Enhancing Security and Privacy in Head-Mounted Augmented Reality Systems Using Eye Gaze

Matthew Corbett

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Application

Bo Ji, Chair
Brendan David-John
Wenjing Lou
David Raymond
Jiacheng Shang

March 27, 2024
Blacksburg, Virginia

Keywords: privacy, security, augmented reality, eye gaze tracking
Copyright 2024, Matthew Corbett

Enhancing Security and Privacy in Head-Mounted Augmented Reality Systems Using Eye Gaze

Matthew Corbett

(ABSTRACT)

Augmented Reality (AR) devices are set apart from other mobile devices by the immersive experience they offer. Specifically, head-mounted AR devices can accurately sense and understand their environment through an increasingly powerful array of sensors such as cameras, depth sensors, eye gaze trackers, microphones, and inertial sensors. The ability of these devices to collect this information presents both challenges and opportunities to improve existing security and privacy techniques in this domain. Specifically, eye gaze tracking is a ready-made capability to analyze user intent, emotions, and vulnerability, and as an input mechanism. However, modern AR devices lack systems to address their unique security and privacy issues. Problems such as lacking local pairing mechanisms usable while immersed in AR environments, bystander privacy protections, and the increased vulnerability to shoulder surfing while wearing AR devices all lack viable solutions. In this dissertation, I explore how readily available eye gaze sensor data can be used to improve existing methods for assuring information security and protecting the privacy of those near the device. My research has presented three new systems, BystandAR, ShouldAR, and GazePair that each leverage user eye gaze to improve security and privacy expectations in or with Augmented Reality. As these devices grow in power and number, such solutions are necessary to prevent perception failures that hindered earlier devices. The work in this dissertation is presented in the hope that these solutions can improve and expedite the adoption of these powerful and useful devices.

Enhancing Security and Privacy in Head-Mounted Augmented Reality Systems Using Eye Gaze

Matthew Corbett

(GENERAL AUDIENCE ABSTRACT)

Augmented Reality (AR) devices are set apart from other mobile devices by the immersive experience they offer. The ability of these devices to collect information presents challenges and opportunities to improve existing security and privacy techniques in this domain. In this dissertation, I explore how readily available eye gaze sensor data can be used to improve existing methods for assuring security and protecting the privacy of those near the device. My research has presented three new systems, BystandAR, ShouldAR, and GazePair that each leverage user eye gaze to improve security and privacy expectations in or with Augmented Reality. As these devices grow in power and number, such solutions are necessary to prevent perception failures that hindered earlier devices. The work in this dissertation is presented in the hope that these solutions can improve and expedite the adoption of these powerful and useful devices.

Dedication

This work is dedicated to my children, Reagan and Connor, who are the reason I had the strength to finish this work.

Acknowledgments

First, I would like to express my gratitude to Dr. Bo Ji, my Ph.D. Advisor. With his faith and mentorship, I found success and satisfaction in my research. Second, I want to thank Dr. Brendan David-John, Dr. Jiacheng Shang, Dr. David Raymond, and Dr. Wenjing Lou for serving as members of my committee. Their suggestions and feedback greatly improved this dissertation and the overall quality of my work. Finally, I want to thank my girlfriend Meghan and my children for their love, support, and unwavering faith in me. Without them, this journey would have ended long ago.

Contents

List of Figures xii

List of Tables xviii

1 Introduction 1

1.1 The Promise of Augmented Reality Devices 1

1.2 Head-Mounted AR Devices Create Unique Problems 2

1.2.1 Single-User Privacy Problems 2

1.2.2 Single-User Security Problems 4

1.2.3 Multi-User Problems 5

1.3 Summary of Contributions 5

1.4 Dissertation Organization 7

1.5 List of Publications 7

2 Review of Literature 9

2.1 Single-User Problems - Privacy 9

2.1.1 Explicit BPP Systems 10

2.1.2 Implicit BPP Systems 11

2.2 Single-User Problems - Security 12

2.2.1 Shoulder Surfing Attacks . . . 12
2.2.2 Passive Solutions . . . 13
2.2.3 Active Solutions . . . 14
2.3 Multi-User Problems . . . 15
2.3.1 Secure Pairing of Mobile Devices . . . 15
2.3.2 Pairing of AR Devices . . . 16

3 Insights on Human Eye Gaze **19**
3.1 Human Eye Gaze . . . 19
3.1.1 Eye Gaze Can Improve Single-User Bystander Privacy Solutions . . . 19
3.1.2 Eye Gaze Can Improve Shoulder Surfing Solutions . . . 20
3.1.3 Eye Gaze Can Improve Existing AR HMD Local Pairing Techniques 21
3.2 AR Devices and Built-In Eye Gaze Tracking Provides Opportunities for Novel Solutions . . . 22
3.2.1 Eye Tracking Can Improve Bystander Privacy in AR Devices . . . 24
3.2.2 AR HMDs and Eye Tracking Can Create Usable SSA Solutions . . . 24
3.2.3 Eye Tracking Provides Novel Methods For Local AR Pairing . . . 25

4 BystandAR: Protecting Bystander Visual Data in Augmented Reality Systems **26**
4.1 Introduction . . . 26

4.2 Background and Motivation . . . 30
4.2.1 The Bystander Privacy Problem . . . 30
4.2.2 Modern AR Devices Escalate the Bystander Privacy Problem . . . 30
4.3 Challenges . . . 31
4.3.1 Design Challenges . . . 32
4.4 BystandAR Design . . . 33
4.4.1 Threat Model and Our Approach . . . 34
4.4.2 High-Accuracy Bystander Detection using History Information . . . 35
4.4.3 On-Device Bystander Detection via Periodic Face Detection . . . 36
4.4.4 Frame Obscuration . . . 39
4.4.5 Putting It All Together - BystandAR Architecture . . . 39
4.5 Prototype Implementation . . . 40
4.6 Evaluation . . . 43
4.6.1 Evaluation Procedure . . . 43
4.6.2 Evaluation Metrics . . . 44
4.6.3 Effectiveness of BystandAR . . . 45
4.6.4 Comparison to Offline Solution . . . 47
4.6.5 Impact of Bystander Characteristics . . . 48
4.6.6 Effect of BystandAR on Bystander Perceptions of Privacy . . . 49
4.6.7 Overhead of BystandAR . . . 50

4.7 Discussion 52

4.8 Summary 55

5 ShouldAR: Detecting Shoulder Surfing Attacks Using Multimodal Eye Tracking and Augmented Reality 57

5.1 Introduction 57

5.2 Challenges 61

5.2.1 Challenges 62

5.3 ShouldAR Design 63

5.3.1 Threat Model 64

5.3.2 AR Headsets - A Feasible Platform for Generalized Anti-SSA Techniques 65

5.3.3 Design Components 65

5.4 Prototype Implementation 68

5.5 Evaluation 70

5.5.1 Evaluation Procedure 70

5.5.2 Understanding ShouldAR's Impact on Comfort in SSA-vulnerable Situations 75

5.5.3 Evaluating ShouldAR's Effectiveness 77

5.5.4 Evaluating ShouldAR's Usability 79

5.6 Discussion 79

5.7 Summary 82

6 GazePair: Efficient Pairing of Augmented Reality Devices Using Gaze Tracking **83**

6.1 Introduction . . . 83

6.1.1 The Importance of Local Pairing of AR Devices . . . 84

6.1.2 GazePair . . . 85

6.2 GazePair Design . . . 88

6.2.1 Threat Model . . . 88

6.2.2 Design Goals . . . 90

6.2.3 Key Challenges . . . 91

6.2.4 Pairing Process of GazePair . . . 93

6.3 Prototype Implementation . . . 96

6.3.1 Microsoft's Mixed Reality Toolkit . . . 97

6.3.2 Host/Client Establishment . . . 97

6.3.3 Gaze and Entropy Extraction . . . 100

6.3.4 Key Generation . . . 102

6.4 Experimental Evaluation . . . 105

6.4.1 Software and Hardware Setup . . . 105

6.4.2 Evaluation Design . . . 105

6.4.3 Pairing Performance . . . 107

6.4.4 Comparison with State-of-the-Art Research . . . 109

6.4.5 Usability . . . 110
6.4.6 The Usability Over Time . . . 112
6.4.7 GazePair vs. State-of-the-Art Solutions . . . 112
6.5 Discussion . . . 113
6.5.1 Availability of Gaze Tracking Sensor . . . 113
6.5.2 Length of the KSC . . . 114
6.5.3 Applicability to the Metaverse . . . 114
6.5.4 GazePair's Performance Under Threat . . . 115
6.5.5 Limitations of GazePair . . . 116
6.5.6 Future Work . . . 117
6.6 Summary . . . 117

7 Conclusion and Future Work **119**

7.1 Summary of Key Findings . . . 119
7.2 Future Work . . . 121
7.2.1 A Vision of the Future of Security and Privacy in AR . . . 121
7.2.2 The Future of Bystander Privacy in AR . . . 122
7.2.3 Shoulder Surfing . . . 124
7.3 Conclusion . . . 125

Bibliography **126**

List of Figures

1.1 An illustration of the sensor available on two modern AR devices: the Magic Leap 2 and the Microsoft HoloLens 2. 2

1.2 An illustration of a memory care use case for AR devices. In this example, a memory care patient requires assistance with identifying close friends and relatives. An application, using an AR device, identifies faces using facial recognition software and presents the names and/or titles of these persons to assist the user. In this example, the user is wearing the device and the subject is the person labeled as "Brother". The bystander, uninvolved with this interaction, is labeled by the red bounding box. Image created by DALL-E [4]. 3

4.1 An illustration of the medical use case of AR, where a nurse wearing an AR device is interacting with a patient while there are bystanders present (watching or not watching the nurse). In this situation, while the patient's medical record information needs to be presented to the nurse via the AR device, bystander information must be protected. 27

4.2 A malicious AR application is designed to request bystander visual data from a device, infer sensitive information from this data, and offload the inference results to another location for exploitation. BystandAR, shown between the device sensors and the malicious applications, is designed to prevent this. . 34

4.3 An illustration of the 2D-to-3D camera-to-world transformation process. Here, matrix M is an arbitrary transformation matrix created from the metadata included with each captured frame; it is used to convert 3D points in camera space to world space. 38

4.4 BystandAR Architecture. Raw data is captured from the device's sensors and is used both in face detection and learning eye gaze and voice history information for bystander detection. Afterward, bystander detection is used to obscure human faces not designated subjects in both camera and depth frame data. 40

4.5 An illustration of the output of BystandAR if a third-party application is requesting visual data. We blur the face of the subject only to protect the participant's identity. Note: the "boxes" over the face of the bystanders in the depth data in (b) reflect the depth of the bystander themselves in a plateauing manner. 42

4.6 BystandAR effectiveness across the two groups of eye gaze and voice thresholds. 45

4.7 Bystander protection rates of BystandAR over a 10-second average. The inset box highlights a drop in protection rates during large bystander movement, but overall performance is maintained for the majority of the data. 49

4.8 The distribution of the Likert responses for the bystander perception survey across the unmodified frames (Control Frames), the frames with example exploitation from DeepFace (Exploitation Examples), and the frames from BystandAR (Protected Frames). 50

5.1 An illustration of two scenarios of SSAs: the attacker is attempting to observe (a) sensitive information on a victim's laptop and (b) the written information of the victim. Images were generated using Microsoft Designer [139]. 58

5.2 ShouldAR Threat Model. Given the human eye's ability to sense motion at nearly 290° when allowed to move the head and eyes (with the torso stationary), we consider the remaining 70° to be the most vulnerable area for an SSA. Additionally, previous work shows that attacks on mobile devices and other common targets are most feasible inside 2 meters. We define the *vulnerable area* in red as the most likely area for an SSA. 63

5.3 ShouldAR Design. The design, shown from left to right, begins by harnessing the AR device's onboard sensors and an additional rear-facing camera. After this, in Step 1 (Data Collection), the raw data is collected for both the user's and the potential attacker's position and eye gaze. This data is analyzed in Step 2 (Context Understanding) to determine the user's gaze location and the potential attacker's facial and eye landmarks. In Step 3 (Threat Assessment), this context is analyzed, and the user is alerted if they are deemed vulnerable to an SSA in Step 4 (User Notification). 64

5.4 (a) shows an image of the ShouldAR prototype. A Magic Leap 2 headset was used as the AR platform, and the rear-facing camera, shown at the rear of the headset, was an ESP32-CAM and OV2640 wide-angle camera attached using Velcro fasteners. (b) provides an image of the baseline solution provided for participants. This two-inch fisheye convex mirror was used in both the typing and writing scenarios and was intended to simulate an ATM mirror. 69

5.5 An image of the testing environment. The red lines illustrate the distances and angles we evaluated inside the vulnerable area. The blue dots show an example of the ordering of the nine positions of the potential attacker. . . . 72

5.6 A comparison of the Likert responses of participants when asked about their comfort in the situations outlined in Table 5.1. The box plots in red are the questions asked assuming no protection. The blue box plots are the questions asked assuming protection from an active system such as ShouldAR. The dotted lines show the median values. Scenarios with an asterisk (*) indicate statistically significant results ($p < 0.05$). 74

5.7 A comparison of the Likert responses of participants when asked about their comfort in the situations outlined in Table 5.1, separated by participant gender. (a) is the responses from self-declared female participants. (b) is from self-declared males. The box plots in red are the questions asked assuming no protection. The blue box plots are the questions asked assuming protection from an active system such as ShouldAR. The dotted lines show the median values. Scenarios with an asterisk (*) indicate statistically significant results ($p < 0.05$). 76

5.8 A comparison of the Likert responses of participants when asked about their comfort in the situations outlined in Table 5.2. The dotted lines show the median values. **Note:** Questions 2 and 6 (marked with a †) were worded in such a way as to ensure the participant was carefully reading the questions. The Likert values for these questions were inverted in this plot for ease of comparison with the remaining questions. 80

6.1 If the attacker has access to the local network, as illustrated in (a), the attacker has the ability to intercept traffic at will and can understand the location of any holograms presented to each user. The attacker cannot, however, overhear or eavesdrop on any activity in the physical proximity of the users. If the attacker has access to the physical location of the pairing, as illustrated in (b), the attacker has access to the vocal communication between the pairing partners, but not the network traffic being passed amongst the pairing partners. Here, the attacker may have knowledge of what is said or done among the pairing partners, but does not have a way to use the passing network traffic to exploit this information. 89

6.2 An illustration of GazePair's three main steps to key generation and device pairing. 93

6.3 A depiction of the pairing protocol, using the classic Alice and Bob analogy. While there is only a single client (Bob) in this illustration, we show that GazePair is scalable to multiple clients easily. 94

6.4 An illustration of Steps 1 and 2 of GazePair. (a) shows the first scene where the host and client select their applicable roles and the pairing protocol establishes their relationships. (b) and (c) show the view of the pairing scene from both the host's and client's points of view, respectively. 98

6.5 An example of the eye gaze discretization technique used in GazePair. After a KSC is spoken to each client from the host, each user finds the required numerically-labeled hologram, gazes at it, signals the protocol to take a snapshot of the gaze location, and moves on to the next required hologram. Each gaze sampling is discretized and becomes a part of the shared secret. In this example, the first gaze sampling is taken when the user gazes at the hologram labeled "6". The raw gaze value is (-1.770, 2.398, 1.678), which is then discretized to "-222", based on the discretization parameters, and added to the shared secret. The length of this value will increase as the user takes more gaze samplings directed by the KSC. 99

6.6 A boxplot of the total pairing time, by the number of pairing partners. The median time is shown with the solid, horizontal, orange line. For one-to-one pairing, the median time is *8.7* seconds; for one-to-two pairing, the median time is *11.83* seconds. The average time for each is represented by the dashed, horizontal, green line. The average time for one-to-one pairings is 9.02 seconds; for one-to-two pairings, the average time is 12.58 seconds. 108

6.7 An illustration of the tradeoff between the length of the KSC, the time required to complete one-to-one pairings, and the generated entropy. Note the generally linear increase in pairing times relative to the KSC length. 109

6.8 The effect of multiple repetitions on the pairing time shown by average time and a linear regression of the time required for sequential tests. (a) shows the results of participants in one-to-one pairing. (b) shows the results of participants in one-to-two pairings. The generally downward trend in the pairing time indicates that as users begin to become familiar with the system and AR gestures, the system becomes more usable. 111

List of Tables

2.1 A comparison of current active solutions to SSAs. 14

3.1 A list of popular MR devices released since 2019 and their onboard sensing capabilities. 23

4.1 An evaluation of BystandAR against an offline bystander detection model. The protection rate was highest for BystandAR when images contain no subject and static bystanders, while the offline model performed marginally better with a moving bystander. 48

4.2 A summary of BystandAR's impact on the HoloLens 2's system resources, compared to device idle. 52

5.1 We asked each participant to assess their comfort in 6 scenarios as an initial estimation of their overall comfort in potentially vulnerable situations. . . . 71

5.2 Participants were asked to rate their agreement with the following statements or each question. A 5-point Likert scale was given, ranging from 1 (Strongly Disagree) to 5 (Strongly Agree). 73

5.3 A summary of the results of the ShouldAR quantitative evaluation. The "Baseline" column represents the tests conducted using the two-inch diameter fisheye mirror. The "Head Pose (HP)" column shows the tests using only the attacker's head pose, and not eye gaze. "HP w/ Eye Gaze" shows the results using both head pose and eye gaze. Each modality is shown subdivided by individual test tasks (e.g., typing or writing) and as a total average. 77

6.1 A comparison of GazePair with state-of-the-art AR device pairing solutions. 110

List of Abbreviations

AR Augmented Reality

BPP Bystander Privacy Problem

HMD Head-Mounted Device

MR Mixed Reality

SSA Shoulder Surfing Attack

VR Virtual Reality

AR technology seeks to overlay the real, physical world with synthetic, digital content. AR is a subset of Mixed Reality (MR).

The BPP refers to instances when data (images, video, audio, etc.) can be used to identify sensitive information (age, ethnicity, physical disability, emotion, etc.) is collected from bystanders who have not consented to be part of the data collection.

An HMD is a device designed to be worn primarily on the human head.

MR is the breadth of devices and displays between, but not including, the unaltered physical world and the completely virtual world to be Mixed Reality [140].

Shoulder surfing attacks (SSAs) are a type of observation attack designed to illicitly gather sensitive data from "over the shoulder" of victims.

VR is a technology that fully immerses the user in a synthetic, digital world.

Chapter 1

Introduction

1.1 The Promise of Augmented Reality Devices

Augmented Reality (AR) devices provide an experience "in which an otherwise real environment is 'augmented' by means of virtual (computer graphic) objects", or more simply, modify the physical reality with digital information [140]. These devices have seen a surge in popularity in recent years, spurred by technological advances, with an expected 200% increase in users and market value by 2027 [165]. This is partly driven by industrial, healthcare, automotive, and military applications, with AR devices creating advances in mental health research, military decision-making, and assisting students with disabilities [173, 193].

In particular, head-mounted devices (HMDs) create some of the most fertile ground for compelling AR applications, especially when considering the lighter, more usable devices of the future [60, 118]. These devices rely on an even more advanced suite of sensors like cameras, depth sensors, eye gaze sensors, and microphones. As an example, the Apple Vision Pro has more than 14 cameras of varying types to accurately absorb and understand the user's environment [52]. It is designed to be worn for long periods over multiple user case scenarios, in what Apple calls "Spatial Computing" [15]. This increasingly accurate understanding of the physical world allows for more compelling applications and content such as large-scale augmented workspaces or room-sized movie viewing. These devices also create unique opportunities for solutions to a host of problems using the inherent functionality of

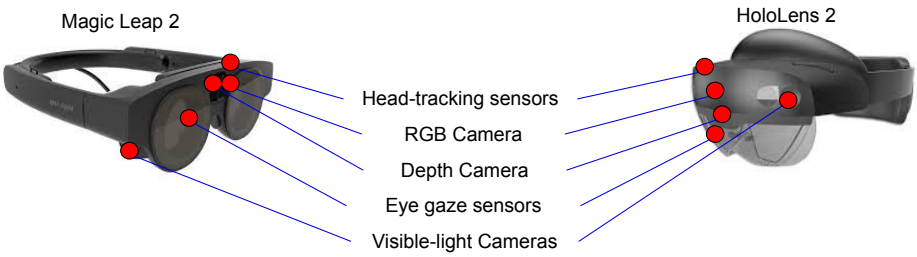


Figure 1.1: An illustration of the sensor available on two modern AR devices: the Magic Leap 2 and the Microsoft HoloLens 2.

AR devices. Figure 1.1 illustrates these sensors and their position on two modern AR HMDs.

1.2 Head-Mounted AR Devices Create Unique Problems

To understand the problems facing AR devices and the unique solutions that AR devices allow, we create a hierarchy of problems based on the number of users and their ability to collaborate.

1.2.1 Single-User Privacy Problems

Security and privacy issues plague even a single AR device. The device's sensors, while essential to the immersive experience that makes AR devices unique and powerful, do not discriminate in the data they collect. AR devices capture data required for well-intentioned tasks (e.g., SLAM, pose estimation, and gesture recognition), but also capture visual (e.g., camera and depth) data about bystanders (i.e., persons surrounding the device during its use), which can potentially be used to identify sensitive information (age, gender, emotion, gait, etc.) of bystanders for malicious purposes [27, 31, 32, 66, 80, 97]. This threat of

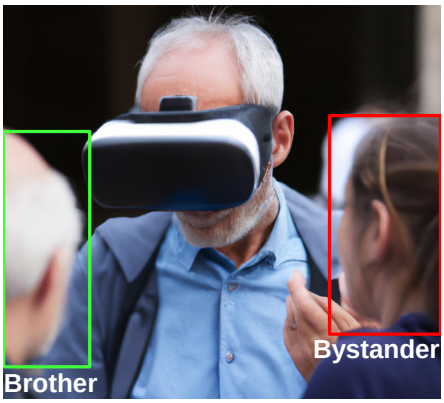


Figure 1.2: An illustration of a memory care use case for AR devices. In this example, a memory care patient requires assistance with identifying close friends and relatives. An application, using an AR device, identifies faces using facial recognition software and presents the names and/or titles of these persons to assist the user. In this example, the user is wearing the device and the subject is the person labeled as "Brother". The bystander, uninvolved with this interaction, is labeled by the red bounding box. Image created by DALL-E [4].

bystander data leak is called the *Bystander Privacy Problem* [44, 45, 156]. The *Bystander Privacy Problem* refers to instances when data (images, video, audio, etc.) that can be used to identify sensitive information (age, ethnicity, physical disability, emotion, etc.) is collected from bystanders who have not given consent to be part of the data collection. Through our research, we believe that this problem is comprised of two main components. The first is the technical vulnerabilities present in modern AR devices. These vulnerabilities stem from coarse-grained permissions for third-party applications and the ability of these applications to offload raw sensor data and violate the privacy of bystanders. This first component creates the conditions for the second, the perception that these devices can invade bystander or user privacy. This component does not have to be founded in technical reality (e.g., an actual privacy vulnerability) but has also been shown to hinder the success of MR devices. Each component can exist without the other, but we believe that the existence of technical vulnerabilities can exacerbate the perception issues that AR devices have suffered from previously.

Example scenarios where the bystander privacy problem can arise include medical treatment scenarios or assisting an Alzheimer's patient with memory care [72]. To make these scenarios more concrete, Fig. 1.2 presents an example. In this example, a patient in memory care is suffering from memory loss from Alzheimer's disease. The patient requires assistance to identify close friends and family. The application, on an AR device, uses facial recognition to identify and label persons of interest to the patient in real-time. However, identifying bystanders, such as other patients or family members of other patients, would violate privacy as their faces were used as part of the application without their consent.

1.2.2 Single-User Security Problems

Shoulder Surfing Attacks (SSAs) are observation attacks designed to compromise sensitive information (e.g., passwords and social security numbers) from "over the shoulder" of the victim [73]. The authentication information gained from these attacks can be used to conduct identity fraud or illegally access a victim's account data. Even if the attack is against other material, like private conversations or video chats, these attacks reveal the victim's sexual preferences and information on relationships or hobbies [58]. For these reasons, we consider this observation attack a security problem.

Head-mounted AR devices are designed to be used during normal interactions with the user's physical environment [60]. This can include when the user is observing sensitive content, including on laptops, desktops, mobile devices, or even while writing sensitive information (e.g., names and addresses) on paper. While SSAs are not unique to victims wearing AR HMD, (such problems exist with accessing sensitive information on cell phones, for example), the use of such a device can exacerbate the problem when the HMD-wearer is potentially distracted with other tasks and less aware of potential dangers [174]. Given these enhanced

dangers and the potential for far more ubiquitous use when these devices are lighter and less intrusive, the SSA problem in AR is without a readily available, viable solution [111].

1.2.3 Multi-User Problems

One of the expectations of an AR user may also be to share the augmentation of their physical reality with other AR users to create a collaborative and synergistic effect. Applications such as search and rescue, telepresence, gaming, and other collaborative AR tools may all require multiple users to exchange holograms or other information to facilitate collaboration [171]. Currently, for two users to pair their AR devices and securely share information such as holograms, the best working solution involves exchanging an alphanumeric authentication string or using centralized systems such as Microsoft Azure [122]. However, these solutions require an active internet connection and make potentially sensitive data available to third-party cloud providers (e.g., Microsoft). Local sharing solutions are required for scenarios without active internet or when transferring data to and from a central location (e.g., a cloud network) is not preferred.

