferred one way or the other?
- Is there anything else you thought of that we didn't cover?

We then asked participants to think about what kinds of things augmented reality might be useful for, either now or in the future. We asked participants:

- What kinds of situations might you want to use AR in?
- What kinds of things would you want to be able to do with AR applications?

For the above 2 brainstorming questions, we had participants silently think and write down their individual thoughts for approximately a minute or two, after which we asked them to discuss their thoughts with us and with each other.

5. Activity: HoloLens Applications. As described in Section 4.2.4, we uniformly randomized the order in which each pair of participants used three HoloLens applications. Within a given pair, both participants used the same apps at the same times. In Section 4.2.4, we discuss our rationale for uniformly randomizing application order. For each app, we provided participants with basic initial instructions on how to use the app, after which we remained passive observers, only speaking in response to explicit questions from participants directed at us.

6. Interview: General Experience. We asked a similar set of questions as those immediately following the tutorial+shell phase, regarding participants' general experiences, now that they had experienced more HoloLens applications. For these and the below questions, as above, we gave participants a brief period of time to write down notes before verbally answering each question:

- What did you think of your experience overall?
- What stood out to you the most?
- What did you generally like, or what about your experience was the "coolest"?
- What generally bothered you about your experience, or what did you find the most frustrating?
- Did you find anything particularly confusing or surprising?

7. Interview: Security, Privacy, and Other Concerns. For the below topics (as discussed in Sections 4.2 and 4.3), we emphasize that while we prompted participants to consider a set of possible adversaries and scenarios, we did not mention any specific threats or concerns that might arise from these adversaries or within these scenarios.

Interview: Abuse of AR Technology. We asked participants the following questions:

- Now I want you to imagine you are someone that is trying to prank or troll someone else, like a sibling or friend, or maybe someone you really dislike. Let's say you're both using AR glasses like HoloLens, in a multi-user scenario like we talked about before. What kinds of things might you try to do to mess with the other person?
- Now imagine someone was trying to troll you, or make you have a really bad experience. What kinds of things would you be worried about them doing?

Interview: Untrusted Applications. We asked participants one question regarding applications downloaded from the Internet:

- Imagine you had downloaded some applications from the Internet for an AR headset. Is there anything that you might worry about those apps doing?

Interview: Bystanders. We asked participants a few questions about bystanders to AR technology:

- Imagine you are in public somewhere, like on the bus, on campus, or in a grocery store - think of places you typically go. If you saw someone wearing an AR headset in these types of situations, how would you feel? And what kinds of things would you think

the person is doing?

- Would your opinions change if the person is someone you know vs. a stranger?
- Would your opinions change if the person was in a more personal space, like your home, rather than in public?
- How would you feel if you weren't wearing an AR headset, but you were trying to talk or interact with someone who was wearing one?

8. Interview: Multi-User Experiences. Finally, we asked participants to consider multi-user AR experiences:

- You've seen different scenarios now; sometimes you could see the same holograms (like the multiplayer game) and sometimes you couldn't (like the robot shooting game). Did you prefer one over the other?
- Can you think of some scenarios where shared views might be more useful, or scenarios where private views might be more useful?