This dissertation presents research that furthers the state-of-the-art in each of these problem domains. Each of the three following chapters explores a single solution to each problem. Below, we present a summary of this work.

1.3 Summary of Contributions

The work presented in this dissertation creates solutions to the problems listed above. These contributions can be summarized as follows:

1. **The Bystander Privacy Problem (BPP):** A first-of-its-kind solution, BystandAR,

was created to address the lack of real-time, on-device, effective solutions to the BPP in AR HMDs. This system exploits human eye gaze to distinguish subjects effectively (e.g., who the user interacts with) from bystanders (e.g., those unaware or unwilling to participate in the interaction). An evaluation of **BystandAR** was successful in protecting 98.14% of bystander faces while maintaining access to 96.27% of subject faces for legitimate application use (e.g., identifying friends and relatives). While no comparable solution exists, **BystandAR** improves on existing offline solutions by 12%. Additionally, **BystandAR** has minimal impact on AR HMD performance.

2. **Shoulder Surfing Attacks (SSAs):** **ShouldAR** is another first-of-its-kind solution designed to detect SSAs against AR HMD users. This wearable, mobile solution uses multimodal eye tracking to determine the location of the user's focus, the user's attention level (to determine vulnerability to attack), and the potential attacker's eye gaze. **ShouldAR** protects users against 87.28% of SSAs across *both* writing and typing scenarios. Additionally, the prototype was shown to improve user comfort in various attack scenarios. It was also shown to be usable, especially compared to an ATM mirror, another widely used SSA solution.

3. **Local Pairing of AR HMDs:** **GazePair** is a novel, high-entropy method for efficient and secure multi-AR HMD pairing. **GazePair** uses eye tracking and a spoken key to improve the scalability, efficiency, and security of existing techniques. **GazePair** averaged a 98.3% success rate with an average of 9.02 seconds to pair two devices. In a first-of-its-kind test, **GazePair** also proved capable of pairing three devices, with a 96.6% success rate and 12.58 seconds to complete the pairing.

1.4 Dissertation Organization

This dissertation is organized as follows: Chapter 2 reviews the existing work, specific to each of the problems discussed above. Chapter 3 gives a high-level overview of human eye gaze, psychology, and their utility as solutions to single and multi-user security and privacy problems in AR HMDs. Chapter 4 presents **BystandAR**, a first-of-its-kind solution to the single-user problem of bystander privacy. Chapter 5 presents **ShouldAR**, a solution to the single-user security problem of shoulder surfing. **ShouldAR** is a mobile, usable, and effective solution to detect SSAs, specifically against AR HMD users in a wide variety of scenarios. Chapter 6 presents **GazePair**, a multi-user local device pairing method. Chapter 7 provides a summary of contributions and potential future work.

1.5 List of Publications

Through solving the above problems and extending the limits of existing work, this dissertation and its related work have led to the following publications:

1. **Matthew Corbett**, Jiacheng Shang, and Bo Ji, "GazePair: Efficient Pairing of Augmented Reality Devices Using Gaze Tracking", IEEE Transactions on Mobile Computing, March 2023.

2. **Matthew Corbett**, Brendan David-John, Jiacheng Shang, Y. Charlie Hu, and Bo Ji. "BystandAR: Protecting Bystander Visual Data in Augmented Reality Systems", Proceedings of ACM MobiSys 2023, Helsinki, Finland, June 2023.

3. **Matthew Corbett**, Brendan David-John, Jiacheng Shang, Y. Charlie Hu, and Bo Ji. "Poster: BystandAR: Protecting Bystander Visual Data in Augmented Reality

Systems", Proceedings of ACM MobiSys 2023, Helsinki, Finland, June 2023.

4. **Matthew Corbett**, Brendan David-John, Jiacheng Shang, Y. Charlie Hu, and Bo Ji. "Securing Bystander Privacy in Mixed Reality While Protecting the User Experience", in IEEE Security & Privacy, vol. 22, no. 1, pp. 33-42, Jan.-Feb. 2024.

5. **Matthew Corbett**, "Enhancing Security and Privacy in Head-Mounted Augmented Reality Systems Using Eye Gaze", ACM SIGMETRICS Performance Evaluation Review, January 2024.

6. **Matthew Corbett**, Brendan David-John, Jiacheng Shang, and Bo Ji, "ShouldAR: Detecting Shoulder Surfing Attacks Using Multimodal Eye Tracking and Augmented Reality, submitted to USENIX Symposium on Usable Privacy and Security (SOUPS) 2024, Philadelphia, Pennsylvania, August 2024.

Chapter 2

Review of Literature

This chapter is designed to provide a general overview of the security and privacy problems, specifically the ones outlined in Chapter 1. We begin with a review of the relevant literature surrounding single-user privacy problems in AR in §2.1. Single-user security issues, specifically shoulder-surfing, are discussed in §2.2. Finally, §2.3 covers multi-user problems, specifically multi-device pairing.

2.1 Single-User Problems - Privacy

To the best of our knowledge, *there is no practical deployment of a bystander privacy protection (BPP) system for head-worn AR devices.* Hence, we discuss existing works on tackling the bystander privacy problem in mobile and wearable devices in general, including works that consider social media photo-sharing and compare their advantages and shortcomings.

There is a large body of existing BPP systems for mobile and wearable devices in general. These systems can be divided into two categories based on whether the input the system requires is *explicit* or *implicit*. *Explicit BPP systems* require the device user or the bystander to interact with the system, through a published privacy policy, hand gestures, etc. In contrast, *implicit BPP systems* use natural actions that occur with human interaction with the system (e.g., eye gaze, voice communication, and physical distance) to differentiate between the bystander and the subject.

2.1.1 Explicit BPP Systems

Explicit BPP systems require either the bystander, the device user, or both to perform some explicit actions to ensure bystander privacy. Works such as[3, 103] require potential bystanders to upload photos of their faces to train a facial classifier or respond to a prompt on their mobile device after a nearby photo capture to achieve some measure of privacy. Systems such as [81, 185] require bystanders to have a pre-defined privacy policy and facial signature on a linked server and potentially require a device user to audit and validate the privacy filters applied to scenarios that the system deems sensitive. Solutions such as [79, 103, 157, 168, 190, 196] are software frameworks and GUIs, requiring special equipment to be worn by bystanders, or by the device user in the presence of bystanders. Methods such as [90, 164, 185] require explicit hand gestures or markings from the device user or bystanders to express their privacy preferences or to exclude certain objects from an image, again requiring a user/bystander to make a conscious decision to be included or not.

In general, by giving the bystander or the user control of the situation, these explicit BPP systems potentially provide the kind of assurances that allow AR devices to be more acceptable in public places [53]. However, requiring the user/bystander to perform explicit actions such as hand gestures or to wear special physical devices imposes a significant burden on the user/bystander, resulting in such required actions being overridden, ignored, or unused if the user/bystander is not paying enough attention. Furthermore, such systems often require the device to be connected to a server to transport raw data, which increases the exposure of the data to potential misuse that the system is designed to prevent [162].

2.1.2 Implicit BPP Systems

Implicit BPP systems seek to protect bystander privacy without requiring any explicit actions to be taken by the user or the bystander. Such systems generally use a machine learning model (e.g., a neural network) to detect bystanders in the images captured by the device's camera [40, 41, 42, 74]. Such inference models extract various types of information from the images, such as distance from the camera, the direction of eye gaze, emotion, and position in the frame, and use them to improve detection accuracy.

By omitting the need for explicit actions taken by the user or bystander, implicit BPP systems can be easily deployed even when the bystander is unaware of the potential for their information to be recorded and hence promise much wider adoption. However, such systems can potentially suffer poor accuracy in bystander detection when bystanders present themselves in unexpected ways in the captured images. For example, two key features extracted and used in the bystander detection model in such systems are the eye gaze direction of a person in the captured image and being closer to the center of the frame than other persons; if a person's eye gaze is toward the device user and/or near the center of the frame, it is likely to be interacting with the device user and hence unlikely to be a bystander. However, as shown in the doctor visit scenario in Fig. 4.1, a bystander in the background could be looking at or moving toward the nurse (i.e., the device user). In this case, existing BPP solutions utilizing the two aforementioned features would erroneously label the bystander as a subject.

2.2 Single-User Problems - Security

We begin with a brief review of existing work on the topic of SSA. While not exhaustive, we believe that the review is representative of the state-of-the-art. We divide this section into three parts: a discussion about the background of SSA, a review of passive solutions (which do not indicate ongoing attacks on the victim), and active solutions (which do).

2.2.1 Shoulder Surfing Attacks

A shoulder surfing attack (SSA) is a type of observation attack designed to gather sensitive information about the victim, such as authentication information or personally identifiable data. This type of attack occurs in public situations (e.g., public transportation or an office building) [93]. While most reported attacks are against mobile device users, attacks can be directed at a variety of targets. SSAs can be on mobile electronic devices (e.g., cell phones), PIN entry pads at an ATM [21], or sensitive written information like Social Security Numbers or home addresses [161]. These attacks allow the attacker to obtain sensitive information that can provide insight into the victim's hobbies, relationships, sexual preferences, finances, and login data [58]. Fig. 5.1 gives an illustration of two of these attack scenarios. These attacks are generally not noticed by the victim, with only 7% of attacks noticed in a controlled study involving attacks on mobile devices [58].

While this shows that SSAs are generally unnoticed, 90.2% of participants in a survey studying SSAs reported that they would feel "anger, unease, pressure, and embarrassment" upon noticing an SSA directed at their information [58].

While simply observing sensitive information over the shoulder of a person is not new, the vast majority of research on SSAs had been published since the announcement of the original

iPhone in 2007 [58]. Given the focus on mobile devices, existing techniques to address SSAs generally focus on protecting the users of mobile phones, although some solutions do exist to protect other attack surfaces, like ATM PIN pads. In many U.S. states, local regulations require the installation of a small fisheye convex mirror to allow rear-facing views when entering PINs and retrieving cash [101]. We divide existing solutions into two categories: passive and active mitigation techniques. Simply put, passive techniques rely on obfuscating or complicating the shoulder-surfing attacker's attempt to collect the victim's sensitive data, not detecting the attack while it is in progress. Active techniques seek to identify the attacker in real-time and alert the victim, giving them a chance to intervene, move, or otherwise address the situation. This technique gives the potential victim the chance to prevent the attack from happening.

2.2.2 Passive Solutions

Passive techniques are those that do not seek to identify the attacker or react to the attack; they merely seek to make the chances of the attacker's success lower. Existing passive SSA solutions can be divided into five categories, based on the methods used to defeat the potential attacker. (1) Eye-gaze-based techniques let users use their eye gaze to enter sensitive information, such as on an ATM PIN pad [46] or on a keyboard [96] and make the information more difficult to observe. Additional eye-gaze-based techniques use "eye gestures" or specific eye movements for authentication or data entry. (2) PIN-based methods such as [22, 48, 98, 155, 170] create PIN entry schemes designed to make it difficult for the attacker to observe, record, or guess the PIN. These techniques seek to reduce attack success by altering the PIN entry format and complicating the attacker's ability to observe. (3) Pattern, location, or graphic-based techniques such as [24, 28, 30, 57] seek to remove a uniquely identifying item such as a password or PIN and add in a pattern like a series of

Active SSA Solutions	Protection Domain	
	Digital	Physical
PrivacyScout [20]	✓	✗
iAlert [8]	✓	✗
Are They Actually Looking? [172]	✓	✗
ATM Mirrors [101]	✗	✓

Table 2.1: A comparison of current active solutions to SSAs.

pictures, or physical location only an authorized user would know. (4) Obscuration-based methods like [7, 29, 192] provide a physical barrier to SSAs (e.g., a screen protector) such that the information can only be viewed from directly in front of the device. Some techniques are designed for users without the ability to use their eyes, such as the visually impaired, and use vibration or Braille-based methods [10, 13]. Finally, (5) multimodal techniques such as [2, 87, 88] seek to combine eye gaze and physical motion to overcome the ability of attackers to track user eye gaze and simulate input. These techniques comprise the majority of existing solutions to SSAs, but they are not usable in scenarios such as protecting sensitive information within written documents.

2.2.3 Active Solutions

Active SSA solutions seek to detect an attack in real-time and to prevent the attack from ever occurring potentially. Table 2.1 outlines existing active solutions and separates them by their protection domain. Some techniques seek to alert a user to a potential attacker by identifying faces in the field-of-view of a front-facing camera and analyzing eye gaze [8, 20, 172]. Other techniques use mirrors, like the built-in protections at ATMs, to help the potential victim identify these faces [101, 146]. These techniques all rely on locating faces around the user and evaluating the distance of the face and/or the person's eye gaze to determine the attack likelihood. However, these solutions are limited by the field of view of the device's camera

or mirror. If the user is positioned in a way that blocks the camera or mirror's view of the attacker, it will be hard to detect the attack. Also, these solutions are limited to a single domain, such as protecting a PIN pad *or* a mobile device. With the potential exception of the ATM PIN Pad, no current active solution can detect or prevent attacks in other scenarios. Finally, these solutions are generally not mobile. An ATM mirror can protect the user at the ATM or in its vicinity, but cannot protect this person once they move away. Existing active solutions, like passive techniques, lack broad use cases.

Passive and active solutions are a convenient way to divide existing work but also are orthogonal to each other. Because passive solutions protect authentication and active solutions alert the user to SSAs in specific scenarios, we cannot compare them directly. We present these solutions not in direct comparison, but as an example of the lack of usability in existing work.

2.3 Multi-User Problems

We review the expanse of current work related to the topic of efficient AR device authentication and pairing and divide the body of work into three subtopics. These topics are: secure pairing of mobile devices (Section 2.3.1), pairing of AR devices (Section 2.3.2), and eye gaze in security applications (Section 3.1.3).

2.3.1 Secure Pairing of Mobile Devices

Numerous studies have been completed on how to intuitively and efficiently *pair* mobile devices. Works such as [94] explain that the primary challenge to pairing these devices is authentication, or the ability to verify who the pairing partner is and what the user expects it

to be. One method proposed and used widely is out-of-band (OOB) communication to verify the intended pairing partner. Such a method allows users to independently verify that the exchanged cryptographic material (e.g., the keys) has not been intercepted or manipulated by a malicious third party through MITM attacks. Various OOB channels, such as human physiology [104, 206], ambient signals and energy [83, 116], simultaneous tapping to attenuate the received signal strength [209], and device acceleration [208], all attempt to solve the MITM problem. While novel, none of these protocols incorporate the unique and powerful capabilities of current AR devices or are otherwise inapplicable to AR uses. These protocols either focus on pairing multiple devices worn or used by a single user or simply are inefficient in terms of user interaction and success rate. To further expand on this point, imagine trying to "shake" two AR headsets together to synchronize accelerometer data as in [208]. This would require users to remove their AR devices, if worn, something not desirable in constant, immersive AR environments. Other techniques, such as Apple's AirDrop [35] or Wi-Fi direct [9], are difficult to scale and are designed for one-to-one connections. None of these techniques are particularly well-suited to the pairing of AR devices, but they are useful to understand as we seek to create an intuitive AR-friendly pairing protocol.

2.3.2 Pairing of AR Devices

While the body of work on this specific topic is still relatively small, three known techniques exist. First, Looks Good to Me (LGTM) [63] is the first research done into using the inherent functionality of AR devices, such as the ability to physically see other users in an augmented space while wearing a device, to assist in secure pairing. LGTM uses facial recognition and wireless localization to authenticate shared keys and to pair AR devices. The wireless localization hardware has never been deployed to any actual AR devices and LGTM cannot be effectively evaluated against any proposed pairing solution, including GazePair. LGTM

also relies on facial recognition, tested using sample faces from the Yalefaces dataset without obstruction. Modern AR devices, such as the Microsoft HoloLens series of devices [124], partially obscure the human face, greatly frustrating this assumption. Even so, LGTM integrates AR-friendly technologies such as facial recognition and certainly breaks ground on efficient methods to pair AR devices.

HoloPair [188] proposes a system that generates a secret locally on one device, creates a public key, transmits the key to a pairing partner, and authenticates this partner's key through a system of OOB physical interaction. HoloPair requires two users to trace the outline of a hologram created from the shared keys for key validation, and wave as a way to prevent MITM attacks during the transmission of the keys. Additionally, as identified in follow-up works by the authors [189], HoloPair allows users to simply accept the hologram verification, regardless of accuracy, potentially compromising the security of the protocol. This would potentially be done as a way to "speed up" the pairing by users. Additionally, HoloPair makes no claims that its protocol is feasible for use in larger than one-to-one simultaneous pairing scenarios. Finally, given the tracing requirement, HoloPair is an AR-only system, as users must see each other. However, HoloPair's reported pairing times, on average, are between 8-9 seconds. This is certainly an efficiency goal that pairing solutions should aspire to meet or exceed.

The most current work, Tap-Pair [189], improves on HoloPair by removing some of the elaborate requirements for user interaction. Tap-Pair also requires two co-located users, both wearing AR devices, to "tap" on the same physical location using a head direction to select a location. The original HoloLens used as the basis for Tap-Pair did not have true eye gaze tracking, but only the ability to sense the direction of the user's head. This "tap" requires users to direct their head at the desired point. This action gives a potential eavesdropping attacker insight into the shared secret. For this reason, the authors do not

assume that an attacker can be located with legitimate users. Additionally, Tap-Pair does not allow for windows to be used as the physical location, limiting use cases. Unlike HoloPair, Tap-Pair claims that it is scalable to multiple users. However, the authors did not attempt to either implement or test such a solution, leaving its feasibility purely theoretical. Tap-Pair is also AR-specific, and users must see the physical environment. Still, Tap-Pair remains the most feasible and practical method for locally pairing two AR devices, specifically two (1st generation) HoloLens devices.

Chapter 3

Insights on Human Eye Gaze

As the name of the dissertation implies, human eye gaze is the underlying theme of each of the proposed single and multi-user privacy and security solutions. Below, we discuss eye gaze with a focus on its ability to improve on the problems presented in Chapter 1.

3.1 Human Eye Gaze

Human eye gaze can provide a wide array of information about human beings. Examples include estimates of information overload [18, 150], human attention [145], sexual orientation, attraction [26], mental health conditions [49] and more. Additionally, eye gaze can be an effective method for user input in digital devices [181]. Eye gaze can also provide insight into the actions of persons around the user. Specifically, eye gaze can indicate when others are done speaking [49] or when others are looking in our direction.

3.1.1 Eye Gaze Can Improve Single-User Bystander Privacy Solutions

A seminal work from the field of psychology shows that in interpersonal communication, a participant is expected to look at her partner more than 60% of the time, with a higher rate of 73% while the participant is listening to her partner speak [17]. Other works report

this rate to be as high as 88% during a conversation [202]. Interestingly, while speaking, the speaker's eye contact can drop to as low as 41% of the total conversation time. Conversely, for eye contact with strangers or persons with whom one does not intend to speak (who are typically bystanders in an interpersonal interaction), other works report an upper bound of 3.3 seconds before the eye contact becomes undesirable to the recipient [25].

From these psychological studies, we draw the following key insights. First, we can infer that *in an interpersonal interaction using immersive devices such as head-worn AR devices, eye contact of the device user with others can be a highly effective indicator or feature in distinguishing the subject from a bystander,* because psychologically, the user is likely to make frequent eye contact with the subject but unlikely to stare at strangers (typically bystanders) for extended periods. Second, while the device user's eye gaze may vary across different cultures (e.g., it may occasionally wander off the listener in certain cultures) [5], it is more likely to be directed at the listener while speaking. This suggests that *combining the information of the device user's eye gaze and voice can be a highly effective indicator of whether the person is the intended subject of the interaction or just a bystander.*

3.1.2 Eye Gaze Can Improve Shoulder Surfing Solutions

To define the common denominators of SSAs, we distill the SSA problem into its core components. An SSA is, generally, (1) an attacker using their eyes to observe sensitive information, and (2) the victim using their eyes to attend to and/or reveal sensitive content simultaneously. While not every situation involves the attacker or the victim's eyes, such as in situations where the potential victim has a visual impairment or the attacker uses electronic surveillance, *this definition covers the majority of attack scenarios* [10, 13, 21, 58, 92]. Additionally, this definition unifies nearly all attack targets, including observing the

victim's written information, ATM PIN, and information typed on laptops, desktops, and mobile devices.

Understanding the importance of the gaze and attention to SSAs, we then leverage works from the field of psychology to extract salient features of the eye that can aid in our efforts to find an effective solution. For attackers, we can use eye gaze to create a more fine-grained estimation of their target and match their eye gaze to that of the user. This technique can remove instances where the potential attacker's head may be directed at something innocuous, but their actual eye gaze is directed at sensitive information. Additionally, this prevents false positive indications from the opposite scenario.

For the user, eye gaze information can provide fine-grained details about where the user is looking, how hard the user is concentrating on a given task or object, the user's emotions, and more [1, 143, 147]. Assuming that increased concentration on a target, say while writing or typing a password, creates a decreased lack of awareness of the user's surroundings, we can also use this metric to gauge the potential vulnerability of the victim to SSAs. If we use a proper threshold for this concentration, we can then decide to alert the user to SSAs only at times when they are less capable of detecting the attack themselves. Works such as [143, 147, 175, 200] give us specific thresholds for fixation duration, the amount of time an eye dwells on a particular area, as metrics for concentration and attention. Thus, *eye gaze is a nearly universal component of SSAs.*

3.1.3 Eye Gaze Can Improve Existing AR HMD Local Pairing Techniques

While we know of no work using eye gaze to *pair* two or more MR devices, numerous works address using eye gaze to *authenticate* a single user. Systems such as [115] ensure

authentication by asking a VR user to manipulate a 3D cube, similar to a Rubik's cube, and enter a password based on number and color using the hand-held controllers and the user's gaze. Other examples such as [87] use a combination of eye gaze and user touchpad input for authentication. Regardless of the system, implementations such as these commonly cite two things: that eye gaze can be more secure, but also suffers from increased input times and error rates over more standard input methods. As an example, [47] shows that eye gaze input methods can be nearly 25-fold more time intensive than manually inputting a standard PIN on a keypad with failure rates of nearly 24%. These cautionary tales guide us toward a solution that maximizes the security qualities of eye gaze, but in a system that does not use intricate eye gestures and can tolerate minor miscalibration of an eye gaze sensor.

3.2 AR Devices and Built-In Eye Gaze Tracking Provides Opportunities for Novel Solutions

As stated and shown in Table 3.1, eye gaze trackers are present on many modern AR HMDs. According to a major manufacturer of eye-tracking devices, "Eye tracking is a sensor technology that can detect a person's presence and follow what they are looking at in real-time. The technology converts eye movements into a data stream that contains information such as pupil position, the gaze vector for each eye, and gaze point" [194]. The power of eye gaze, combined with its "no cost" availability on modern AR devices, gives an interesting and potentially novel approach to security and privacy problems in AR.

Device	Released	Environment Mapping	Cameras	Eye Tracking	Depth Cameras	Microphone
Microsoft HoloLens 2 [124]	2019	Yes	5	Yes	Yes	Yes
Google Glass Enterprise Edition 2 [67]	2019	No	1	No	No	Yes
Varjo XR-3 [199]	2020	Yes	2	Yes	Yes	No
Magic Leap 2 [99]	2022	Yes	5	Yes	Yes	Yes
Meta Quest Pro [120]	2022	Yes	5	Yes	Yes	Yes
Xreal Air [205]	2022	No	0	No	No	No
Meta Quest 3 [121]	2023	Yes	4	No	Yes	Yes
Apple Vision Pro [15]	2024	Yes	12	Yes	Yes	Yes

Table 3.1: A list of popular MR devices released since 2019 and their onboard sensing capabilities.

3.2.1 Eye Tracking Can Improve Bystander Privacy in AR Devices

Using the insights in § 3.1.1, it begins to become clear that we can use eye gaze to determine the salient people in a human-to-human conversation. Exploiting the availability of eye tracking in AR devices specifically, this insight can be operationalized and made into a viable solution. Other platforms, like head-mounted eye gaze trackers, do not have the computing power to conduct the necessary facial detection, tracking, or onboard processing required for a potential solution. *Eye tracking, specifically when integrated into AR HMDs gives the required information about the user's eye gaze, and also provides the processing power to detect human faces and match the user's eye gaze to actual people.* Finally, as noted in Chapter 1, the BPP is especially an AR problem, given its outsized sensor suite and potential for broad use cases.

3.2.2 AR HMDs and Eye Tracking Can Create Usable SSA Solutions

With the insights in SSAs from §3.1.2, we can use the integrated eye tracking in many AR devices to envision AR-based solutions to SSAs. We can also use the inherent functionality of AR headsets (e.g., the ability of the AR user to interact with the physical environment while wearing the device) to make the solution mobile and more usable. AR headsets are designed to allow the user to view the physical world with an overlay of digital content. This allows the user to interact with the physical world as they usually would. For example, users can access their bank accounts at an ATM, use a laptop or desktop, or write with pen and paper all while wearing an AR headset. While current AR headsets are wearable and mobile, their weight and battery life limit their utility. Future AR devices, however, are not expected to have this problem [111, 118]. As these headsets get smaller, lighter,

and last longer between charges, they may become as common as cell phones. Additionally, the integrated, onboard processors and graphics processing units (GPUs) give AR devices adequate computing power to accomplish the facial detection and gaze estimation tasks we require. Finally, assuming the AR-based system only announces an attack when detected and does not require the user to interact otherwise, the notification of the attack is the only expected cognitive load.

3.2.3 Eye Tracking Provides Novel Methods For Local AR Pairing

Given the solutions presented in Chapter 2, the local pairing of AR devices remains an open problem. Existing solutions are not feasible due to hardware requirements, time, or low entropy. *Using the insights in §3.1.3, we can exploit the relatively intuitive and efficient eye gaze input method for a more efficient solution.* Eye gaze is an analog input, allowing for as many input values as can be measured by the digital eye-tracking system. For example, a keypad or keyboard is limited in the entropy it can provide by the number of keys that can be presented to the user. Eye gaze measurements are limited only by the tracking system's enumeration of this gaze. Eye gaze can be quantified as single points, blinks, patterns, motion rate, and other methods. This gives us an intuitive way to harness the integrated power of eye tracking in AR devices.

With these insights in mind, we move toward solutions that capitalize on these advantages. The following three chapters will present a single solution to each problem outlined in Chapter 1. Each will harness the power of the human eye to solve these problems. Additionally, each chapter will individually outline the limitations of these solutions and present potential future work.

Chapter 4

BystandAR: Protecting Bystander Visual Data in Augmented Reality Systems

4.1 Introduction

Augmented Reality (AR) devices are expected to reach an estimated 1.7 billion users by 2024, expanding from 1 billion in 2022 [12]. This is driven in part by industrial, healthcare, automotive, and military applications, with AR devices creating advances in mental health research, military decision-making, and assisting students with disabilities [173, 193]. These applications rely on the unique capabilities of AR devices, namely the ability to understand the physical world, and seamlessly blend the physical world and the holographic, digital world. This ability to create a virtual mapping of a physical space through Simultaneous Localization and Mapping (SLAM), establish synthetic holographic contact, and sense user eye gaze and hand gestures, is made possible by the integrated and powerful suite of sensors on modern AR devices. These sensors include Visible Light Cameras (VLCs), depth sensors, eye-tracking sensors, embedded microphones, accelerometers, and more [183].

Such sensors, while essential to the immersive experience that makes AR devices unique and

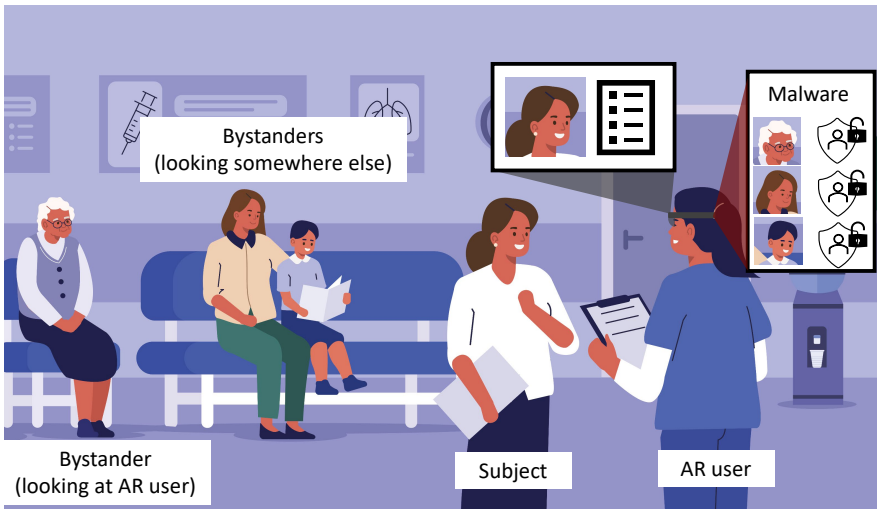


Figure 4.1: An illustration of the medical use case of AR, where a nurse wearing an AR device is interacting with a patient while there are bystanders present (watching or not watching the nurse). In this situation, while the patient's medical record information needs to be presented to the nurse via the AR device, bystander information must be protected.

powerful, do not discriminate in the data they collect. AR devices capture data required for well-intentioned tasks (e.g., SLAM, pose estimation, and gesture recognition), but also capture visual (e.g., camera and depth) data about bystanders (i.e., persons surrounding the device during its use), which can potentially be used to identify sensitive information (age, gender, emotion, gait, etc.) of bystanders for malicious purposes [27, 31, 32, 66, 80, 97]. This threat of bystander data leak is called the *bystander privacy problem* [44, 45, 156] (see definition in §4.2).

Example scenarios where the bystander privacy problem can arise include so-called *life-logging* or recording of visual evidence of everyday activities [59], assisting an Alzheimer's patient with memory care [72], and the AR functionality required to understand and interact with the physical world [163, 183]. Additionally, consider an AR-assisted medical service scenario, where an AR application is employed to assist nursing staff with triage at a hospital. The AR application may use facial recognition to present the nurse with a medical record chart for a patient, displayed in the AR device worn by the nurse. In a typical usage scenario

(as shown in Fig. 4.1), a patient may be accompanied by a family member who sits or stands near her during a visit, and there could be other nearby patients as well. In this situation, the AR application may identify the faces of the bystanders (i.e., her family member and other patients) and unintentionally retrieve and present their medical records instead of, or along with, that of the patient. This violates bystander privacy.

While there are very few studies addressing the bystander privacy problem specifically for AR devices [168, 190], there exists a large body of bystander privacy protection (BPP) systems for mobile and wearable devices in general. These systems can be divided into two categories based on whether the required input is *explicit* or *implicit*. *Explicit BPP systems* require the bystander or the user to possess an item, perform a gesture, or enroll in a system to expect privacy protection [3, 90, 103, 103, 157, 168, 185, 190, 196], which is not desired. In contrast, by using *implicit* information about the bystander (e.g., distance from the camera, the direction of eye gaze, emotion, and position in the frame) to determine if she is meant to be a part of the captured image. Also, *implicit BPP systems* seek to protect bystander privacy even when the bystander is unaware of the presence of the device [41, 74]. However, such systems can perform poorly when bystanders do not present themselves as expected in the captured image. In addition, existing implicit systems are implemented exclusively *off-device*, meaning that bystander data must be transferred to another device for processing. This opens an additional attack surface during the movement of this unprotected bystander data to a remote location [162].

Contributions. Due to the limitations of these existing BPP systems (which are designed for mobile and wearable devices in general, not specifically for AR devices), there is currently no practical BPP solution for head-worn AR devices. To that end, *we propose* **BYSTANDAR**, *the first practical BPP system that can effectively differentiate a subject from a bystander and protect bystander visual (camera and depth) data in real-time with only on-device processing*

that does not negatively impact the immersive applications AR devices offer. The main contributions of our work are as follows.

(1) From interesting studies in the field of psychology, we draw a useful insight that the device user's eye gaze and voice are highly effective indicators for subject/bystander determination in interpersonal interaction. Inspired by this key insight, we propose BystandAR, a novel bystander privacy protection system that uses the AR user's eye gaze and voice data to accurately determine the subject of interpersonal interaction, by leveraging novel AR capabilities such as eye gaze tracking and wearer-focused microphone array.

(2) Building such a system that exploits our key insight is highly nontrivial and entails two practical challenges: (C1) The device user's eye gaze may wander off the subject during the interaction; (C2) Performing the four basic tasks of bystander privacy protection (face detection, eye gaze tracking, subject identification via face-eye-gaze matching, and obscuration of bystanders) for every captured frame on-device is too costly and infeasible to keep up the frame rate.

To address these challenges, we leverage the inherent functionality of AR devices, namely spatial awareness and eye gaze tracking, to locate faces in 3D and monitor the user's interactions with the detected face, hence removing the need to infer the location of faces from every captured frame and maintaining a usable frame rate with only *on-device* processing.

(3) Through an evaluation involving 16 participants, BystandAR was successful in protecting 98.14% of bystander faces through obscuration and in identifying the subject of an AR interaction in 96.27% of output frames. This ensures that the visual data of identified subjects remain available for legitimate uses. Our evaluation also shows an improvement in bystander protection by 12% over the most accurate existing solution and shows a marked increase in bystander perceptions of privacy. These improvements are gained while keeping bystander

data *on-device*, removing the need to offload unprotected bystander data to another device, and maintaining frame rates as high as 52.6 frames per second (FPS).

4.2 Background and Motivation

In this section, we provide a brief background of the bystander privacy problem and explore the impact of modern AR devices.

4.2.1 The Bystander Privacy Problem

We first give some key definitions. An *AR user* (or simply *user*) is a person who wears an AR device; a *subject* is a person with whom the user intends to interact; a *bystander* is any non-user, non-subject third-party surrounding the device during its use. The *Bystander Privacy Problem* refers to when data (images, video, audio, etc.) that can be used to identify sensitive information (age, ethnicity, physical disability, emotion, etc.) is collected from bystanders who have not given consent to be part of the data collection [44, 45, 156]. Such bystander data leaks could also happen when well-meaning users, using AR applications for well-intentioned purposes, unintentionally violate bystander privacy.

4.2.2 Modern AR Devices Escalate the Bystander Privacy Problem

Surveys of bystander attitudes toward the presence of digital communication devices [154] were conducted as early as the year 2000, which shows negative perceptions of cellphone use (then novel) in public places. The survey participants reported that strangers felt "intruded

upon" when these devices breach the physical/digital barrier in their presence.

The rise of modern mobile devices, such as AR devices, opens the door to the development and growing adoption of novel immersive applications such as AR. These applications perform continuous sampling and recording of the physical world as part of their inherent operations, and in doing so, have propelled the bystander privacy problem into prominence. For example, a recent study focused on bystander perception of AR devices shows that bystanders are concerned with data collected by AR devices being used to uniquely identify them, and would prefer a way to require permission for the data to be recorded [51, 89]. Even medically assistive devices, including those designed to help persons with visual impairments, have been shown to elicit negative bystander feedback [6]. On the other hand, more recent work shows that bystanders are more willing to allow visual data to be captured if a blurring filter is added during capture; an additional 17.5% stated that they were willing over the un-blurred baseline [53]. Hence, there is a pressing need to develop a solution that can effectively protect bystander privacy without sacrificing the immersive experience AR devices offer.

4.3 Challenges

In this section, we introduce our key insight that the AR user's eye gaze along with her voice can be much better indicators in differentiating the subject and the bystander and discuss the main challenges in designing a bystander privacy-preserving system that exploits this key insight.

4.3.1 Design Challenges

A straightforward design of a bystander privacy protection (BPP) system that exploits the above key insights would simply perform the following four basic tasks for *every camera-captured frame*, to identify the subject and bystanders in the frame: (1) *Face detection*: identify all of the faces in the frame, e.g., using a state-of-the-art face detection model; (2) *Eye gaze tracking*: track the eye gaze of the device user during the frame interval; (3) *Subject identification via face-eye-gaze matching:* identify the face in the frame that the user's eye gaze intersects with, and label the face as the subject of the current interaction, and the remaining faces as bystanders; (4) *Obscuration of bystanders:* obscure the faces of the bystanders in the frame using masking or other techniques (blurring, avatar, silhouette, pixelating, etc.) and export the frame (e.g., recording it or sending it to a third-party application). However, such a straightforward design would not work well due to the following two practical challenges.

Challenge 1: The device user's eye gaze may wander off the subject during the interaction. As shown in numerous psychological studies discussed in §3.1.1, while the device user's eye gaze tends to remain on the subject's face during a personal interaction, it is not 100% of the time. This happens for two possible reasons. *First*, the user's eye gaze occasionally wandering away from the subject's face is a normal part of human conversations, expected as a way to signal the natural transitions in the conversation [50].

Second, the device user may need to refer to outside aids, such as maps or charts, as part of this interaction [202]. The consequence of this eye gaze wandering behavior is that simply relying on face-eye-gaze matching for each individual frame can misidentify a subject as a bystander (i.e., false negative) and a bystander as a subject (i.e., false positive).

Challenge 2: Performing the aforementioned four tasks for every frame on the

device is too costly and infeasible to keep up the frame rate. Face-eye-gaze matching (Task 3) and obscuration of bystander faces (Task 4) tend to be light-weight, and eye gaze tracking (Task 2) comes at almost no cost with hardware support (such as the Microsoft HoloLens 2 [132]). However, identifying faces in a frame (Task 1) generally requires performing face detection inferences, and high-accuracy face detection using deep neural network (DNN) models (e.g., DeepFace [180]) tends to be compute-intensive and incurs a long inference time when running on resource-constrained mobile devices.

For example, works such as [68, 71, 82, 108, 201] show that, while possible, on-device inference in resource-constrained mobile devices is generally difficult, and in this AR context, negatively impacts the frame rates that are directly correlated with user experience.[1] Even models specifically designed for use on mobile devices can limit frame rates to an unacceptable level [71, 107].

Therefore, we seek a practical solution that protects bystander privacy without compromising user experience by maintaining usable frame rates (e.g., near 60 FPS).

4.4 BystandAR Design

In this section, we discuss the threat model that our proposed BystandAR system is designed against, and how the key system components of BystandAR overcome the two main challenges discussed in §4.3.1.

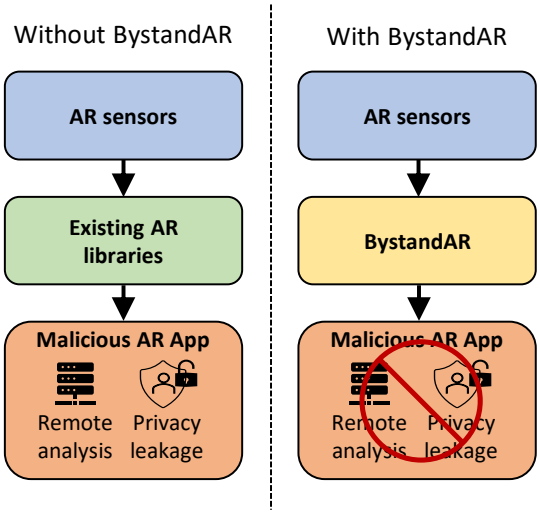


Figure 4.2: A malicious AR application is designed to request bystander visual data from a device, infer sensitive information from this data, and offload the inference results to another location for exploitation. BystandAR, shown between the device sensors and the malicious applications, is designed to prevent this.

4.4.1 Threat Model and Our Approach

BystandAR is designed to prevent malicious AR applications running on AR devices from collecting sensitive information from visual data of bystanders of interpersonal interactions during the execution of the AR application. Such malicious applications may perform *hidden operations* that extract sensitive information in bystander visual data captured during AR application execution [102].

As AR applications, these malicious applications can have full access to AR device cameras, microphones, and network stack after a cursory set of permissions requests, which will be granted by the device user [56, 62]. We note that, while BystandAR is designed to prevent malicious activities, the design prevents accidental bystander privacy leaks as well. Fig. 4.2 gives an illustration of this threat.

[1]Microsoft recommends that applications using their HoloLens 2 maintain a frame rate of 60 FPS in order to provide a positive user experience [125].

The high-level approach of BystandAR to overcome the above threat is to intercept the frame input operations of the application by (1) modifying the operating system's method of allowing access to raw visual data frames, (2) identifying the bystanders/subjects and obscuring bystander faces accordingly, and then (3) passing on the obscured frames to the application. Alternatively, one can implement BystandAR by requesting applications to use special APIs for reading visual data frames. These APIs, which are provided as libraries or a modified framework, implement the key tasks performed by BystandAR, noted above. During application installation, permissions will be given to these special APIs but not the regular APIs for reading visual data frames.

4.4.2 High-Accuracy Bystander Detection using History Information

Recall from Challenge 1 in §4.3.1 that we cannot simply match the location of detected faces in every frame with the user's eye gaze. Doing so would disregard the assumption that the user's eye gaze will wander as a natural part of human interaction. When this wandering gaze intersects with a bystander, this bystander could erroneously be labeled a subject and left unprotected in the output frame, violating bystander privacy.

We overcome this challenge by collecting data on the history of the user's gaze, as opposed to instantaneous information. We then use the historical information for different persons in these recent frames to determine who is the *subject(s)*. Specifically, the identification of the subject/bystander in the current frame is controlled by a threshold. Since we have two input modalities (i.e., eye gaze and voice), the threshold is two-fold. We set a threshold for purely eye gaze contact with a face and one for eye gaze and simultaneous voice contact. Since previous work suggests that eye and voice contact is more indicative of human attention [176,

202], we give the eye gaze threshold with voice a lower value. We do not provide for a voice-only threshold, as the user's voice alone does not identify a face.

These thresholds are the minimum total rate of eye/voice contact over the life of the detection and are informed by the works discussed in §3.1.1. These works give a range of eye gaze expected in conversations ranging from 41% to 73% and can be as low as 6% when the user is referring to maps, charts, or other visual aids. Additionally, when speaking to a person, eye contact is made less often than when listening to a person speak. Since BYSTANDAR has no way of knowing if the user is listening, we treat the user's eye gaze and voice as a more sure sign of a conversation than purely eye gaze and provide a lower threshold when voice is present.

Algorithm 1 shows the history-based bystander detection algorithm. First, BYSTANDAR uses the eye-tracking sensors and wearer-focused microphones present on nearly any modern AR device [113, 119, 130], to log the device user's eye gaze and voice data and establish context (Lines 5 and 6).

To determine the history of a user's contact with persons in the field of view, we also monitor the total amount of eye/voice contact with every detected face (Lines 7-14). Every face detection begins labeled a bystander. If over the life of the detection, the user has made enough contact with the detection to meet the threshold, the face is labeled a subject. BYSTANDAR also allows for labeling multiple subjects using this method. It is possible that the label of *subject* can revert to *bystander* if the threshold for contact is no longer met.

4.4.3 On-Device Bystander Detection via Periodic Face Detection

The history-based bystander detection method described above still assumes performing face detection on *every* camera frame. As noted in Challenge 2 in §4.3.1, performing face

detection on every frame is too costly on mobile devices and cannot keep up with the high frame rate (e.g., near 60 FPS) needed to support a high quality of user experience (QoE). To overcome this challenge, we explore how to avoid performing face detection on every frame.

If during an interpersonal interaction, the AR device does not move, consecutive frames captured by the camera would have the same spatial frame of reference, and we could simply skip every N frames for face detection in the above history-based algorithm and still be able to successfully match faces with the user's eye gaze and voice.

One challenge is that the history of eye-gaze and voice information needs to be accumulated for the same person across frames, but the faces (of bystanders or the subject) may move across frames. We thus need to keep track of their movement so we know faces in different frames correspond to the same person. This could be achieved by lightweight motion tracking techniques such as optical flow [142].

The above simple frame-skipping scheme, however, cannot handle the movement of the device itself. This can happen often in interpersonal interactions as the user moves her head and position and changes the spatial frame of reference of consecutive frames, which makes it much harder to match faces in different frames to the same person.

To tackle this challenge, we observe that one of the built-in capabilities of AR devices, SLAM, which is the foundation of the device's spatial awareness, can be used to track the location of a detected face when the device moves. We can exploit this unique AR device capability to compensate for the movement of the device.

To do this, we estimate the time required for a face to move outside of a face detection's bounding box (described in more detail in §4.7 (**BystandAR** Limitations)). From this, we can then estimate the number of subsequent frames N during which the faces will remain in the same bounding box, and skip face detection for these N frames. During the capture of these

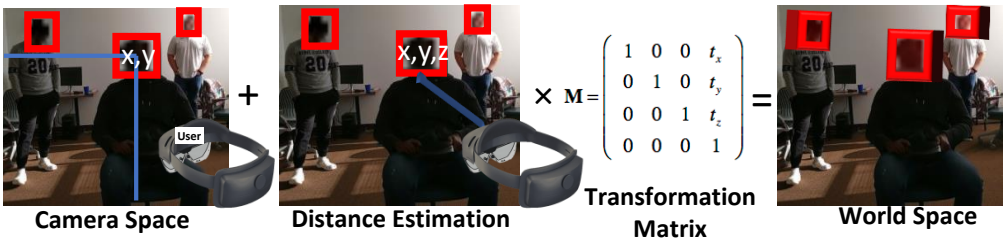


Figure 4.3: An illustration of the 2D-to-3D camera-to-world transformation process. Here, matrix M is an arbitrary transformation matrix created from the metadata included with each captured frame; it is used to convert 3D points in camera space to world space.

subsequent N frames, we assume that the face will remain inside its previous bounding box, removing the need for per-frame detection. In other words, we exploit a novel capability of AR devices to track faces over time, even with device movement, which removes the requirement to perform face detection more often than the device can support.

Tracking face movement using 3D-2D transformation. Using Algorithm 1, we explain this process as follows:

If a frame is selected for inference at the interval N, we capture the frame (Line 17) and infer the location of each face in the captured frame (Line 18). We must now locate each face detected in this frame using an absolute spatial reference, called a *world coordinate system* or *world space* in AR [138]. By doing so, we ensure that the face can be accounted for as the user moves, even when this motion causes the face to move completely out of the next camera frame. We can use the method provided by AR cameras to convert a 2D point (e.g., that of a face detected in the 2D frame) to the 3D world space (Line 20). If a face overlaps with a previous detection, indicating that the face has moved, we update the location of the face. In this way, we can track face movement (Lines 21-25). To prevent stale faces, if the face has not been updated in a given Time-to-Live (TTL) window, we remove it (Line 22). Fig. 4.3 presents an illustration of this process.

4.4.4 Frame Obscuration

Any usable solution to increase bystander expectations of privacy must also allow third-party applications access to any obscured frames. In the final step, BystandAR performs obscuration of the faces of detected bystanders in every frame and its corresponding depth frame, before passing them to the requesting AR applications (Lines 31 and 32).

Specifically, every frame is compared against existing, detected faces for potential obscuration to ensure bystander privacy as follows. As each new frame is captured, BystandAR compares existing facial detections and labels to the current frame. Using a system similar to the one we designed to create 3D locations from 2D detections, it then converts the 3D location of any detected face to a 2D position relative to the camera's current frame. If the detection has been labeled a subject and the user is making eye contact, we do nothing. If it has been labeled a bystander or is a subject not currently under the user's attention, we obscure it in both the camera and depth frame.

4.4.5 Putting It All Together - BystandAR Architecture

Fig. 4.4 shows the architecture of our proposed BystandAR system. The camera and depth frames are continuously captured by the AR device camera. At a given sampling interval, the face detection module infers the 2D location of any faces present in the frame, and BystandAR locates these faces in 3D after 2D-to-3D transformation. Using this location, we create a 3D bounding box, invisible to the user, that serves as the 3D anchor for each detection. By default, these faces are labeled bystanders. As sampled face detection continues and the position of the face changes, BystandAR updates the location of the face and moves the 3D bounding box accordingly. We provide a discussion about the speed of face movement BystandAR can tolerate in §4.7.

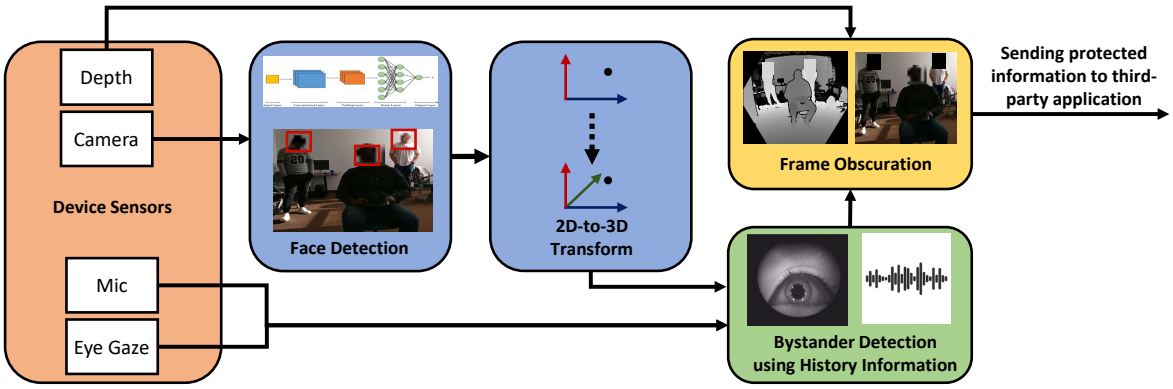


Figure 4.4: BYSTANDAR Architecture. Raw data is captured from the device's sensors and is used both in face detection and learning eye gaze and voice history information for bystander detection. Afterward, bystander detection is used to obscure human faces not designated subjects in both camera and depth frame data.

In parallel with the above face detection and tracking process, BYSTANDAR collects information about the user's eye gaze and voice using the AR device's onboard eye gaze tracking and wearer-focused microphone. For every camera frame, BYSTANDAR tracks on which face the user's attention is currently focused and maintains a history of this information for all currently detected faces. Once the history of the user's attention (eye gaze or simultaneous eye gaze and voice input) meets a pre-specified threshold, the detection is labeled a subject. With this context, the face obscuration module obscures the faces of each detection as required. After bystander visual data has been removed from each frame, the frame is safe for release to any third-party application.

4.5 Prototype Implementation

We built the BystandAR prototype in Unity 2021.3.11f1, using Microsoft's Mixed Reality Toolkit (MRTK) version 2.7.2, and we deployed BystandAR on a Microsoft HoloLens 2 device running Windows Holographic for Business Build 20348.1528. The core functionality

of BystandAR was implemented in about 1,200 lines of code. All source code was completed in C#.

We use Microsoft's *MediaCapture* class [136] to capture camera and depth frames, as well as collect the transformation matrices required for the 2D-to-3D detection conversion. The *FaceDetector* class of Microsoft's *FaceAnalysis* namespace is used for face detection [129]. This library provides accurate and tested functionality for determining the location of faces in 2D images, something we require to infer the 3D location of faces using spatial awareness. This face is represented by a Unity GameObject [198]. The onboard SLAM then continuously monitors the face in the physical world as the user moves. This functionality is not designed by us but comes "for free" with AR devices. We use a sampling rate of every *8* frames, informed by pilot testing and determined to be a good balance of accuracy and device resource load. This rate was the highest inference rate where we could expect 50+ FPS of frame rate during pilot testing. On a system with more computing power, this rate could be increased, yielding higher accuracy and similar frame rates. We collect camera frames with an input resolution of 1290×1080 pixels, a parameter determined in pilot testing to be a balance of inference accuracy and latency. We test our implementation at two different threshold levels. One, designed to test the higher limits of expected human eye/voice contact, sets a minimum of 50% pure eye contact or 25% simultaneous eye gaze and voice contact over the life of the detection. A lower threshold, 25% and 15%, respectively, was designed to test the lower bound of expected contact. These thresholds were created after studying literature on the dynamics of human eye contact in interpersonal interactions (see §3.1.1), and validated during our pilot testing. They are designed to serve as the highest and lowest values of the middle 50% of expected eye and voice contact between human being engaging in a conversation. However, these values can be altered for optimal use across different contexts.



(a) Camera Frame (b) Depth Frame

Figure 4.5: An illustration of the output of BystandAR if a third-party application is requesting visual data. We blur the face of the subject only to protect the participant's identity. Note: the "boxes" over the face of the bystanders in the depth data in (b) reflect the depth of the bystander themselves in a plateauing manner.

For the obscuration of the frame, we use a complete mask of the camera frame informed by the face detection's bounding box. The complete masking, as opposed to blurring, is informed by works (e.g., [207]) that show *deblurring* of such an image is indeed possible. Unique to the depth obscuration, we do not simply change the bit values of the raw depth data to create a mask, but seek to *plateau* the existing data to keep from creating "depth holes" in the image. Instead of completely masking the depth area, we smooth over the face of bystanders using the depth at the edges of the detection as a reference. This keeps from creating "depth holes" artificially. As noted in works such as [210], depth holes create inconsistencies that can hamper the AR user experience. If using the depth frame, knowledge of the presence of the face is still kept, but the details and potentially uniquely identifying contours of the face are removed. We show an example of the output of BystandAR in Fig. 4.5.

We use the HoloLens 2's onboard eye gaze tracking, wearer-focused microphone, and built-in spatial awareness to accomplish these tasks through the MRTK [135]. These APIs allow us to leverage the novel capabilities of AR devices in order to fully implement the BystandAR design.

4.6 Evaluation

Our evaluation consists of 16 participants. It is designed to not only test the effectiveness of BystandAR (i.e., the ability of BystandAR to differentiate the subject from a bystander) but also to evaluate performance when implemented on a Microsoft HoloLens 2. We evaluate the effectiveness of BystandAR by evaluating the success rate, defined as the amount of correctly obscured faces compared to the total number of detected faces corresponding to bystanders in every frame. We evaluate the performance of our prototype by measuring the frame rate, compared with the minimum frame rate for preserving the user experience recommended by the device manufacturer. Additionally, we measure the effect BystandAR has on bystander perceptions of privacy in the presence of AR devices with a post-testing survey.

4.6.1 Evaluation Procedure

The evaluation for BystandAR has been approved by our organization's Institutional Review Board. Sixteen participants were recruited using a graduate student mailing list and using distribution lists for undergraduate Human-Computer Interaction courses. All participants were 18 or older. Prior to testing, each user was given a 10-minute tutorial on AR gestures, specific to the Microsoft HoloLens 2, and given instructions on fitting and operating the device. Additionally, each user was instructed to complete an eye gaze calibration using HoloLens 2's calibration function. For evaluation, the prototype was designed to offload every 10th obscured frame. This is separate from the inference sampling interval and was used only to capture obfuscated data for evaluation.

The sixteen participants were grouped into eleven tests that contain one user and one to three bystanders. Every test contains a subject, except for the "no subject" test in §4.6.4. Among the eleven tests, two contain three bystanders, seven contain two bystanders, and two

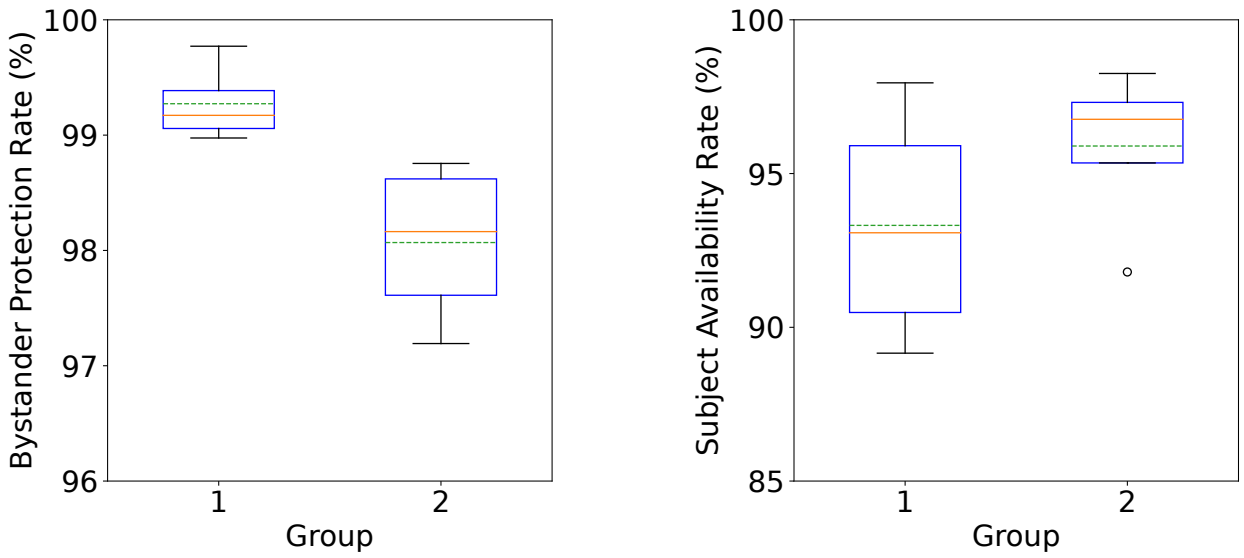
contain a single bystander; five contain the movement of bystanders, and six do not. During this test, the AR user was instructed to ask questions of the partner (i.e., subject) seated approximately 2 meters from them, for a total of 3 minutes. The AR user then swapped roles with their partner by giving them the AR headset and repeating the test.

We divided the data collection sessions into two groups to test two eye gaze and voice thresholds. Group 1 used a gaze threshold of 50% contact over the life of the face detection. Since we know that eye gaze and voice have a stronger correlation with the user's attention, we gave a lower threshold of 30% to eye gaze contact when the AR user is speaking. Group 2 had these thresholds set to 25% and 15%, respectively, to explore a shorter duration for subject detection.

4.6.2 Evaluation Metrics

For each obscured frame, we analyze the effectiveness of the bystander protection mechanism by using *DeepFace* [180], an open-source facial analysis tool, to evaluate BYSTANDAR's ability to identify subject/bystanders, and when necessary, protect them. Any bystander face found by DeepFace in the obscured frame indicated a failure of our system. In this experiment, we used a confidence threshold of 90% for DeepFace detections, as recommended by the model author [179]. We compute the total amount of obscured bystander faces compared to the total amount of detected bystander faces as the *bystander protection rate*.

We then analyze each frame to quantify the prototype's effectiveness in determining the subject of the interaction. Knowing the identity of the intended subject, we inspect every frame and record whether the face of the subject was unobscured. If the eye gaze of the AR device user was directed at the face of the subject, we expect the face of the subject to be unobscured. If the eye gaze is directed at the subject and the face remains obscured, we



(a) Bystander protection

(b) Subject availability

Figure 4.6: BystandAR effectiveness across the two groups of eye gaze and voice thresholds.

consider this a failure. We compute the total number of subject faces properly unobscured compared to the total number of subject faces as the *subject availability rate*.

Finally, in order to measure performance (in terms of frame rate), we separately calculate the FPS of the prototype as it runs on the Microsoft HoloLens 2. Each FPS calculation is done over a testing period of 3 minutes with the prototype obscuring bystanders in each frame in order to simulate a third-party application requesting visual data.

4.6.3 Effectiveness of BystandAR

Camera Data. We evaluate how well BystandAR, using eye gaze and voice data from the user, protects bystanders from DeepFace face detection on both camera and depth frames captured by the HoloLens 2's onboard camera.

We compare the two groups of eye gaze and simultaneous eye gaze and voice thresholds, Group 1 (50% gaze and 30% gaze/voice) and Group 2 (25% gaze and 15% gaze/voice),

to determine which is ideal for BYSTANDAR effectiveness. Fig. 4.6 illustrates the tradeoff in bystander privacy protection between the two groups. Lower thresholds result in lower rates of bystander privacy protection but a higher subject availability rate. Specifically, the impact of a lower threshold on bystander protection rate is marginal (about 1%), and bystander protection rates of both two threshold options were high (99.32% and 98.14% for Groups 1 and 2, respectively). In terms of subject availability rate, the lower thresholds can improve it by about 2.6%, from 93.63% to 96.27%. In our system, we use the lower thresholds (used in Group 2) as the default option since it provides a more balanced performance for both subject and bystanders.

Depth Data. BYSTANDAR also provides bystander protection on depth frames. For each frame, using the established face detections and their label as a "bystander" or a "subject", we obscure faces using a depth mask that is the same as the depth levels surrounding the detection. This provides the *plateau* effect mentioned in §4.4. While the HoloLens 2's depth data was insufficiently detailed to use existing depth-based face detection methods, we verified that every depth frame was obscured in the same way as its camera frame counterpart.

Threshold Recommendation. Between Group 1 and Group 2 thresholds, we achieved comparable bystander protection rates, with a 2.6% improvement in subject availability rates. For this reason, we recommend the lower threshold of Group 2. With this in mind, we conduct all following evaluations of BYSTANDAR using both gaze and simultaneous gaze and voice thresholds of 25% and 15%, respectively.

4.6.4 Comparison to Offline Solution

We compare the bystander protection rate between BystandAR running in real-time and a highly accurate offline bystander detection model [41]. The offline approach extracts features from each face and classifies it as bystander or subject with up to 94.3% accuracy using Gradient Boosted Decision Tree (GBDT). The extracted features capture the 3D head pose, the angle between the gaze direction and the camera, if the face was out of focus, and the distance from the camera. The offline approach processed every frame and could not feasibly be implemented on an AR device in real-time.

Visual data from three additional scenarios were collected for a robust comparison: a single bystander with no subject, a subject with static bystanders, and a subject with bystanders that include movement. The identified bystanders were positioned on either side of the subject and both in front of or behind them for 60 seconds at a time while camera frames from BystandAR were recorded. Raw frames were also recorded for input to the offline model. Each of the recording scenarios lasted for two minutes in total.

Table 4.1 presents the bystander protection rates between BystandAR and GBDT. Our BystandAR prototype performed better overall, with an overall rate of 94.1% compared to 82.3%. In the "No Subject" scenario, one individual was present but did not interact with the AR user. In this scenario, BystandAR had a perfect protection rate of 100%, while the GDBT model classified the face as a subject in every frame, producing a protection rate of 0%. BystandAR detects the subject based on eye gaze and voice interaction and does not suffer any false classification as a result. Accurate bystander recognition is important in the absence of a subject, as this is a typical scenario for AR users on a daily basis.

For scenarios that include a subject and multiple bystanders, both BystandAR and GBDT performed worse with a moving bystander. While BystandAR had a 2% higher protection

Table 4.1: An evaluation of BYSTANDAR against an offline bystander detection model. The protection rate was highest for BYSTANDAR when images contain no subject and static bystanders, while the offline model performed marginally better with a moving bystander.

Scenario	**GBDT[41] Protection Rate**	BYSTANDAR **Protection Rate**
No Subject	0%	100%
Static Bystanders	95.3%	97.3%
Moving Bystander	91.6%	88.3%
Overall	**82.3%**	**94.1%**

rate with static bystanders, GDBT had a 3.3% higher rate with a moving bystander. Rates from these two scenarios show that a moving bystander is more difficult to identify and obscure, and an offline model applied to every frame is only marginally better than BYSTANDAR running in real-time with an inference sampling interval of 8 frames.

4.6.5 Impact of Bystander Characteristics

As seen in Table 4.1, the effectiveness of BYSTANDAR can vary based on the characteristics of bystanders. We further evaluate the impact of different numbers of bystanders in the scene and where motion degrades protection rates.

The bystander protection rates across sessions with one, two, and three bystanders were computed from the procedure described in §4.6.1. We found average protection rates were the highest for three bystanders (99.26%), the median for two bystanders (98.74%), and the lowest for one bystander (98.30%). Overall, we found that increasing numbers of bystanders had a negligible impact on bystander protection.

Next, we evaluate BYSTANDAR results temporally based on the motion level of bystanders. We identified data collection sessions where the bystander(s) made large movements, defined as movements from one side of the user's FOV to the other. In the tests with this motion,

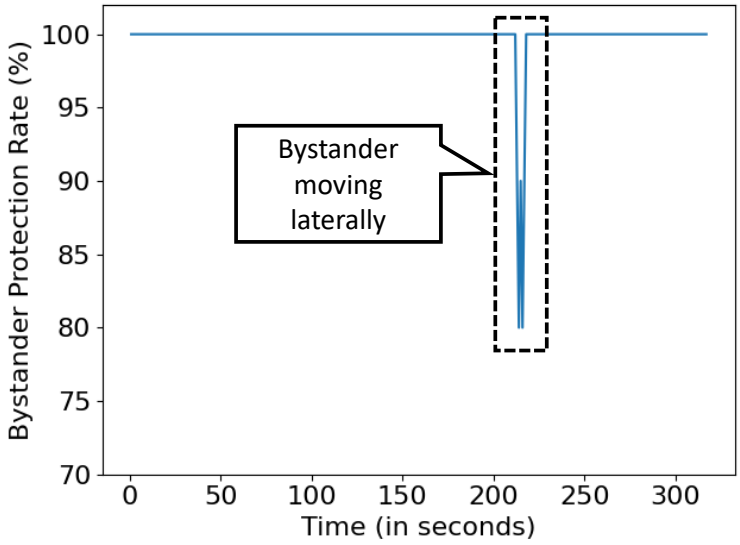


Figure 4.7: Bystander protection rates of BystandAR over a 10-second average. The inset box highlights a drop in protection rates during large bystander movement, but overall performance is maintained for the majority of the data.

BystandAR protected the visual data of bystanders in 98.91% of frames. For tests where the bystanders remained static, we found a protection rate of 98.77%. Additionally, Fig. 4.7 shows a single test run and the impact that dramatic bystander movement had on the protection rate. Both the figure and the aggregate results show the minimal impact bystander movement had during our testing, but we further discuss situations where this result may differ in §4.7 (BystandAR Limitations).

4.6.6 Effect of BystandAR on Bystander Perceptions of Privacy

Following the completion of testing, each participant was asked to complete a brief survey designed to investigate the effect BystandAR has on bystander perception of privacy. Each respondent was first presented with a camera and a depth frame showing a subject, and two bystanders from a user's point of view. The respondent was asked to assume they were one of the two bystanders and was then asked to rate their comfort level with this information

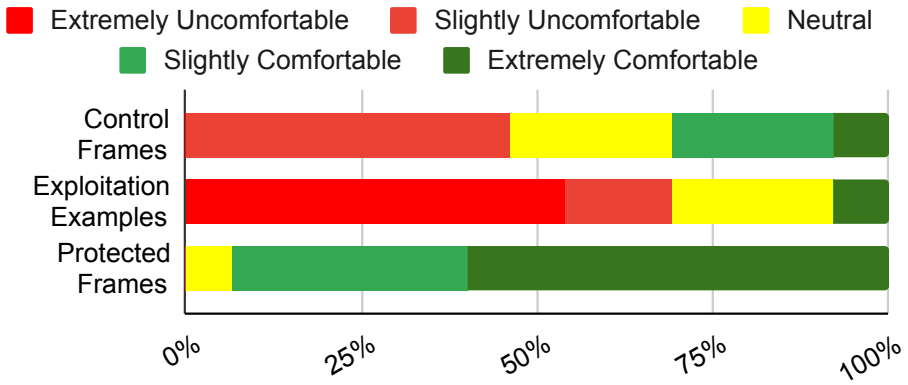


Figure 4.8: The distribution of the Likert responses for the bystander perception survey across the unmodified frames (Control Frames), the frames with example exploitation from DeepFace (Exploitation Examples), and the frames from BystandAR (Protected Frames).

being made available to a third-party application on a Likert scale of one to five with one being "Extremely Uncomfortable" and five being "Extremely Comfortable".

The respondents were then presented with two additional sets of images, one using added DeepFace inference results, and the other after **BYSTANDAR** obscured the bystander faces in the camera and depth frame. For the protected set, respondents were told that the image was protected from exploitation like DeepFace inference, but also to assume this system was less than perfect. They were then asked to rate their comfort level for each. Fig. 4.8 shows the aggregated results of this survey. We further discuss these results in §4.7 (Bystander Perceptions).

4.6.7 Overhead of BYSTANDAR

Using an inference sampling interval of 8 frames, **BYSTANDAR** runs at **52.6 FPS** while not being required to obscure frames for release to a third-party application. In fact, this method is likely to be the most widely used if the device is not running an application that requires camera or depth frames. In this mode, **BYSTANDAR** still collects raw images and obscures

faces according to the bystander/subject detection described in §4.4. It does not, however, apply any masks to any output images. This frame rate is comparable to the HoloLens 2's recommended 60 FPS (when not capturing media frames) as suggested by Microsoft [125]. When the prototype is configured to release obscured frames, BystandAR achieves **33.6 FPS**. Such a drop in frame rate is expected for our prototype, as Microsoft's standard sensor data logging API states that frame rates will drop to around 30 FPS [137]. We note here that BystandAR would switch between offloading frames and not based on whether a third-party application was requesting them.

To stress test our system, we also ran tests using an inference sampling interval of 1 frame, meaning every captured camera frame was used for inference. While not obscuring frames (i.e., simulating running while no third-party application is requesting frames), the prototype runs at 15.2 FPS; while obscuring frames, the prototype runs at 11.5 FPS. This significant drop shows the problems with per-frame inference as stated in §4.3 as both values are well below any thresholds recommended by Microsoft for usable frame rates as a result of inferring on every frame.

Finally, Table 4.2 shows a breakdown of BystandAR impacts on the system resources of the HoloLens 2, as compared to the device's idle load. As a third-party application, BystandAR increases the CPU load on the HoloLens 2 by 27%, with minimal increases in memory footprint and power consumption. Also, given the 7% increase in power utilization, we project a roughly 12-15 minute decrease in the 2-3 hour expected battery life of the Microsoft HoloLens 2 [132]. We believe that BystandAR would need to be implemented at the OS level, as discussed in §4.7, reducing these requirements.

Table 4.2: A summary of BYSTANDAR's impact on the HoloLens 2's system resources, compared to device idle.

Total System Load	**Average CPU Usage**	**Average GPU Usage**	**System Memory**	**Power Util.**
w/BYSTANDAR	72%	0%	2.8 GB	68%
System Idle	45%	0%	2.2 GB	61%

4.7 Discussion

Scope. BYSTANDAR is designed to protect bystanders (and make subjects available) in interpersonal interactions. Scenarios such as jogging, where the user is not making meaningful contact and moving quickly would be less advantageous for our system. However, the impact of the speed of the user or bystander's movement can also be mitigated, as discussed later in this section.

Bystander Perceptions. As shown in Fig. 4.8, BYSTANDAR increases bystander confidence in the protection of their visual information in the presence of AR devices. Even when told that such a system may not be completely successful in all use case scenarios, participants were generally more comfortable with a third-party application having access to their visual data, assuming that a system such as BYSTANDAR had protected their privacy first. We believe that systems such as BYSTANDAR are even more effective at increasing perceptions of privacy when bystanders are familiar with the potential threats as well as the protection provided.

OS-level Implementation. In the current demonstration prototype, BYSTANDAR is implemented as a third-party application running on the Microsoft HoloLens 2's Windows Holographic OS. On the HoloLens 2, only one AR application is allowed to run at a time. BYSTANDAR cannot intercept and obscure frames as a sole third-party application. An assumption stated in multiple areas of this work is that, if ever implemented on a production

system, BystandAR would need to be implemented at the OS level. This could add challenges, such as how and when to update inference models used by core OS processes, as was required during the COVID-19 pandemic [33]. This also provides some advantages for efficiency, as toolkits and APIs such as MRTK [135] might not be necessary. We do not investigate this further but leave this for future work.

Performance in Multi-user AR scenarios. As part of our testing design, we also sought to evaluate the effectiveness of BystandAR if the subject was also an AR device user. For this, we designed a second test that uses Microsoft's Azure Spatial Anchors [122] to share an absolute understanding of a physical space. The users were required to collaborate by building a block structure while the application synchronized the location of their blocks on a cloud game server created by Photon Unity Networking [159]. The chosen model, FaceDetector [129], proved to be unreliable when detecting the faces of "subjects" wearing AR devices but just as reliable for device-less bystanders as expected. Models such as these can be retrained to identify faces with AR devices, similar to the case in the recent pandemic [33], but we believe that training models specifically for this purpose is beyond the scope of our work.

BystandAR Limitations. BystandAR is built around the idea that per-frame inference is not necessary to create a reliable system. This is grounded in the fact that face movement, like movement in all media capture, is replicated with a series of frames giving the illusion of actual movement. Faces, people, things, etc., are not "moving" in videos, they are merely shifting in relative position across every frame. We believe that, if the inference rate is fast enough, the bystander/subject face cannot move outside of the obscuration between inferences. However, this does have a limit. For example, at a frame rate of 52 FPS and an inference sampling interval of 8, like BystandAR, an inference occurs about every 154 milliseconds. Given an average face width of 0.15 meters [133], a face at the center of an

obscuration box must move 0.15 meters in 154 milliseconds to evade the box and be fully revealed, which leads to the following speed: $\frac{0.15 \text{ meters}}{0.154 \text{ seconds}} = 0.97$ m/s. Given this speed, a face could elude the inference rate and be unprotected. Even in testing, of the small number of bystander faces exposed, about 50% were due to movement. This can be ameliorated through more rapid inference at the cost of lower frame rates. More advanced and powerful AR devices in the future can allow more rapid inference with less (if any) frame rate degradation.

BystandAR is designed to work in a variety of different interpersonal situations. For instance, we asked bystanders to stand, sit, move, or remain still during the various tests conducted. However, we never asked a bystander to move in front of the subject during testing. As the bystander's face would be recognized as a face that had not yet met the thresholds of eye/voice contact to be labeled a "subject", this face would be obscured in any output image. If this face was directly in front of the subject, BystandAR would obscure both faces and temporarily render the subject unavailable in the image for as long as this occurred. Additionally, BystandAR works on the principle that only one subject should be revealed at once. While BystandAR is capable of classifying more than one face as a subject, we only reveal the face of the subject when the user's eye gaze intersects with the subject's face. This could be altered to allow both subjects' faces.

Another limitation comes when we consider a malicious user. BystandAR is built with the natural dynamics of human eye and voice contact in mind. Given the social barriers to staring at persons not part of an interaction, shown in §4.3, we also believe that the risk of users unintentionally staring at the exposed face of a bystander is possible but limited. However, if the user intends to expose the face of a bystander without them being part of an AR interaction, they can certainly choose to keep eye contact on an exposed face to prevent obscuration by the system. This would override the assumptions made about normal human eye and voice contact. It would also be noticeable to the bystander and other parties that

a person not involved in an interaction or conversation was staring at a seemingly random person. We present BystandAR not as the perfect solution, but rather as an example of an on-device, context-enabled method to further bystander protection in AR systems.

4.8 Summary

In this work, we harnessed the dynamics of human interaction to improve bystander visual data protection in AR devices by creating a novel system called BystandAR. This is achieved *on-device* while maintaining usable frame rates on AR devices. We believe that this work expands the understanding of the capability of modern AR devices to protect bystander privacy and to further the trust of bystanders that their privacy is protected, using unique capabilities that only these exciting, advanced AR devices possess.

Algorithm 1 BystandAR Control Loop

```
Parameters: sampling interval N
FrameCounter = 0
while True do
    Increment FrameCounter
    Compute the current location of the eye gaze
    Monitor if voice input is above noise floor
    if Eye-gaze/Voice intersects with a face then
        Increment eye/voice tracker for face
        if Eye-gaze/Voice history > Threshold then
            Label face a subject
        else
            Label becomes/remains bystander
        end if
    end if
    if FrameCounter ≥ N then
        FrameCounter = 0
        Retrieve raw depth and camera frames
        Infer location of all faces in frame
        for each Face detected do
            Transform 2D detection to 3D world space
            if face overlaps with an existing face then
                Replace current detection; reset TTL
            else
                Create new detection
            end if
            if Application requesting sensor data then
                Obscure bystander faces in frame
            end if
        end for
    else
        if Application requesting sensor data then
            Obscure bystander faces in frame
        end if
    end if
    Release frames to application
end while
```

Chapter 5

ShouldAR: Detecting Shoulder Surfing Attacks Using Multimodal Eye Tracking and Augmented Reality

5.1 Introduction

Shoulder surfing attacks (SSAs) are a type of observation attack designed to gather sensitive information (e.g., passwords, social security numbers, or home addresses) from the victim by directly observing this information without the victim's knowledge. This is generally done "over the shoulder" of the victim in a way that does not allow the victim to notice this attack, as illustrated in Fig. 5.1. Published work on this topic is plentiful, with 8,420+ works related to "shoulder surfing" listed on a recent (February 2024) Google Scholar search [70], including 3600 being published since 2017 [58].

The vast majority of shoulder-surfing research has focused on reducing shoulder surfing *during authentication* (e.g., while entering passwords) through advanced authentication schemes or other obfuscation methods. To prove this point, 70 of the 92 works published in the last year focused on improved authentication schemes [148]. These works require biometrics, graphical passwords, or other advanced input mechanisms to prevent an attacker from see-



(a) SSA on a laptop

(b) SSA on written text

Figure 5.1: An illustration of two scenarios of SSAs: the attacker is attempting to observe (a) sensitive information on a victim's laptop and (b) the written information of the victim. Images were generated using Microsoft Designer [139].

ing sensitive authentication information. These solutions have also been shown to introduce extra authentication time and effort [57]. Other solutions attempt to obscure the attacker's view of the sensitive information [192]. We refer to these solutions as *passive solutions*, as none of these works give the victim of these attacks an indication that an attack is in progress. Not only do these solutions provide little protection for other information (e.g., their written addresses, bank account information, video chats, and texts), but they also offer no protection *after the user has completed the authentication*.

Active solutions attempt to detect and alert the user to a potential SSA in real-time. This gives the user a chance to act and reduce or remove the threat. The most recent active solution is nearly two years old and is limited to mobile devices [20]. Additionally, of the other three known active methods, two are designed only for mobile devices, and the other is a convex mirror attached to ATMs [8, 146, 172]. This shows a lack of research into this active method of preventing SSAs.

While both types of solutions differ in their protection methods, both passive and active solutions are currently limited to a specific type of protection domain. These solutions work for either mobile devices, Personal Identification Number (PIN) entry pads at an Automated Teller Machine (ATM), desktop/laptop computers, or other devices, and one solution does not protect users across multiple domains. This limits each solution's real-world feasibility and makes them less usable.

In this work, we define *usable* in two parts: the ability of the solution to protect the individual user in a wide variety of scenarios, and the solution's minimal negative impact on the user's completion of the potentially SSA-vulnerable task. To highlight the lack of usable solutions to SSAs, we offer a simple observation. Throughout daily life, especially since the advent of mobile devices, people all over the world are often asked to both write and type sensitive information regularly. It is easy to imagine a person being asked to write an address or personal information on a government form and log into a bank account/ATM on the same day. As of the time of writing this paper, no existing solution could protect individuals from SSAs during all of these activities as they occur in different environments and scenarios. Additionally, the vast majority of existing solutions are designed to only protect information during authentication, such as during PIN or password input, and add extra time and effort to this process.

To create a unified solution to these different attack scenarios, we must understand the basic components of any SSA. With few exceptions, these attacks involve a victim using their eyes to input or write sensitive information, and an attacker using their eyes to observe it. Exceptions, such as electronic surveillance, exist but are generally not as prevelant [10, 13, 21, 58, 92]. Using this understanding, we can imagine a solution that uses multimodal eye tracking (i.e., from both the attacker and victim) as a universal method to detect and warn of SSAs. Head-mounted eye-tracking devices provide the ability to track the eyes of

the device user to monitor their focus through information such as eye fixations [114, 143]. Finally, with the addition of a modular rear-facing camera and readily available facial and iris detection methods, we can analyze the potential attacker's eye gaze as well.

Such a solution also needs a way to notify the user of a detected attack. To address a potential attack, the device user must know generally where the attack is occurring to protect themselves. If the user does not know this, moving or adjusting the position of their sensitive information may not have the desired effect. Head-mounted devices, such as augmented reality (AR) headsets discreetly provide this capability, only visible to the user. A recent article from the National Institute of Standards and Technology (NIST) recommended AR to "help preserve data confidentiality during display of sensitive information compared to handheld or desktop displays" [65]. Currently, these headsets are not as ubiquitous as other devices, like mobile phones, but are expected to become more and more common as they become lighter and less obtrusive [111, 118]. As an example, the Apple Vision Pro had reported sales of 200,000 devices in preorders alone [75].

For these reasons, we will adopt head-mounted AR as a platform to harness user eye gaze, collect rear-facing camera frames (with added off-the-shelf hardware), and discreetly notify the user of an attack. AR headsets, especially when envisioning a future of smaller, lighter headsets, can also be worn in a wide variety of tasks, including most attack scenarios (typing, writing, walking, driving, etc.) [111]. AR headsets provide a platform of data sensors that we require for our system design but are not necessarily the only platform that fits our proposed design.

We summarize the contributions of this paper as follows:

- We present our design called **SHOULDAR**, a novel, wearable, mobile, usable system designed to protect users from SSAs. Using an off-the-shelf camera and microcontroller,

an AR headset, and multimodal eye tracking, we use eye gaze information from both the user and attacker to alert the device user to potential shoulder surfing threats in both typing and writing scenarios.

- Through a user study with 24 participants, we evaluate the effectiveness of this system, the effect of the system on comfort in SSA-vulnerable situations, and the perceived usability. Our solution successfully detected an SSA in 87.28% of tests across *both* typing and writing tasks, a more than two-fold increase in protection over the baseline solution using a rear-facing mirror, a widely used solution to the SSA problem.

- Our system improves perceived participant comfort in a variety of representative scenarios and was judged to be usable, especially when compared to the baseline solution.

5.2 Challenges

We seek a solution that improves the usability of existing work based on the definition presented in §5.1. If we imagine such a solution, two key requirements begin to emerge: **(R1)** the system must be generalizable to a myriad of scenarios (e.g., ATMs, office desks, whiteboards, mobile devices, and desktop/laptop computers); **(R2)** the system must not interfere with the user's actions. Specifically, the system cannot prohibit typing and writing, or create long task completion (e.g., login) times. As passive systems generally only protect authentication and are not generalizable, we seek an active solution to this problem. Next, we explore two prominent challenges of such an active system.

5.2.1 Challenges

A usable solution should work in various types of scenarios. While §2.2 reviews multiple works aiming at deterring or identifying SSAs, each solution is confined to a particular domain. For instance, graphical or PIN-based methods require a digital display or a physical PIN pad. Existing works designed to alert users that a person is looking at their device generally require front-facing cameras that can be obscured by the user and are not generalizable solutions. Rear-facing mirrors are mostly mounted on ATMs and are generally not mounted in other areas, like in office settings or on desktop/laptop computers. Therefore, *the first challenge is to create a method that can be generalized across multiple domains, both digital and physical* ***(C1)***.

A usable solution must not complicate the user's task or add cognitive burden. Passive techniques, like those discussed in §2.2, have been shown to require good color vision or hand-eye coordination and can be difficult for the elderly or persons with intellectual disability to use [23]. Additionally, many of these schemes have been shown to increase task completion (e.g., login) times or add a cognitive burden to users that have been deemed unacceptable [57].

Some existing active solutions, like the ATM mirror, require the user to constantly monitor their surrounding for threats. Such a requirement can add a cognitive burden to the user by forcing the user to assess and respond to perceived attacks on their sensitive information. Additionally, to ensure that an active solution is timely and useful, the system should operate in real-time. If not, the user would have to decide if the threat is still present. This can remove attention from the task at hand. Therefore, *the second challenge is to create a system that runs in real-time, does not interfere with task completion, and does not place a cognitive burden on the user* ***(C2)***.

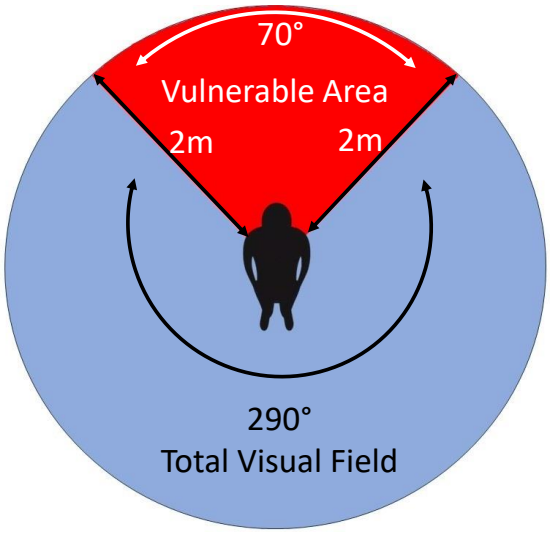


Figure 5.2: ShouldAR Threat Model. Given the human eye's ability to sense motion at nearly 290° when allowed to move the head and eyes (with the torso stationary), we consider the remaining 70° to be the most vulnerable area for an SSA. Additionally, previous work shows that attacks on mobile devices and other common targets are most feasible inside 2 meters. We define the *vulnerable area* in red as the most likely area for an SSA.

5.3 ShouldAR Design

This section incorporates the insights in §5.2 into a coherent design, called ShouldAR, centered around eye tracking and AR. ShouldAR is based on an AR headset that includes eye-tracking sensors and is capable of detecting attacks in a wide variety of scenarios. Additionally, all the information collection, processing, and decision-making are transparent to the user and conducted implicitly by the device and in real time. The user simply has to wear their AR headset while an application such as ShouldAR is running, and does not need to remember specialized information, such as PIN or image combinations. By combining these advantages, the ShouldAR design presented in this section is capable of protecting against SSAs in a wide variety of scenarios. We begin with a brief review of the threat model with an eye toward the specific limitations of a feasible SSA, followed by discussions about the key components of ShouldAR.

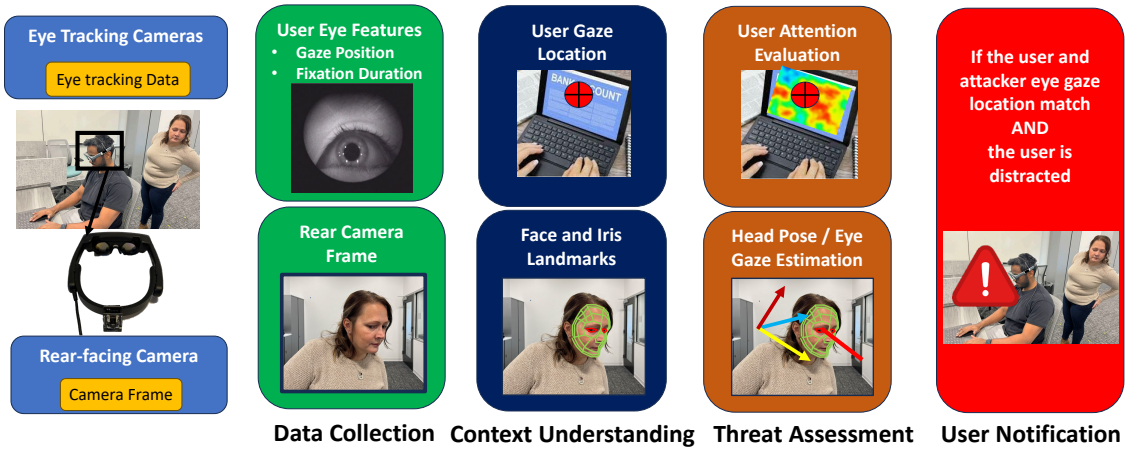


Figure 5.3: SHOULDAR Design. The design, shown from left to right, begins by harnessing the AR device's onboard sensors and an additional rear-facing camera. After this, in Step 1 (Data Collection), the raw data is collected for both the user's and the potential attacker's position and eye gaze. This data is analyzed in Step 2 (Context Understanding) to determine the user's gaze location and the potential attacker's facial and eye landmarks. In Step 3 (Threat Assessment), this context is analyzed, and the user is alerted if they are deemed vulnerable to an SSA in Step 4 (User Notification).

5.3.1 Threat Model

Informed by works such as [20, 78, 155], we define the limits of the SSA in terms of space and angle from the target of the attack. SSAs are generally from behind the user, where the user is unable to detect and prevent the attack. While attacks are possible against large displays at larger distances, existing literature on the topic shows that a maximum attacker distance of 2 meters is generally the farthest an attack can feasibly be conducted against a wide range of targets (mobile devices, paper, desktops, etc.) [20]. Also, the field of view (FOV) of the average human (assuming a stationary head, but mobile eyes) is as much as 290° [78]. This is also known as the *total visual field*. This leaves roughly 70° of FOV where the person is unable to detect either objects or motion to their rear. We call this area roughly 70° behind the user at a distance of no more than 2 meters the *vulnerable area*. Any SSA from outside the vulnerable area would risk the possibility of the potential victim noticing this attack.

For this reason, we limit our threat model to 40° on either side of the potential victim (a total of 80°, which covers the vulnerable area) as measured from directly behind the person, allowing for a small margin at the extreme edges of human sight. Fig. 5.2 gives an illustration of this area. Finally, in this threat model, we assume that the face of the potential attacker is visible and is not deliberately obscured by either digital facial obscuration method such as [157], or any other method such as using a hand or an object. We discuss this more in §4.4.

5.3.2 AR Headsets - A Feasible Platform for Generalized Anti-SSA Techniques

As noted in §3.1.2, AR headsets provide a suitable platform for our envisioned SSA detection technique. AR headsets are portable and can be expected to be worn in a wide variety of situations especially when considering the lighter and less cumbersome versions of the future [15, 118, 166]. Additionally, AR headsets normally have eye-tracking sensors that give us insight into a variety of attack scenarios. Our design centers around such an AR headset. However, to observe the area behind the user (i.e., the vulnerable area), we require a rear-facing camera that most current AR headsets do *not* have. To overcome this obstacle, we retrofit the headset with a rear-facing camera, something potentially integrated into future headsets. With this addition, we can estimate the eye gaze of both the potential attacker and user simultaneously on a mobile device usable in a variety of scenarios.

5.3.3 Design Components

Fig. 5.3 gives an illustration of the ShouldAR design. ShouldAR collects raw eye tracking and camera data (Step 1), analyzes this data (Step 2), evaluates user vulnerability and threat

(Step 3), and alerts the user if required (Step 4).

Step 1: Data Collection. Our design must collect multimodal eye gaze information about both the user and potential attackers in the physical environment, specifically in the vulnerable area up to 2 meters and in a 70° area behind the user. With the retrofitted, rear-facing camera, we can capture images directly behind the device user and identify potential attackers. Additionally, using the inherent functionality of the AR headset, we can capture metadata from the front-facing camera that gives us the user's current location and head pose (e.g., the roll, pitch, and yaw of the user's head) for each of the captured rear-facing frames. This information will be used later to locate the device and adjust rear-facing camera detections according to the position and rotation of the user's head. Finally, using the onboard eye-tracking cameras, we have a constant estimate of the user's eye gaze. We can also extract eye-tracking features (e.g., the fixation duration) identified as important to our design in §3.1.2.

Step 2: Context Understanding. In this step, we use the information collected from the headset and the rear-facing camera to develop context about the scenario. From the AR headset, we can gather the user's position and head pose. Additionally, with the benefit of the AR headset's eye tracking, we know where the user is looking in real-time. From the rear-facing camera frame, we use a facial detection inference model (e.g., MediaPipe [69] or OpenCV [151]) to locate any face in the vulnerable area. This inference model must have enough detail in its output to accurately calculate the *Perspective in Point (PnP)* of the face [184].

One of the outputs of the *PnP* calculation is the rotation of the face, needed to estimate the potential attacker's head pose. Using the localization and head pose of the user, combined

with the rotation and local position of the face in the rear-facing camera, we can calculate the position and orientation of the face to the user's rear. We also require information about the person's iris position, pupil position, or other information indicating the direction of the eye [203]. Facial detection models, such as MediaPipe [69], include this information with the output of the model.

Step 3: Threat Assessment. To reduce false positives, we harness the user's eye gaze in another way to determine when they are vulnerable to a threat. As mentioned in §3.1.2, fixation duration gives us insight into the concentration level of the user. Using the device's onboard eye-tracking sensors, we can measure the average fixation duration of the user and compare this against a threshold (discussed in more detail in §5.4). We can then estimate the direction of the attacker's gaze and notify the user of a threat when it aligns with the user's area of focus.

Step 4: User Notification. Knowing the location of a potential attacker, their head pose and eye gaze, and the position and attention level of the user, we can combine this information to determine (1) if a person is positioned in a vulnerable area behind the user, (2) the target of the device user's eye gaze and their attention level, and (3) if the person to the rear of the user is looking over the shoulder of the user towards the same target. Applying minimum values to the distance between the user and the potential attacker's eye gaze and the fixation duration of the user, we can also create gaze overlap and attention thresholds. If these thresholds are met (e.g., a person to the rear of the user is looking over their shoulder while the user is potentially unaware), we alert the user to the threat.

5.4 Prototype Implementation

To understand the usability and effectiveness of the SHOULDAR design outlined in §5.3, we implemented the design using a Magic Leap 2 (ML2) AR headset [99]. This section outlines the technical specifics that were used to realize the SHOULDAR design presented in §5.3.

We developed the prototype using Unity version 2022.2.19f1. All code was written in C and C#. To implement the design outlines in Step 1 (Data Collection), we implemented the SHOULDAR design using an ML2 AR headset, as shown in Fig. 5.4a. This device has integrated eye-tracking sensors and also provides 3D spatial understanding through simultaneous localization and mapping (SLAM) [99]. The rear-facing camera requirement was satisfied using an ESP32-CAM microcontroller and an OV2640 camera with a wide-angle (140°) fish eye lens [14, 61]. A socket connection was created from the ESP32-CAM to the ML2 using a TP-Link AC1900 wireless router [195]. Raw frames were captured using the ESP32 and OV2640 and transmitted using this socket for analysis on the headset itself.

For Step 2 (Context Understanding), the face and iris landmark detection was created using MediaPipe [69]. Our prototype did not require offloading facial inference, as the ML2 proved to be fully capable of conducting this task on-device with negligible impact on the device's expected frame rate and an average of 25 millisecond inference time. As our evaluation only included a single attacker, each captured frame was analyzed for the presence of a face, and if one was detected, the prototype used MediaPipe to gather the required landmarks for Step 3.

In Step 3 (Threat Assessment), we used a 300-millisecond threshold for average user eye fixation duration as a minimum value to determine user attention. This threshold was decided during pilot testing in keeping with the maximum expected fixation duration average when a human is not concentrating [114]. Above this, we expect "deeper cognitive processing",



(a) ShouldAR Prototype



(b) Baseline Solution

Figure 5.4: (a) shows an image of the ShouldAR prototype. A Magic Leap 2 headset was used as the AR platform, and the rear-facing camera, shown at the rear of the headset, was an ESP32-CAM and OV2640 wide-angle camera attached using Velcro fasteners. (b) provides an image of the baseline solution provided for participants. This two-inch fisheye convex mirror was used in both the typing and writing scenarios and was intended to simulate an ATM mirror.

indicative of increased concentration. After determining the head pose (e.g., the rotation of the head), we use a simple method for determining the gaze direction of the potential attacker by estimating which side of the eye the iris is nearest. We also require a threshold to consider the user and potential attacker's gaze to have overlapped, indicating they are looking at the same target. The point of the user's eye gaze location is surrounded by an invisible, 3D sphere with a radius of 0.5 meters. If the attacker's gaze intersects with the user's gaze sphere, we consider this as gaze overlap.

Finally, in Step 4 (User Notification), we evaluate the user's attention (using fixation duration) and the overlap of the user and potential attacker's eye gaze locations. If conditions are met, we present the user with an arrow pointing to the location of the suspected attack in 3D space. While we do not test more than one notification technique, we suggest possible

alternatives in §5.6.

5.5 Evaluation

To understand the impact of the SHOULDAR prototype, we crafted an evaluation designed to measure the system's effectiveness (through quantitative testing in a variety of situations) and its usability. Before and after participants used the physical prototype, we asked them subjective questions about their comfort in different situations, spanning multiple angles and distances inside the shoulder surfing vulnerable area. These questions were designed to understand the impact, if any, that systems such as SHOULDAR might have on user comfort in potentially vulnerable situations. Additionally, we asked a series of questions designed to evaluate the usability of the design and prototype.

5.5.1 Evaluation Procedure

Our evaluation was designed to measure participant comfort with and without an active SSA protection system, and the effectiveness of the SHOULDAR system itself. This evaluation was approved by the Institutional Review Board (IRB) of our university.

Demographics. Participants were recruited from university graduate and undergraduate mailing lists and solicitations to local, non-student campus personnel and were compensated $20 for approximately 45 minutes of time. Our evaluation included 24 participants across both the pilot and main study. Of the 24 participants, 54.1% self-declared as male, while 45.8% self-declared as female. The ages of participants ranged from 19 to 39, with occupations ranging from nurses to graduate students at our university. Since the ML2 requires

Table 5.1: We asked each participant to assess their comfort in 6 scenarios as an initial estimation of their overall comfort in potentially vulnerable situations.

Scenario	**Task**	**Attacker Distance**	**Attacker Direction**
S1	Typing	1 meter	20° Left
S2	Typing	1.5 meter	0° Behind
S3	Typing	2 meter	40° Right
S4	Writing	1 meter	20° Left
S5	Writing	1.5 meter	0° Behind
S6	Writing	2 meter	40° Right

prescription corrective lenses specific to the device, we did not allow participants to wear glasses. While 76.2% of participants were wearing neither contacts nor glasses during the testing, 23.8% were wearing contacts. The majority, 56.5%, of participants had never used AR before, or only once or twice.

Initial Participant Comfort Assessment. The evaluation was designed to explore the impact of an active, usable system on comfort in a situation that was vulnerable to an SSA. We began with introductions and signing an informed consent form. Then, we asked the participant to answer six questions about their comfort in six different scenarios. The scenario descriptions are outlined in Table 5.1. The participant was then asked to imagine a person standing behind them at a given distance and direction. They were then also asked to imagine that they were either typing a password at a desktop computer (the *Typing* task) or writing sensitive information on a piece of paper at a desk (the *Writing* task). Participants were asked to rate their comfort on a 5-point Likert scale ranging from 1 (Least Comfortable) to 5 (Most Comfortable) [117]. While a sketch of each scenario was provided, the evaluator was also positioned to aid in understanding the attack parameters and increase realism.

ShouldAR Prototype Evaluation. After the completion of the initial survey, participants were given a brief tutorial on the use of the ML2 and completed the ML2's fit and eye

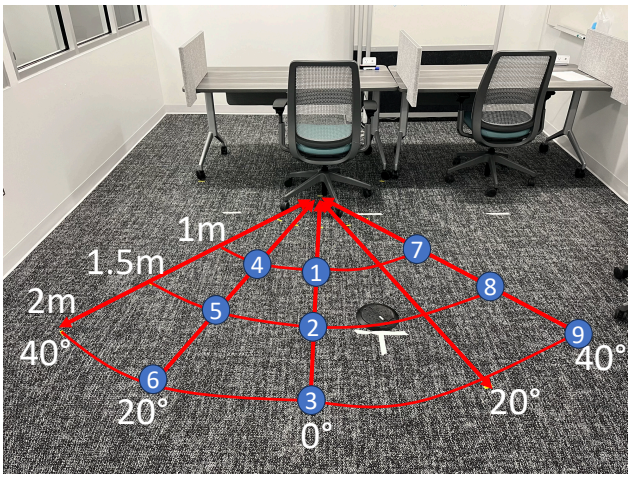


Figure 5.5: An image of the testing environment. The red lines illustrate the distances and angles we evaluated inside the vulnerable area. The blue dots show an example of the ordering of the nine positions of the potential attacker.

gaze calibration. After calibration, participants were asked to complete one of two tasks: data entry into an online typing assessment or completing a set of short answer questions using pen and paper. These tasks were designed to emulate the input of sensitive information during both typing and writing, similar to entering a password or writing an identification number or an address on a form.

During each of these tasks, a simulated attacker moved to nine different positions in the evaluation area behind the user. The test environment with the locations of the nine positions is illustrated in Fig. 5.5. The attacker remained in each of these positions for exactly 10 seconds. The participant was given instructions to report when they believed this attacker was observing their actions by either raising a hand or speaking the word "attack". The nine positions of the attacker were repeated for both the typing and writing tasks, across five individual test types, representing a baseline and two variations of the SHOULDAR prototype, for a total of 54 individual positions. The participant was instructed to conduct their task (typing or writing) and report potential attacks simultaneously. Each of the nine positions included variations in attacker distance, offset (e.g., angle from participant), head position,

Table 5.2: Participants were asked to rate their agreement with the following statements or each question. A 5-point Likert scale was given, ranging from 1 (Strongly Disagree) to 5 (Strongly Agree).

Question	Usability Evaluation Statements
1	The prototype you evaluated was easy to use.
2	The prototype was overly intrusive on your activities.
3	The requirements of the prototype (e.g., wearing the headset) were worth the benefit I received.
4	If made smaller and more lightweight, I would wear this system as part of my daily routine.
5	I prefer the prototype's level of protection to methods such as the mirror to inform me of shoulder surfing attacks.
6	I believe that using a method to defend against Shoulder Surfing Attacks that require direct observation (such as a mirror) is more usable than the prototype.
7	I would use this system if made less cumbersome and integrated into future AR devices.
8	I felt more secure against shoulder surfing attacks using the prototype than I would with any other method I know of.

and eye gaze angle. At each distance (1m, 1.5m, and 2m), the attacker's head and eyes were directed away from the participant's information, head and eye directed at the participant's information, and with the attacker's head rotated away, but eyes directed toward the information. These tests were designed to test for false negatives as well as the usefulness of head pose alone and with eye gaze, to isolate the added value of using the attacker's eye gaze.

For the baseline, the participant was provided with a small (two-inch diameter) fisheye convex mirror attached to the laptop at their desk (see Fig 5.4b for an illustration). This mirror was designed to emulate an ATM mirror a widely used solution to the SSA problem that could be used in both typing and writing scenarios. Other solutions, like [20], are designed for only a single situation, like mobile devices. The participant was allowed to reposition this baseline solution as they saw fit during this part of the evaluation. For each test, the "attacker" moved to 18 positions for both the typing and writing tests (9 for each). In total, there were 54 tests for the baseline and two versions of ShouldAR.

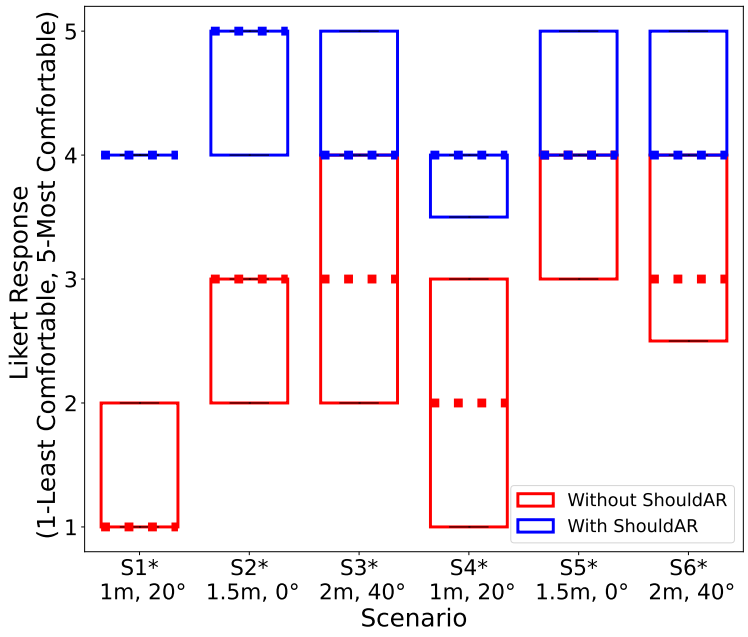


Figure 5.6: A comparison of the Likert responses of participants when asked about their comfort in the situations outlined in Table 5.1. The box plots in red are the questions asked assuming no protection. The blue box plots are the questions asked assuming protection from an active system such as SHOULDAR. The dotted lines show the median values. Scenarios with an asterisk (*) indicate statistically significant results ($p < 0.05$).

Final Participant Assessment. Immediately following the prototype evaluation, the participants were asked to complete a final questionnaire. The first six scenario descriptions were nearly identical to Table 5.1, but asked the participant to imagine these scenarios *while wearing the prototype they had just evaluated*. This was designed to gauge any change in the participant's concern in these scenarios if protected by a system such as SHOULDAR. Participants were asked eight additional questions designed to assess the usability of the prototype (see Table 5.2). Participants were asked to rate their agreement or disagreement with each question using a 5-point Likert scale ranging from 1 (Strongly Disagree) to 5 (Strongly Agree).

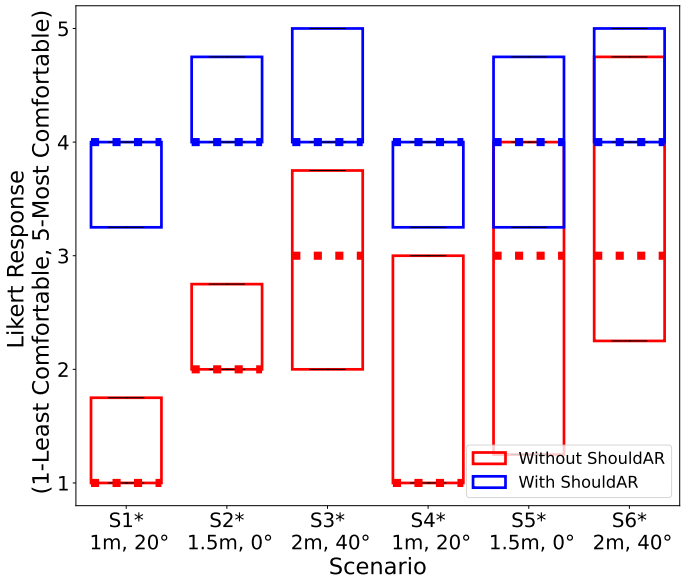
We provide an evaluation of SHOULDAR in three parts. First, §5.5.2 shows the results of the Likert responses of the participants before and after experiencing an active, mobile protection

system such as ShouldAR. Second, §5.5.3 gives quantitative results on the effectiveness of ShouldAR, expressed as true positive and false negative rates in various scenarios. Finally, we seek to understand how participants felt about the prototype's usability. §5.5.4 outlines the qualitative responses from users when asked to agree or disagree with eight statements about usability.

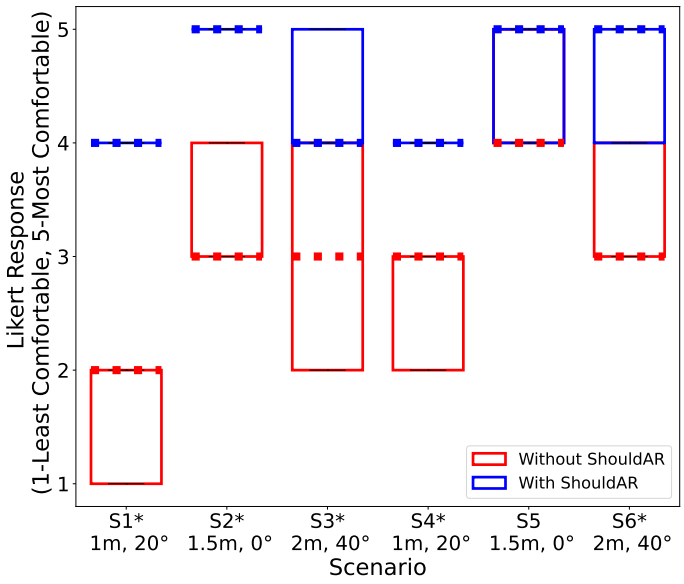
5.5.2 Understanding ShouldAR's Impact on Comfort in SSA-vulnerable Situations

Before and after evaluating the ShouldAR prototype, participants were asked to rate their comfort in six scenarios on a 5-point Likert scale. The scenarios are outlined in Table 5.1 and consist of three different angles at three different distances while the user is completing one of two tasks (typing or writing). Fig. 5.6 shows the responses for the six scenarios before (in red) and after testing the prototype (in blue). Overall, *participants were more comfortable in each situation while wearing an active protection system, such as* ShouldAR*, than they were without.* Intuitively, participants were more concerned with the attack scenarios where the attacker is close (S1 and S4) than the longer-range scenarios. Generally, ShouldAR increased median responses for participant comfort from neutral or lower to positive responses, with the largest individual scenario changes in the close-range and typing scenarios. For each question, a Wilcoxon signed rank test showed statistical differences in the pre-evaluation and post-evaluation Likert results ($p < 0.05$).

Fig. 5.7 breaks down the participant comfort analysis by self-declared gender. Here as well we can see that participants, regardless of gender, reported an increase in comfort while protected by an active system such as ShouldAR. For male participants, ShouldAR increased median responses for participant comfort from neutral or lower to positive responses. For the



(a) Comfort Result for Female Participants



(b) Comfort Result for Male Participants

Figure 5.7: A comparison of the Likert responses of participants when asked about their comfort in the situations outlined in Table 5.1, separated by participant gender. (a) is the responses from self-declared female participants. (b) is from self-declared males. The box plots in red are the questions asked assuming no protection. The blue box plots are the questions asked assuming protection from an active system such as ShouldAR. The dotted lines show the median values. Scenarios with an asterisk (*) indicate statistically significant results ($p < 0.05$).

	Baseline		Head Pose (HP)		HP w/ Eye Gaze	
User Actions	**Typing**	**Writing**	**Typing**	**Writing**	**Typing**	**Writing**
True Positive	43.86%	41.23%	58.77%	53.51%	**89.47%**	**85.09%**
Total True Positive	42.54%		56.14%		**87.28%**	
False Positive	0.88%	0.88%	**0.00%**	1.75%	0.88%	**0.00%**
Total False Positive	0.88%		0.88%		**0.44%**	

Table 5.3: A summary of the results of the ShouldAR quantitative evaluation. The "Baseline" column represents the tests conducted using the two-inch diameter fisheye mirror. The "Head Pose (HP)" column shows the tests using only the attacker's head pose, and not eye gaze. "HP w/ Eye Gaze" shows the results using both head pose and eye gaze. Each modality is shown subdivided by individual test tasks (e.g., typing or writing) and as a total average.

male results, all questions showed statistical significance ($p < 0.05$), except Q5 ($p = 0.18$). Female participants also increased median responses for participant comfort from neutral or lower to positive responses considering a prototype like ShouldAR. All questions showed statistical significance ($p < 0.05$).

5.5.3 Evaluating ShouldAR's Effectiveness

Next, we seek to evaluate ShouldAR quantitatively by analyzing the True Positive (TP) and False Positive (FP) rates recorded during the evaluation. §5.4 outlines the participant tasks and attacker positions evaluated. The baseline solution consisted of a two-inch diameter, fisheye mirror positioned by the participant during both the typing and writing tasks. This baseline is compared to two versions of the ShouldAR prototype, one using both attacker head pose and eye gaze estimations, and one only using head pose. This was designed to isolate the effect of eye gaze on the results.

Table 5.3 provides a summary of the results of these three tests. The baseline (mirror) solution showed the lowest TP rate with an average of 42.54% of attacks being detected by the participant. The FP rate of the baseline was also nearly one percent, equal to the

head-pose-enabled solution. The majority of the False Negative (FN) results, where the participant failed to indicate an ongoing attack, occurred when the participant obscured the view from the mirror.

The head-pose-only solution, using only the direction of the potential attacker's head to decide if they are attempting an SSA, improved on the baseline performance with a TP average of 56.14%. The majority of the FN results for the head pose method occurred when the attacker's head was directed *away* from the participant's task, but their eyes were directed *toward* the task. This shows the vulnerability of such a solution to these scenarios.

The eye gaze solution, including both the attacker's head pose and eye gaze to detect an attack, performed best. This version of the prototype averaged an 87.28% TP rate over both writing and typing tasks. While not perfect, this shows roughly a *two-fold increase in TP rate over the baseline, and a 31% increase over the head-pose-only solution.* This 31% increase in TP rate is directly related to the successful head-away/eyes-toward scenarios that hampered the head pose-only version. The majority of the FN tests (the tests where the attacker was observing the participant's information but the participant did not indicate an attack) occurred at the 2 meter attack distance (18 out of 29 total false negatives). This shows a limitation of this technique, as small errors in eye gaze estimation are magnified at larger distances from the participant. Previous work shows, however, that distances farther than 2 meters are less vulnerable to SSAs [20], making this less of a concern. Additionally, the FP rate of this system was less than 0.5%, resulting in only a few unnecessary warnings to the participant.

5.5.4 Evaluating ShouldAR's Usability

Finally, we evaluate the results of the usability-focused questions posed to the participants. Fig. 5.8 shows the results of the questions outlined in Table 5.2. Questions 1, 3, 4, 5, 7, and 8 were all designed to elicit a higher Likert score if the user believed the prototype was usable in general, while Q2 and Q6 were designed such that lower responses indicated higher usability. The flipped scales were introduced to ensure participants were not just blindly answering the short questionnaire. With the flipped scale, all usability questions score over a 4, showing agreement or strong agreement, with only outliers at neutral scores and below.

In conversations after the evaluation, users expressed that the prototype allowed them to concentrate on the writing or typing task far more easily than the baseline (mirror) solution. The baseline required them to constantly evaluate their vulnerable area and distracted them from their task. Knowing that ShouldAR would alert them in the case of a potential attack, participants expressed a general preference over the baseline, shown empirically in the usability evaluation results.

5.6 Discussion

To address a lack of usable solutions to SSAs, we designed, implemented, and evaluated ShouldAR. This system harnesses both the potential attacker's and victim's eye gaze to determine if an SSA is likely and to give the victim information that can be used to stop it. Through our evaluation, we determined that such a system improves the perception of comfort when participants were asked to imagine potentially vulnerable situations. Additionally, we proved that such a system can accurately detect SSAs with a low (e.g., less than 1%) false positive rate in the scenarios we evaluated.

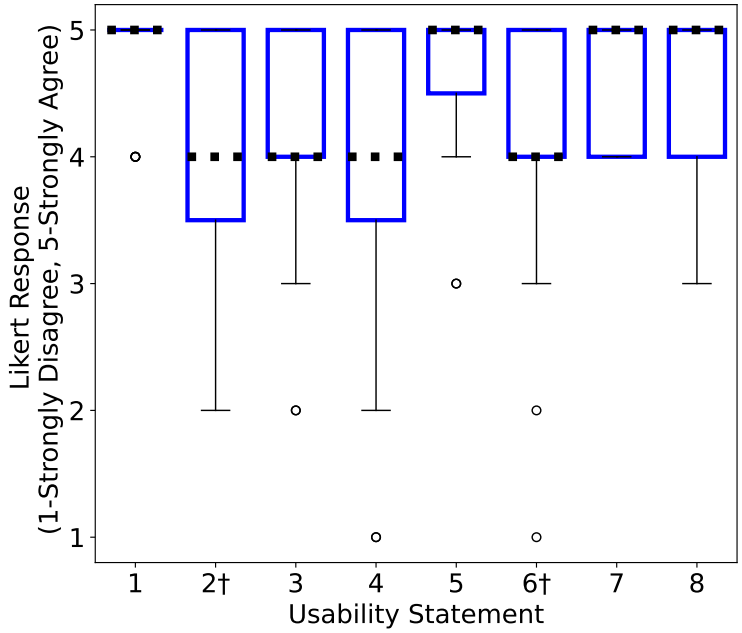


Figure 5.8: A comparison of the Likert responses of participants when asked about their comfort in the situations outlined in Table 5.2. The dotted lines show the median values. **Note:** Questions 2 and 6 (marked with a †) were worded in such a way as to ensure the participant was carefully reading the questions. The Likert values for these questions were inverted in this plot for ease of comparison with the remaining questions.

What if the Attacker's Eyes Are Not Visible? SHOULDAR works by matching the user's eye gaze with the potential attacker's eye gaze in 3D. However, the inference model that is analyzing the rear-facing camera frames may detect a face but not the person's eyes. This can be due to intentional or unintentional obscuration with the person's hand, sunglasses, or other means. If the prototype detects a face but does not have the benefit of the eye-gaze estimation, the prototype defaults to head-pose-only estimation. Using head pose only as a last resort is sufficient when the attacker's eyes are in the same direction as the head, but results in reduced true positive rate when the head and eyes are not aligned. Last, if the attacker is intentionally wearing a mask to obscure their face from facial detection and head pose estimation, more advanced sensors such as thermal cameras could be used to provide an alternative method of estimating head pose [91].

Calculating the Attacker's Eye Gaze. There are many ways to estimate the eye gaze of a person from an image of their face. Techniques such as [110, 152] use the neural networks, trained on labeled images of specific eye orientations, to determine the gaze angle. Our prototype sought to use a lighter, faster method, that did not involve neural networks. Using a rough estimation of the gaze angle based on the distance of the iris from the eyelid landmarks, we were able to estimate eye gaze using hard-coded methods such as [38]. This is not as accurate as using neural networks (e.g., a convolutional neural network), but we believe it to be less likely to suffer from delay due to the inference time required for each frame. Other methods that could create a more fine-grained estimation of the eye angle, such as [178], exist as well. These were not used as part of our prototype. In our test, a maximum attacker distance of 2 meters was used for the reasons described in §3.1.2. At farther ranges, this course-grained estimation method may begin to fail. This limitation is less of a concern in light of previous work explaining the limited feasibility of SSAs from beyond 2 meters [20].

Attack Notification Techniques. Our evaluation was designed to measure the user comfort resulting from using our prototype as well as the system's effectiveness at detecting SSAs. While wearing the Magic Leap 2 during the evaluation, future studies into designing optimal notification cues for SSAs and evaluating user perceptions are necessary.

Future Work. In future work, we plan to explore improving system accuracy at or beyond 2 meters. Additionally, we plan to develop user studies to investigate an optimal method for alerting the device user once an SSA is detected. This work could analyze other visualizations or multimodal (e.g., audio or haptics) alerts. Little research effort was given to determining and analyzing the best methods to alert the device user to an SSA once one is detected. The simple 3D arrow used in our prototype was not the focus of rigorous analysis or investigation.

Our future work will analyze other, possibly less intrusive or overt methods to alert the user to an SSA.

Additionally, the system user has no way to declare that they are aware of and consent to a specific person's observation of their sensitive data. Currently, the user must halt ShouldAR temporarily if they wish to prevent false positives in this scenario. There are potentially more intuitive ways to give a user the ability to label a face as a "friend" and to ignore potential attacks from this person. Specifically, creating a facial recognition system to allow the user to select certain faces, and compare these "trusted" faces to each facial detection. If the faces match (e.g., the face is a "friend" or a "collaborator" on a project with a common display), this detection could be ignored.

5.7 Summary

To address a lack of usable solutions to the SSA problem, we designed, implemented, and evaluated ShouldAR, a novel, wearable, mobile, usable anti-SSA system. This system uses an AR headset and multimodal eye gaze information from both the headset user and the potential attacker. ShouldAR is usable in a wide variety of scenarios, including when writing, using an ATM PIN pad, or using a mobile device. This system was shown to be preferable over a widely used anti-SSA solution, a rear-facing mirror. While AR headsets are not as ubiquitous as mobile phones, yet, we believe that such a solution would be feasible. After improvements in weight and battery life, a solution such as ShouldAR would be effective at reducing user concern in SSA-vulnerable scenarios.

Chapter 6

GazePair: Efficient Pairing of Augmented Reality Devices Using Gaze Tracking

6.1 Introduction

According to recent market research, the global Augmented Reality (AR) market in 2021 was estimated at $14.7 Billion, with an expected value of $88.4 Billion by 2026 [165]. As companies and users begin to explore the possibilities of the Metaverse and the future of human interaction [11, 204], AR is rapidly expanding past its current hardware limitations to even more ubiquitous use. These uses, such as in the windshields of automobiles [54], use by the military for soldier augmentation [19], performing remote surgery and training healthcare workers [158], educating children in the classroom [134], and visualizing logistics bottlenecks [153], portend a future where AR devices are used as a part of our normal, everyday lives. With the advent of the Metaverse, even the governments of large cities are turning to AR to ensure that their citizens have access to essential services through this emerging and potentially pervasive technology [100].

As these AR devices grow in utility, use, and impact on daily lives, schemes to pair two

or more of such devices will become even more important. The pairing of AR devices and sharing of experiences is at the core of the value of AR devices, allowing users to not only experience a synthetic augmentation of the physical world but to share these objects, normally known as holograms, with others. On the other hand, AR devices present unique challenges and opportunities for pairing. AR devices, especially head-mounted displays (HMDs), allow the user to interact with her physical environment while the headset places synthetic objects such as holograms into the user's perception of the physical world. HMDs also allow for integrated eye tracking, a feature deemed one of the most transformative to the technology [182]. These features allow for unique ways to generate the entropy required for secure pairing of AR devices, noted as 60+ bits of entropy in [36]. This is in contrast to other mobile devices, such as smartphones, that have limited ways for users to interact for device pairing.

6.1.1 The Importance of Local Pairing of AR Devices

The importance of efficient pairing of AR devices is made evident in previous works. As originally described in the earliest work on AR-specific local pairing, Looks Good To Me (LGTM) [63], local sharing is a problem that deserves to be studied separately. Local sharing allows users to communicate without using large-scale data backbones, for-profit cloud services, or cellular connections. It also gives users the freedom to decide to keep their data local and within a more closed sphere of control. Currently, for two users to pair their AR devices and securely share information such as holograms, the best working solution involves exchanging an alphanumeric authentication string or using centralized systems such as Microsoft Azure [122]. If focusing on local, bootstrapping methods of pairing, the alphanumeric string method is the only known, implemented method. These strings are error-prone and time-consuming to use [84]. To alleviate this problem, recent research has created systems to

use AR's spatial awareness capability, combined with the AR user's ability to interact with the physical environment, to efficiently pair two AR devices. LGTM, HoloPair [188], and Tap-Pair [189] are all current research works that focus on using AR-specific technologies to pair such devices. Each of these works presents a novel way to use wireless localization or holograms to authenticate a shared secret and secure communication paths without using Public Key Infrastructure (PKI) to create keys. However, none of these works implement or test methods of pairing more than two devices, and none of them explore a new and powerful technology, *eye gaze tracking*, for AR device pairing. Additionally, their proposed pairing techniques are specific to AR. AR-specific gestures or technologies require that each of these solutions be deployed to AR devices only, greatly limiting the deployability and scope of the solutions (e.g., not applicable to Virtual Reality (VR) devices). In light of this, *it remains highly challenging to achieve a high level of entropy required for AR device pairing, while simultaneously creating a scalable, usable, and widely deployable solution.*

6.1.2 GazePair

We propose to use eye gaze tracking to create the entropy required for secure pairing, and the usability and scalability desired by users. We adopt eye gaze tracking in our design for the following reasons. *First*, harnessing eye gaze simply requires the user to direct their eyes, or look, at a target. It requires little explanation. *Second*, eye gaze tracking is a relatively new method of user interaction with digital systems, and has been identified by some of the largest companies producing Mixed Reality (MR) devices (including AR and VR) as one of the technologies driving the future of this industry [169, 182, 186]. *Third*, an AR user's eye gaze is nearly invisible to an outside observer. Most AR devices have a partially opaque visor concealing the user's eyes. This prevents easy, direct observation of the target of the user's gaze.

Using eye gaze to generate the symmetric encryption keys required to pair devices, however, introduces unique challenges. *First*, eye gaze detection and logging involves inherent error [85], even if small. As a result, the discretization of this data can be challenging. Eye saccades, the natural movement of the eye from point to point, eye fatigue, and even user inattentiveness make this and other techniques difficult to implement and discretize. *Second*, the transition of eye gaze data to a symmetric encryption key is non-trivial. The system must not only be robust but also scalable (i.e., able to simultaneously pair more than two devices). *Third*, eye gaze and iris/retinal data can be uniquely identifying and are a potential privacy risk if leaked accidentally. Such a system must protect user identity and unique biometric data. *Fourth*, this system must use the created entropy to generate symmetric keys without requiring key authentication systems that limit scalability. We judge that minimum entropy must be 60+ bits in keeping with [36].

To address these challenges, we propose a novel pairing solution, GAZEPAIR, which uses *eye gaze tracking* to produce the 60+ bits of entropy required to pair devices securely. GAZEPAIR is a proven advancement in terms of usability and scalability over existing solutions and is also deployable on the breadth of emerging MR devices that use eye gaze tracking. GAZEPAIR uses a set of numerically-labeled holograms, randomly generated in 3D space, and a spoken *key sequence cue (KSC)*, to pair AR devices and mitigate the threat of Man in the Middle (MITM) attacks. GAZEPAIR also does not need to publicly exchange discretization or error correction parameters and does not require recording or sharing gaze data that can jeopardize user privacy. To prototype GAZEPAIR, we use the Microsoft HoloLens 2 AR HMDs to prove the hypothesis that efficient and secure local pairing is possible using eye gaze tracking as the primary entropy source for symmetric keys. Currently, the Microsoft HoloLens 2 is the most advanced AR HMD on the market [124]. It incorporates spatial awareness, eye gaze tracking, a relatively large field of view, and is completely wireless. While our prototype

implementation requires Microsoft's Mixed Reality Toolkit (MRTK), we expect that the GazePair design is applicable to any MR device that supports eye gaze tracking.

We summarize the contributions of this paper as follows:

- To the best of our knowledge, this is the first work to show that eye gaze data generated in AR devices can be used to produce the 60+ bits of entropy required for efficient and secure pairing.

- We design a novel system, GazePair, to locally pair AR devices using eye gaze tracking and a spoken KSC. GazePair improves on existing techniques in terms of usability, scalability, and security. Additionally, the design of GazePair is applicable to any MR device with eye gaze tracking.

- We implement a prototype system of GazePair using Microsoft HoloLens 2 AR HMDs. The prototype achieves a *98.3% success rate and an average of 9.02 seconds to complete a one-to-one pairing over 120 tests.* With three users, a one-to-two pairing takes *an average of 12.58 seconds and has a 96.6% success rate over 120 tests.* No existing work is proven to locally pair more than two AR devices or could claim to be deployable across the breadth of MR devices incorporating eye gaze tracking.

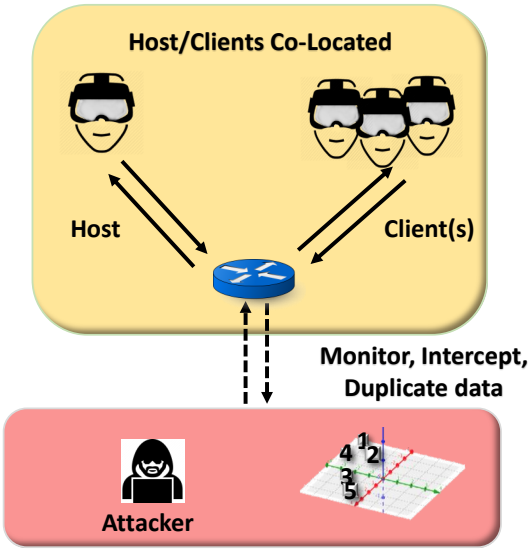
We organize the rest of this paper as follows. We first address the design and prototype implementation of GazePair in Sections 6.2 and 6.3, respectively. In Section 6.4, we present the experimental results of our user testing-based evaluation. Finally, we discuss the limitations of GazePair and future work in Section 6.5 and make concluding remarks in Section 6.6.

6.2 GAZEPAIR Design

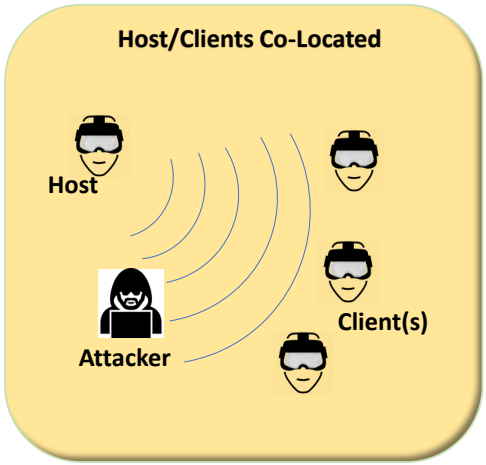
In this section, we propose a novel pairing system, GAZEPAIR, which leverages *eye gaze tracking*, a new and powerful AR technology. GAZEPAIR improves on the shortcomings of existing pairing strategies discussed in Section 2.3.2. In GAZEPAIR, we assume that users are wearing AR HMDs with gaze-tracking sensors. These AR HMDs have access to a local network, but may or may not have access to the larger internet through this connection. We also assume that the users must not be required to remove these AR HMDs to conduct any of the required tasks for pairing their devices. Additionally, we assume that the users are co-located, in such proximity that a spoken word can be easily heard by all legitimate clients. In the following, we first introduce the threat model in Section 6.2.1, present the design goals in Section 6.2.2, discuss the key challenges of using eye gaze tracking for pairing in Section 6.2.3, and elaborate on the design of GAZEPAIR in Section 6.2.4.

6.2.1 Threat Model

For GAZEPAIR, we assume a threat model that is similar to those proposed in the most recent research on AR device pairing [63, 188, 189]. We assume that this threat has access to the local network that the legitimate users are using to conduct the pairing, and may monitor, intercept, and duplicate packets arbitrarily. The threat may also be physically located with the users, something additional to current research on the topic. However, we assume that the attacker *cannot* do both simultaneously, such as intercepting packets on the local network while also being within earshot of the legitimate users. The threat may also have access to an AR device (e.g., a HoloLens 2), but does not need it to intercept the initially shared holograms. We believe that this is realistic. One bit of logic behind this assumption is that it is either similar to those used in existing pairing techniques [63] or more advanced.



(a) Attacker can observe shared holograms



(b) Attacker can eavesdrop on the vocal communication

Figure 6.1: If the attacker has access to the local network, as illustrated in (a), the attacker has the ability to intercept traffic at will and can understand the location of any holograms presented to each user. The attacker cannot, however, overhear or eavesdrop on any activity in the physical proximity of the users. If the attacker has access to the physical location of the pairing, as illustrated in (b), the attacker has access to the vocal communication between the pairing partners, but not the network traffic being passed amongst the pairing partners. Here, the attacker may have knowledge of what is said or done among the pairing partners, but does not have a way to use the passing network traffic to exploit this information.

Other techniques consider only that the attacker can be physically present [188] *or* that they can have access to network traffic [189], but do not consider these scenarios in the development of the same technique. Finally, we do not consider Denial of Service (DoS) attacks in this research. Such DoS attacks include limiting access to the local network, physically interrupting the attempted pairing, and destroying traffic for the sole purpose of disrupting pairing. Attack strategies are depicted in Fig. 6.1.

6.2.2 Design Goals

In order of priority, we design GAZEPAIR with five principles in mind.

- **Usability.** The GAZEPAIR user interface and required operations must be intuitive to novice AR device users. GAZEPAIR must not require the use of any advanced interaction techniques to navigate the user interface. The completion of a GAZEPAIR pairing must not take an excessive amount of the user's time.
- **Success rate.** GAZEPAIR must be effective, meaning that user pairing success rate must be 95% or higher.
- **Scalability.** GAZEPAIR must allow more than a one-to-one pairing and must be able to complete these pairings without excessive additional time requirements. Since keys generated from this pairing are symmetric, GAZEPAIR must be able to efficiently generate multiple matching keys.
- **Security.** GAZEPAIR must produce an acceptable level of security through the level of entropy from the shared secret created by harnessing eye gaze. We believe this level to be more than 60 bits of entropy [36]. GAZEPAIR must also be resistant to MITM attacks.
- **Device requirement.** GAZEPAIR should operate across the breadth of MR devices that

incorporate eye gaze tracking, and allow hardware-independent, efficient pairing for each. GazePair must be deployable on any such MR device quickly and with no software package conflicts, by simply changing the intended deployment platform in Unity.

We propose that eye gaze tracking can be harnessed to achieve each of the design goals above while improving on the shortcomings of current works noted in Section 2.3.2. However, there are unique challenges with using eye gaze for this purpose. They are outlined in Section 6.2.3 and are addressed in our GazePair design.

6.2.3 Key Challenges

As mentioned in Section 6.1, using eye gaze tracking for both user input and entropy generation involves inherent error. The underlying mechanisms to harness this gaze data once collected are non-trivial. Additionally, the user's uniquely identifying eye and gaze-tracking data must be protected. In the following, we discuss these key challenges in detail.

Challenge 1: Eye gaze tracking error. Eye gaze tracking has inherent error. Both research works and Microsoft's eye tracking documentation note a roughly 1.5 to 3-degree variance in expected gaze location compared to the recorded gaze location [85, 123]. This error only increases when the user moves and dictates the minimum size of eye gaze targets with the user stationary or in motion.

Challenge 2: The requirement for OOB communication. The ability of an attacker to intercept and exploit the location of a shared hologram led us to believe that OOB communication was necessary, similar to the authors of HoloPair [188] and Tap-Pair [189]. Additionally, direct communication with intended pairing partners also provides built-in authentication, similar to existing works as well.

Challenge 3: Eye gaze for symmetric key generation. While gaze data has been heavily studied for the purposes of authentication in MR devices [77, 95, 105, 106, 109, 187], it has yet to be studied for the pairing of AR devices in order to exchange information securely. To this point, gaze data has been identified as a rapidly increasing research area for securing Human-Computer Interaction (HCI) [86], but to our knowledge, mechanisms to create symmetric keys from eye gaze data do not yet exist. A system must be created that can take a user's eye gaze, discretize as required, and use this secret information to create a symmetric encryption key. This system must be robust enough to ensure a reasonable amount of security without creating a burden to the user.

Challenge 4: Data privacy. Eye gaze data can uniquely identify a user and has the potential to harm users if the collected data is misused. As such, techniques have been developed to remove the uniquely identifying traits from gaze data, if and when it needs to be transmitted in insecure environments [43]. A system using eye gaze tracking must ensure that the uniquely identifying biometric data is not leaked or transmitted in such a way as to jeopardize the user's privacy.

GAZEPAIR overcomes these challenges and creates a usable, scalable, and efficient pairing system using eye gaze. GAZEPAIR mitigates inherent eye tracking data by constructing holograms to meet the design specifications in [85] and creates a discretization mechanism to compensate for gaze tracking error. Additionally, GAZEPAIR implements a method to transform the gaze data into a shared secret and creates a symmetric encryption key to be used to protect further user communication across a local network. A spoken sequence cue is used to direct the users to the correct numbered holographic keys and is used as the OOB channel. The design of GAZEPAIR and the underlying pairing protocol are detailed in Section 6.2.4.

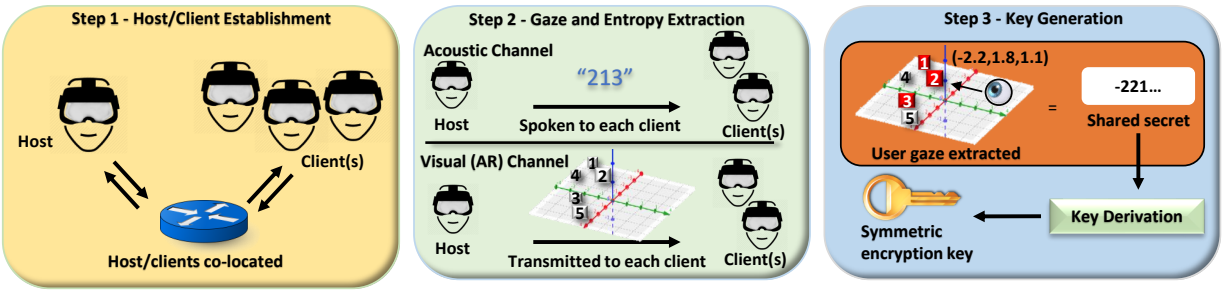


Figure 6.2: An illustration of GazePair's three main steps to key generation and device pairing.

6.2.4 Pairing Process of GazePair

In order to create the symmetric encryption key required to complete the pairing of the AR devices, GazePair operates in three main steps as shown in Fig. 6.2. We explain these steps as follows:

Step 1: Host/Client Establishment. GazePair creates a host/client relationship on a local network, e.g., through a User Datagram Protocol (UDP) connection. This is necessary to ensure that information (i.e., numerically-labeled holograms) are accurately shared between two or more users.

Step 2: Gaze and Entropy Extraction. The host randomly places ten numerically-labeled holograms (denoting digits from 0 to 9) within a certain visibility threshold. The size of the keys is determined by the eye gaze tracking error in *Challenge 1*, based on pilot testing. Specifically, this size is a compromise between large keys limiting generated entropy and small keys limiting the accuracy of eye gaze. The host then speaks[1] a *key sequence cue (KSC)* to each client. This process addresses the OOB communication challenge noted in *Challenge 2*. Each client selects the required numbers in sequence, dictated by the KSC, one at a time using eye gaze. The shared secret becomes the concatenated, discretized values of the 3D

[1]The KSC can also be written if users have significant obstacles to speaking and/or hearing the KSC.

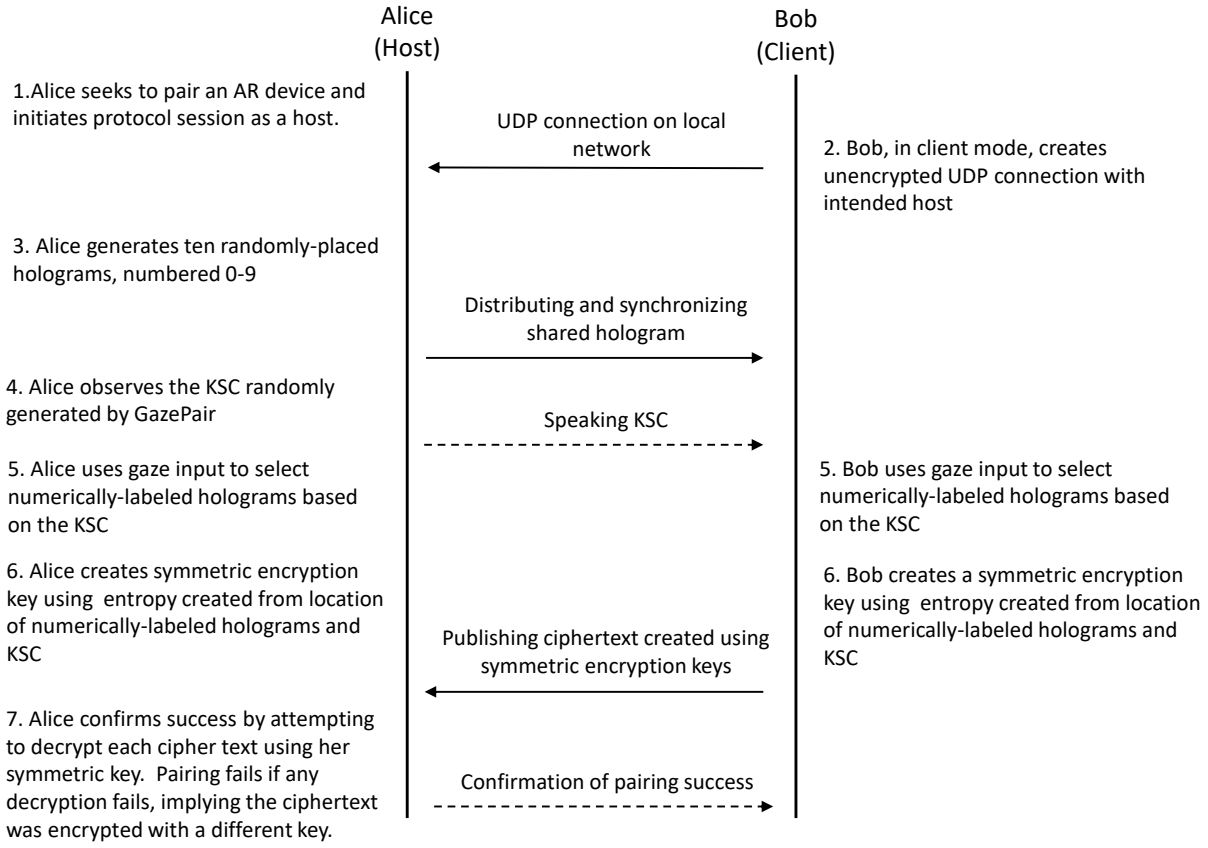


Figure 6.3: A depiction of the pairing protocol, using the classic Alice and Bob analogy. While there is only a single client (Bob) in this illustration, we show that GAZEPAIR is scalable to multiple clients easily.

location of the user's gaze when the user selects each numerically-labeled hologram. This process harnesses the user's eye gaze to obtain the entropy needed for creating symmetric encryption keys, while not requiring storage of any data that can compromise user privacy, which addresses *Challenges 3 and 4*.

Step 3: Key Generation. Key generation uses the shared secret to create a symmetric encryption key, e.g., an Advanced Encryption Standard (AES) key. This key is a function of the shared secret created by the gaze tracking data. Hence, the entropy of this shared secret must be as large as possible to create strong keys.

Next, we describe the pairing protocol. As shown in Fig. 6.3, a GazePair session begins with the host selecting their role in the initial GazePair connection scene (Step 1). This allows a client or clients to initiate the protocol appropriately, and establish an unencrypted UDP connection to the host (Step 2). With this connection established, the host randomly generates ten numerically-labeled holograms, numbered 0-9 (Step 3). These numerically-labeled holograms are randomly placed in the augmented 3D space, with limits. These limits are designed to ensure that the holograms are within a given visibility threshold. These holograms are placed relative to the position of each pairing participant and are not anchored to any one physical location. The client then receives these locations and reproduces the set of numerically-labeled holograms on the client's HMD. Then, the host observes the randomly generated KSC (Step 4) and speaks the KSC to the client. Each participant uses gaze input to select numerically-labeled holograms in sequence based on the KSC (Step 5).

If the gaze ray impact point is inside the discretization range of any hologram, that hologram turns red to indicate to the user the key has been successfully entered. The discretized values of the gaze position are then concatenated to generate the shared secret string independently on each HMD. Consider the example in Fig. 6.2. Suppose the KSC is "213". For the first numerically-labeled hologram selected (i.e., digit "2", the first digit of the KSC), a gaze location of $(-2.2, 1.8, 1.1)$ is captured and might be corrected to the string "-221" based on discretization parameters, and this becomes the first three digits of the shared secret. This continues until the required KSC length is met, and the protocol waits for each participant to complete this process. After completing the entry, each HMD independently creates a symmetric encryption key (Step 6) and encrypts a message using this key. The pairing protocol succeeds if and only if the host can decrypt each client's ciphertext (Step 7).

Remark 6.1. GazePair uses a KSC similar to a padlock combination or a keypad on a door. On the face of the fact, this sounds similar to a traditional passcode-based pairing instance

and may seem insecure or trivial. However, the KSC is of no value to any attacker without detailed knowledge of the location of the numerically-labeled holograms in 3D space.

Remark 6.2. It may seem that the use of gaze tracking to select the numerically-labeled holograms is extraneous. It is certainly possible to modify GAZEPAIR to use other standard AR (or generally MR) gestures and inputs. For instance, using the "hand ray" gesture, creating a simple ray from a user's extended hand, can be used to select the numerically-labeled holograms in sequence based on the KSC. While it is possible to do this, we lose the benefit of the obscuration of the user's eyes. With the HoloLens 2, the user's eye direction is greatly obscured by the plastic visor upon which the holograms are projected. Without this, an attacker could more easily understand the location that the user intends to select, aiding in deciding on the location and sequence of the numerically-labeled holograms being detected. This vulnerability has been well-documented in works such as [16], where researchers use channel state information to conduct side-channel key logging attacks against VR users with hand held controllers. To prevent this, GAZEPAIR uses the "air tap" gesture which does not require the gesture to be oriented toward the intended target in any way, unlike the "hand ray" gesture [128, 131]. We elaborate on the air tap gesture's utility in Remark 6.3.

6.3 Prototype Implementation

Using the GAZEPAIR design described above, we then seek to create a usable prototype to prove our hypothesis. Using the Unity development environment, Microsoft's Mixed Reality Toolkit (MRTK), and multiple HoloLens 2 AR HMDs, we create a working prototype of GAZEPAIR that advances the state-of-the-art with proven usability and scalability. GAZEPAIR creates 64 bits of total entropy in the shared secret while not exposing the user's uniquely identifying eye data to potential misuse. Additionally, GAZEPAIR is deployable on any MR

device capable of eye gaze tracking. In the following, we first briefly introduce the Microsoft's MRTK (Section 6.3.1), and then describe the detailed implementation of GazePair prototype in three main steps: Host/Client Establishment (Section 6.3.2), Gaze and Entropy Extraction (Section 6.3.3), and Key Generation (Section 6.3.4).

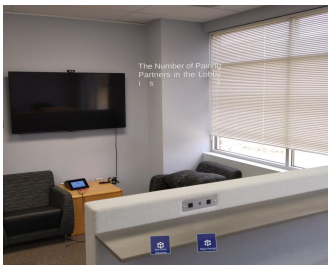
6.3.1 Microsoft's Mixed Reality Toolkit

Created and maintained by Microsoft, the Mixed Reality Toolkit (MRTK), is an attempt to standardize the development of MR applications [135]. MRTK is at the core of the HoloLens, and its successor, the HoloLens 2. Additionally, it is used to design applications for Meta's Oculus series of MR devices, as well as HTC's Vive, and other Windows Mixed Reality headsets. Most importantly to this research, it also contains the APIs used to access the HoloLens 2's powerful eye-tracking technology. As we begin to explore the possibility of using eye gaze to pair multiple MR devices, the deployability of this prototype on MR devices depends on the use of MRTK for eye tracking, but the design itself is usable on any MR device incorporating eye gaze tracking. At this moment, only the HoloLens 2 incorporates eye tracking using MRTK, but as this expands, so does any pairing solution created with MRTK's eye-tracking APIs, including any and all MR devices using this specific capability.

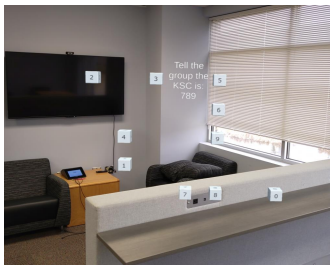
6.3.2 Host/Client Establishment

Mid-level API (MLAPI)

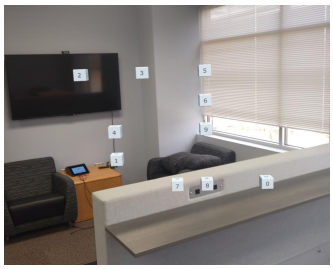
In order to distribute these numerically-labeled holograms, a networking suite must be used to create and synchronize these holograms. For GazePair, we choose a Unity-developed API called Mid-level API (MLAPI) to create this functionality. MLAPI is open source and



(a) Host/Client identification



(b) Host's view of the pairing



(c) Client's view of the pairing

Figure 6.4: An illustration of Steps 1 and 2 of GAZEPAIR. (a) shows the first scene where the host and client select their applicable roles and the pairing protocol establishes their relationships. (b) and (c) show the view of the pairing scene from both the host's and client's points of view, respectively.

simply seeks to abstract transport layer functionality to ease the integration of networking functionality into Unity-created applications. MLAPI allows the user to easily declare their intent to initiate a session as a host or a client, and function accordingly. While MLAPI communicates to clients without encryption, the information we share publicly (i.e., the location of numerically-labeled holograms) is not sufficient to breach the security of GAZEPAIR. After confirmation of matching symmetric Advanced Encryption Standard (AES) 256-bit keys, users can use these keys to encrypt any communication desired.

Host/Client Identification

GAZEPAIR creates a host/client relationship between two or more HMDs. For testing, we hard-code the IP address of the host on all HoloLens 2 HMDs, and make a single HoloLens 2 HMD the dedicated device for initiating the pairing procedure, to ensure mistyped IP addresses do not alter the testing data. In future implementations, either method can be used. Fig. 6.4a shows the view of the host and the client in this stage.

The host, and the host only, can monitor the number of clients registered by MLAPI as present in the pairing lobby. This serves to ensure that the host has made a connection

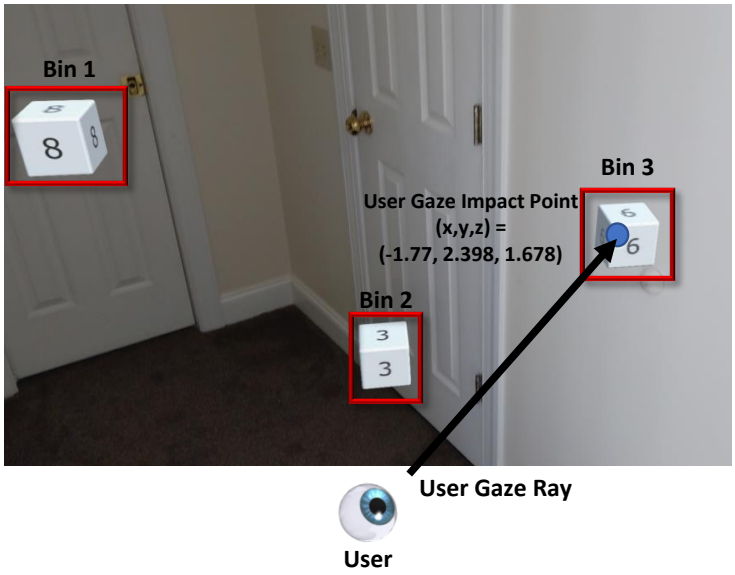


Figure 6.5: An example of the eye gaze discretization technique used in GazePair. After a KSC is spoken to each client from the host, each user finds the required numerically-labeled hologram, gazes at it, signals the protocol to take a snapshot of the gaze location, and moves on to the next required hologram. Each gaze sampling is discretized and becomes a part of the shared secret. In this example, the first gaze sampling is taken when the user gazes at the hologram labeled "6". The raw gaze value is (-1.770, 2.398, 1.678), which is then discretized to "-222", based on the discretization parameters, and added to the shared secret. The length of this value will increase as the user takes more gaze samplings directed by the KSC.

with the number of clients they expect, and also serves to ensure that, if clients additional to the number expected join the pairing instance, the host is informed. This will not stop a surreptitious MITM-style attack but will mitigate the risk of a client accidentally joining the incorrect pairing instance by allowing the host visibility of how many clients are connected to their pairing session.

6.3.3 Gaze and Entropy Extraction

After the creation of the host/client relationship between two or more HoloLens 2 devices, the host randomly distributes 10 numerically-labeled holograms (denoting digits from 0 to 9) within a given visibility threshold. These numerically-labeled holograms are placed no closer to each other than twice the distance of a given error threshold in any direction. Attention is paid to ensuring that these numerically-labeled holograms are created in user-friendly locations, and not on top of users, behind users, or too far from users. An example of this from the host's and the client's viewpoint is shown in Figs. 6.4b and 6.4c, respectively. The host then speaks a randomly-generated KSC of a given length. Each user, independently, selects the numbers specified by the KSC, one at a time using eye gaze. The selection is done using MRTK's GazeProvider APIs [126, 135], to access a gaze ray, from the user's eyes to the hologram, striking at a given point. Once the user's gaze is placed on the intended hologram, the user takes a snapshot of the current gaze location by using the "air tap" gesture [128].

Remark 6.3. The "air tap" gesture is used jointly with the user's gaze location to harness gaze data. This gesture requires a user to "pinch" in the air to signal the device to select an eye gaze target for action. This gesture does not require the user to direct this "tap" at any given hologram, only that the gesture is visible to the HoloLens 2's Articulated Hand Tracking (AHAT) short-throw camera [197], which has a field of vision several feet outside what the user can see through the HoloLens 2's visor.

As part of the prototype testing, we also implemented a gesture-less method of gaze collection. This method asks a user to simply keep their gaze on an intended hologram, and after an elapsed period (e.g., two seconds), GazePair records the gaze position and discretizes this portion of the shared secret. This method is also effective, but vulnerable to novice AR users staring too long at a given hologram while they are learning the GazePair system. We have chosen the "air tap" as it is a way to ensure an extremely low error rate in gaze collection, improving the pairing success rate.

The user's gaze location is then discretized shown in Fig. 6.5 to allow for effective shared secret generation and error correction. Given an error threshold, GazePair creates a grid of 3D spaces covering all possible hologram locations. Since the host generates the shared holograms at specific locations at the center of these "error cubes", and given the error threshold, it is not possible for a user to select a given hologram and produce the discretized location of another. This allows GazePair to ensure discretization within a given threshold without having to publicly exchange discretization parameters as in Tap-Pair [189]. Each discretized value becomes a part of the shared secret, and each value is concatenated to become the final string used as the input to the Password-Based Key Derivation Function v2 (PBKDF2) for symmetric key generation. If the users enter an incorrect key, the shared secrets and symmetric keys will not match, and the pairing will fail. This shared secret, once successfully generated, is kept local to each device and will never be shared across the local network.

Remark 6.4. During the pairing process, users are allowed and encouraged to move freely around the room. There is no requirement in GazePair to sit, stand, or remain stationary. In fact, some users find it more entertaining to alter the way they view the numerically-labeled holograms, while some prefer to sit. Either is acceptable and did not alter GazePair's efficiency or effectiveness during evaluation.

6.3.4 Key Generation

After the establishment of the shared secret, each instance of GAZEPAIR creates a 256-bit AES key using the shared secret as an entropy source. A well-known and well-established technique to generate symmetric keys from passwords or other shared secrets is Password-based Key Derivation Function v2.1 (PBKDF2), as identified in IETF RFC 8018 [141]. While known vulnerabilities exist in this method of key generation, including vulnerability to rainbow table attacks using advanced Graphics Processing Units (GPUs), GAZEPAIR requires potential pairing partners to commit to a single symmetric key during the pairing process, using unique and random initialization vectors and salt values, mitigating the problem of offline rainbow table attacks.

PBKDF2

Accepted by the IETF in 2000 and updated in 2017, PBKDF2 seeks to protect a relatively low-entropy secret used to create keys, namely a password or other shared secret [141]. To do this, PBKDF2 hashes a password input with an initialization vector and adds artificial computational work to make the specter of a rainbow table or dictionary attack more difficult. This computational work, similar to Bitcoin's Proof of Work concept for validating blockchains [39], is intended to increase the computational cost of hashing all possible passwords a user might input. If the attacker knows the initialization vector, they must then begin the computationally expensive process of generating all possible values of the password and wait the time required for the artificial computational work defined by the number of iterations. As recently as 2021, PBKDF2 has been proposed for sensitive systems, such as cyber-physical systems, with modifications to increase the computational resources required to generate password digests [112]. For GAZEPAIR, we choose to use the SHA256 hashing

function for PBKDF2 due to its collision resistance and high security, and a total of 50,000 iterations. We choose 50,000 iterations as it is a number that creates no noticeable performance degradation on the HoloLens 2, while larger values create noticeable and unacceptable degradation.

Both a salt and initialization vector are required for PBKDF2 key generation and AES encryption and decryption. Using the Microsoft .NET pseudo-random number generator [127], the host generates a 64-bit random number used as the salt and initialization vector. These values are published to all clients, ensuring that the salt and initialization vector are both random and known to all.

Analysis of Generated Entropy

Using the accepted method of calculating password entropy, denoted by E, from the total number of shared secret possibilities, denoted by S, specifically we calculate entropy as follows:

$$E = \log_2 S. \tag{6.1}$$

Assume a total number of K numerically-labeled holograms. Each hologram has an (x, y, z) value on the 3D Cartesian plane, and these values are limited to the value pool, i.e., the total possible values of each axis, denoted by X_t, Y_t, and Z_t. Each (x, y, z) value is non-repeating for each combination to prevent two numerically-labeled holograms from spawning in the same location. We use the permutation below to solve for the total number of possible key arrangements of the total K numerically-labeled holograms, denoted by N_K:

$$N_K = \binom{X_t \cdot Y_t \cdot Z_t}{K} \cdot K! = \frac{(X_t \cdot Y_t \cdot Z_t)!}{(X_t \cdot Y_t \cdot Z_t - K)!}. \tag{6.2}$$

The numerically-labeled holograms selected are dictated by the order and value of the KSC, which is generated by GAZEPAIR and spoken by the host. The number of numerically-labeled holograms selected is the length of the KSC, denoted by P. Since the KSC has non-repeating digits (to aid in input and error detection), the total number of KSCs, denoted by N_P, can be calculated as follows:

$$N_P = \binom{K}{P} \cdot P! = \frac{K!}{(K-P)!}. \tag{6.3}$$

Hence, the total number of shared secret possibilities is $S = N_K \cdot N_P$. Plugging it into Eq. (6.1), we calculate entropy E as follows:

$$E = \log_2 S = \log_2 (N_K \cdot N_P). \tag{6.4}$$

Specific to GAZEPAIR, we calculate entropy slightly differently, as all the numerically-labeled holograms are on the same Z-axis to ensure visibility, and thus, there are fewer permutations. We also use a KSC length of 3 digits, further explained in Section 6.5. Additionally, we do not spawn numerically-labeled holograms at (0,0) on the (x, y) planes, so as to not obscure the host's view of the KSC prompt. In GAZEPAIR, we use $X_t = 7$, $Y_t = 6$, and $Z_t = 5$. We also have a total of 10 numerically-labeled holograms, 0-9, so $K = 10$. We have $P = 3$, as the KSC has 3 digits, which are non-repeating. Hence, we can calculate the following:

$$\begin{aligned}
N_K &= \binom{X_t \cdot Y_t - 1}{K} \cdot Z_t \cdot K! = \frac{(X_t \cdot Y_t - 1)!}{(X_t \cdot Y_t - 1 - K)!}. \\
Z_t &= \frac{(7 \cdot 6 - 1)!}{(7 \cdot 6 - 1 - 10)!} \cdot 5 = 2.034 \times 10^{16}, \\
N_P &= \frac{K!}{(K-P)!} = \frac{10!}{(10-3)!} = 720 \text{ (9 bits of entropy)}, \\
E &= \log_2(N_K \cdot N_P) \approx \mathbf{64} \text{ bits of entropy.}
\end{aligned} \tag{6.5}$$

6.4 Experimental Evaluation

One-to-one pairing tests take an average of 9.02 seconds, with a 98.3% pairing success rate. One-to-two pairing tests take an average of 12.58 seconds, with a 96.6% success rate. This is accomplished using a prototype that is deployable on the breadth of AR devices that use eye gaze tracking and MRTK, and with a design that is deployable on the breadth of MR devices using any form of gaze tracking.

6.4.1 Software and Hardware Setup

We built the GazePair prototype in Unity 2020.3.16f1, using MRTK version 2.7.2 and MLAPI version 0.1.0. GazePair was deployed on Microsoft HoloLens 2 HMDs running Windows Holographic for Business Build 20348.1438. Blender version 3.0.1 was used to create the custom numerically-labeled holograms, and all source code was completed in C#. Our prototype is posted as open source [34].

6.4.2 Evaluation Design

The GazePair prototype was tested by 20 participants with varying ages, vision capabilities, and technical backgrounds. The entirety of the evaluation was approved by the University's IRB. Participants were recruited using the University's graduate student email distribution list, or from the local, non-student population. Upon arrival at the test site, each participant signed a consent form and completed a basic demographic questionnaire. Each user was then given a brief (ten minutes) tutorial on the basic operation of the HoloLens 2, including adjustment of the fit of the HoloLens 2 HMD, accessing the main menu, AR gestures, and operating GazePair. Additionally, we asked each user to complete an eye calibration, as

suggested by Microsoft, for each user before every set of tests. The participants were allowed to remain stationary during the tests, or move freely, as they felt comfortable. For each group of participants, a host was randomly selected and given the HoloLens 2 with the hard-coded IP for the host, while the other participants served as the client. All tests were conducted over a Netgear AC1750 router without an internet gateway, to ensure that all tests were over a local network. Additionally, at the end of the testing session, the participants completed a usability survey, with responses on a 5-point Likert scale ranging from "Strongly Disagree" to "Strongly Agree" [117].

Below are the metrics measured for each test iteration.

- **Total time required to complete pairing.** The GAZEPAIR logging script [177] begins a timer at the moment that the host is satisfied with the number of participants in the pairing lobby and begins the pairing protocol. From that moment, the elapsed time is measured until the completion of the KSC entry of the final participant to finish the sequence.
- **Success rate.** GAZEPAIR records the boolean value that corresponds to the host's ability to decrypt messages from all clients in the pairing protocol. If any message cannot be decrypted by the host, the pairing protocol is considered a failure, and the host is notified.
- **Number of pairing partners.** The logging script records, for each test, how many participants were present for the pairing session.

Demographics

Our testing consists of 6 self-described women and 14 self-described men, only four of which self-profess any experience with AR devices. The ages of the participants range from 20 to 72. Users have a wide variety of vision capabilities, ranging from perfect vision without

glasses or contacts to impaired with vision correction. Users come from backgrounds that vary from computer science graduate students, military service members, to retirees.

Evaluation Metrics

Additionally, we chose to evaluate the ability of GazePair to satisfy its design objectives with a series of tests and evaluations. Specifically, we evaluate the GazePair design using the following experiments.

- **Success Rate.** We evaluate GazePair's effectiveness by analyzing the number of successful tests against the total tests, and the reasons for each failed test. This is done for both one-to-one and one-to-two pairing attempts.
- **Usability.** We evaluate the time required to complete a pairing iteration in order to establish evidence of its efficiency. We discriminate between one-to-one and one-to-two pairing attempts.
- **The usability over time.** As a hypothesis, we expect usability to increase as users increase their familiarization with the system. We present pairing times, compared against the number of pairing attempts completed, as a way to analyze this.
- **GazePair vs. existing solutions.** We evaluate GazePair qualitatively and quantitatively against all known AR pairing solutions.

6.4.3 Pairing Performance

With 240 tests, 97.5% of pairings complete successfully. The six tests that failed were due to misselected numerically-labeled holograms, either selected in the incorrect order or due to misunderstanding the KSC. *For one-to-one pairing attempts, 118 of 120 (98.3%)*

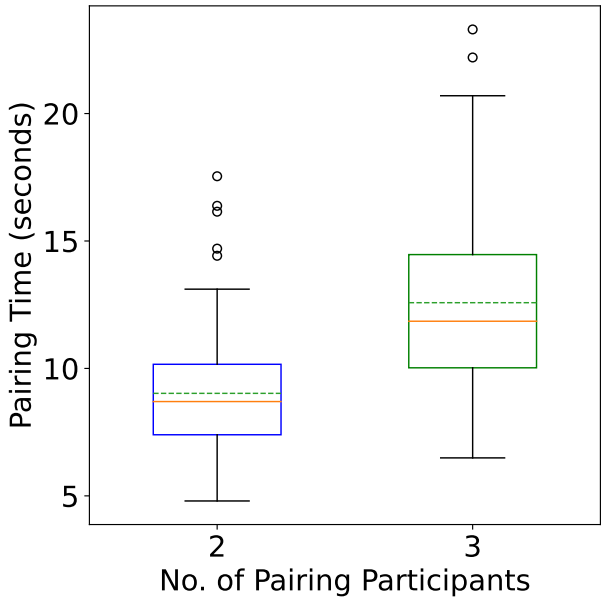


Figure 6.6: A boxplot of the total pairing time, by the number of pairing partners. The median time is shown with the solid, horizontal, orange line. For one-to-one pairing, the median time is *8.7* seconds; for one-to-two pairing, the median time is *11.83* seconds. The average time for each is represented by the dashed, horizontal, green line. The average time for one-to-one pairings is 9.02 seconds; for one-to-two pairings, the average time is 12.58 seconds.

are successful, while 96.6% of the 120 one-to-two pairing attempts succeed. The one-to-one pairing performance is an 8% improvement on the most current and advanced AR device pairing technique, while the 96.6% pairing success rate of the one-to-two tests shows GAZEPAIR *'s scalability.*

Additionally, we compare the times to complete a one-to-one pairing with the length of the KSC. Fig. 6.7 shows both the average time to complete a pairing attempt at each possible KSC and also the resulting entropy. This shows that the pairing time generally increases as the length of the KSC increases. We also note that as the KSC began to increase, participants were no longer able to speak the KSC in a single phrase and began to divide the KSC up into parts.

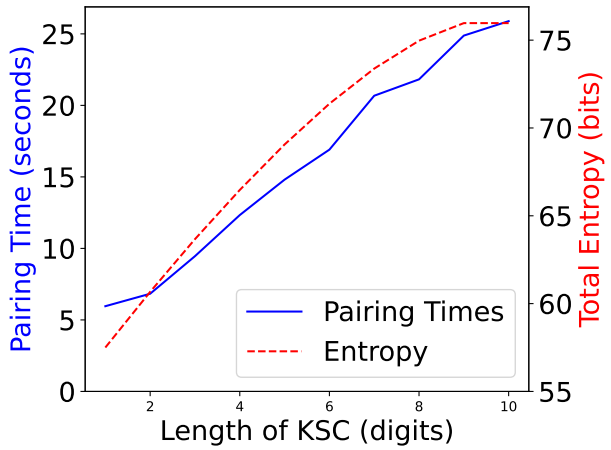


Figure 6.7: An illustration of the tradeoff between the length of the KSC, the time required to complete one-to-one pairings, and the generated entropy. Note the generally linear increase in pairing times relative to the KSC length.

6.4.4 Comparison with State-of-the-Art Research

In each of the discussed AR pairing solutions, users are required to be wearing an AR device (and only an AR device). Additionally, no solution has been proven to be usable to pair more than two devices. All three existing systems require users to see each other or interact with the physical environment, limiting the system to AR applications. Also, other methods in the field of authentication exist, such as [64]. Like Tap-Pair, these systems leverage the device user's understanding of the physical and digital world in order to create or validate some secret. However, this technique is shown to have low entropy (about 12 bits) and high failure rates similar to Tap-Pair. Like two of the three existing solutions, GazePair also requires an OOB communication channel for pairing. However, GazePair only requires that the pairing initiator speak a key sequence cue of an arbitrary length, something that requires much less motion than the tracing/waving or walking/pointing required from HoloPair and Tap-Pair respectively. We improve on these solutions by designing, implementing, and testing GazePair, a novel pairing system that is user-friendly and scalable. GazePair extends to more than two users and uses eye gaze for entropy generation, a technique that simply

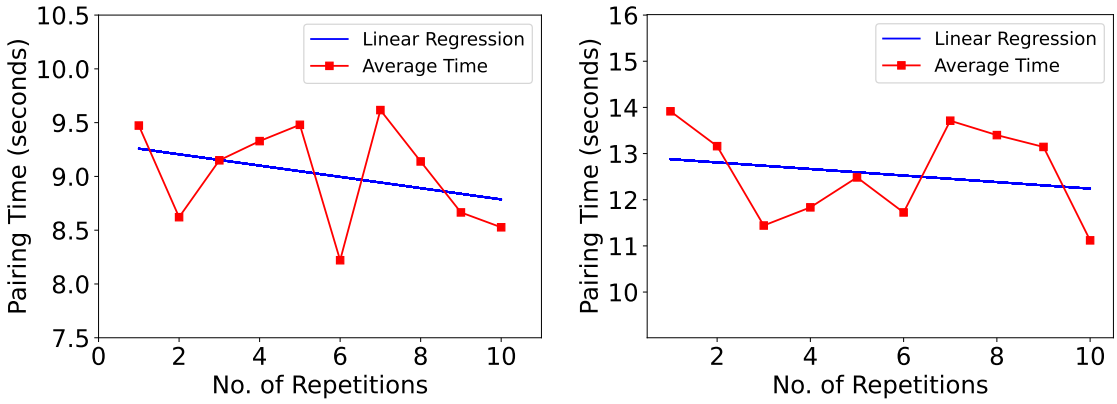
Method	Interaction Technique	Success Rate	Pairing Time	Scalability	Security	Device
LGTM [63]	No out-of-band communication	58.4%	Not reported	Unproven	Relies on wireless localization, entropy not reported	AR only
HoloPair [188]	Tracing & waving	98%	10-11 seconds	Unproven	Vulnerable to inattentiveness, 53 bits of entropy	AR only
TapPair [189]	Head direction and tapping	90%	Not reported	Unproven	9-11 bits of entropy	AR only
GAZEPAIR (this work)	Spoken key sequence cue & eye gaze	98.3%	9.02 seconds	2+ user pairing proven	64 bits of entropy	MR with eye gaze tracking

Table 6.1: A comparison of GAZEPAIR with state-of-the-art AR device pairing solutions.

requires users to direct their eyes at a holographic target. Eye gaze requires very little movement from the users, is partially obscured from bystanders due to the opaque visor on most AR devices, and is shown during evaluations to be easy to learn and understand. Table 6.1 presents a comparison of our proposed GAZEPAIR system with these existing AR pairing solutions.

6.4.5 Usability

Fig. 6.6 presents a box plot of all the pairing tests conducted, by the number of pairing partners. *The average time to complete one-to-one pairings is 9.02 seconds*, from the host initiating the pairing protocol to the host's successful decryption of each client's ciphertext using the host's AES key. For one-to-two pairings, the average pairing time increases to 12.58 seconds. We believe that this shows the efficiency of the GAZEPAIR system, as measured by user time requirements. GAZEPAIR matches or improves on the pairing time of the



(a) One-to-one pairing (b) One-to-two pairing

Figure 6.8: The effect of multiple repetitions on the pairing time shown by average time and a linear regression of the time required for sequential tests. (a) shows the results of participants in one-to-one pairing. (b) shows the results of participants in one-to-two pairings. The generally downward trend in the pairing time indicates that as users begin to become familiar with the system and AR gestures, the system becomes more usable.

three known local AR pairing solutions. As noted in Remark 6.3, we also implemented a gesture-less version of GazePair using gaze dwell as the trigger to collect the gaze data. This technique is potentially even more user-friendly, but the likelihood of novice AR users accidentally selecting the incorrect holograms while learning AR gestures increases. While this would be lessened with experience, we choose to use the "air tap" due to the much lower occurrence of mis-selected numerically-labeled holograms by novice AR users.

Additionally, we evaluate the usability of the system based on the feedback from participants at the end of the testing based on the 5-point Likert scale. Of the 20 participants, 95% stated that they "Strongly Agreed" or "Agreed" the system was easy to use, and that they believed most users would learn the system "quickly"; 90% responded that they "Strongly Disagreed" or "Disagreed" the system was cumbersome. We believe that this user feedback further reinforces the usability of the GazePair design.

6.4.6 The Usability Over Time

Fig. 6.8 presents the time required to complete a test, regardless of pairing success, over multiple pairing iterations. The generally downward trend of the pairing time, as the familiarity with the system increases, shows the effect of experience with the system. As the participants spend more time with GAZEPAIR, their familiarity with AR increases, and they are able to more easily complete pairing attempts. Specifically, the average pairing time for the first attempt at a one-to-one pairing is 9.47 seconds, falling to 8.52 seconds by the tenth. Similarly, the average first attempt with three participants in a one-to-two pairing is 13.9 seconds, falling to 11.09 seconds by the last attempt. Given the relative inexperience using AR of the participants, this appears to be expected. Additionally, we believe this speaks to the usefulness of eye gaze. Users, regardless of AR experience, intuitively understand how to gaze at an object. No testing participant expressed confusion over the use of eye gaze.

6.4.7 GAZEPAIR vs. State-of-the-Art Solutions

Table 6.1 gives a comparison of GAZEPAIR to the state-of-the-art research into AR pairing. LGTM is useful as a baseline solution, but it is not usable on any known AR device given its requirement for wireless localization, a requirement that also contributed to its low success rates. Additionally, its facial recognition will be frustrated by AR devices with visors obscuring the user's face. HoloPair's success rates and time requirements are on par with GAZEPAIR's, but HoloPair is hampered by its requirement for tracing and waving between two users and the ability of these users to simply "skip" the pairing attempt. Additionally, HoloPair's generated entropy is about 26 bits in its most complex variation, something the authors claim is "not large". For usability, GAZEPAIR and Tap-Pair are roughly equivalent. Both require user gestures. TapPair requires a user to direct their head toward a place desig-

nated by the pairing initiator, designated by a finger or post-it, and "tap". GazePair requires vocal communication as well as using the "air tap" gesture. Also, similar to Tap-Pair, the authentication system in [64] leverages the same insight as our system - AR users have their unique knowledge of the virtual objects in the scene, but also produces low entropy and high failure rates in evaluation scenarios. Compared with Tap-Pair and similar authentication systems, GazePair *increases the success rate of pairing attempts by 8%, and quantifiably proves its scalability with more than two users, something only hypothesized in Tap-Pair.* GazePair also improves on the 9-11 bits of total entropy generated by Tap-Pair, by adding randomness in the location of holograms and in the choice of the KSC. In addition, the work in [64] uses physical controllers, which allows attackers to observe the input behaviors and guess the password (in 12.5% to 18.5% of attempts). Differently, we use eye gaze, which is invisible to attackers, so we can provide better defense against an attacker with the ability to observe the pairing. Finally, GazePair's design is applicable to the breadth of MR devices that use eye gaze tracking, and the prototype itself is deployable on any device harnessing eye gaze using Microsoft's MRTK, something no other solution claims.

6.5 Discussion

6.5.1 Availability of Gaze Tracking Sensor

The gaze tracking sensor is essential for supporting GazePair. Through a survey of documentation related to current or upcoming AR HMDs, we find that most of the popular MR headsets already (or will) support gaze tracking. For example, the AR headsets of Magic Leap, Magic Leap 1, support gaze tracking in their current model, and this feature will continue to be available in their upcoming new models [191]. In addition, Meta (formerly

Facebook) earlier stated that they would include sensors for both face and eye tracking on their Oculus Quest Pro [76]. This shows that the gaze tracking sensor has already been and will continue to be essential hardware on future MR headsets, indicating that GAZEPAIR can be easily implemented on future MR devices without introducing any extra overhead.

6.5.2 Length of the KSC

In the GAZEPAIR evaluation, we used a KSC length of 3 digits. This was a compromise between the time required to input the KSC and the additional entropy provided by longer KSC lengths. For example, at a KSC length of 10 digits, GAZEPAIR can provide 76 bits of entropy. We chose 3 digits, as this is the smallest KSC that provides entropy deemed to be "strong" [36] while remaining usable in terms of pairing time compared to existing solutions.

6.5.3 Applicability to the Metaverse

As discussed in Section 6.1, the advent of the Metaverse is a motivational factor behind the expansion of AR. Large corporations, such as Meta, envision this digital collaboration space as the future of work and entertainment [37]. Certainly, the Metaverse could affect our daily lives in the near future. However, these large environments are envisioned to be hosted and implemented on centralized servers, likely under the control of large corporations such as Meta. Data collected can include visual and audio recordings of the user, iris data, body type, and movement style. What if a small organization, such as a social group, does not wish to have data like this transferred across these large servers in far-away places? Such small groups could use ad-hoc, local, secure pairing techniques such as GAZEPAIR to transfer spatial data, visual and audio data, and the like while not having to worry about corporate data collection or misuse.

6.5.4 GazePair's Performance Under Threat

To clarify GazePair's performance under attack, we discuss the ability of an attacker to compromise the security offered by GazePair in each of the proposed threat conditions. If the attacker has access to GazePair's network traffic, they can use this information to deduce the location of each numerically-labeled hologram and the IV/Salt values. This is the most advantageous attack vector as the only barrier to compromise of the symmetric encryption keys is the knowledge of the KSC. Given a KSC length of 3, the attacker has a 0.1% chance to correctly guess the KSC, or a total of 9 bits of overall entropy as outlined in Section 6.3.4. This chance is as low as 0.00003% with a KSC length of 10.

In the second threat condition, where the attacker is co-located with the pairing participants but does not have access to GazePair's network traffic, the challenge for the attacker increases. As outlined in Section 6.3.4, GazePair randomly generates all 10 numerically-labeled holograms in three dimensions. First, GazePair chooses a randomly selected z value (i.e., depth) from 5 possible choices. Then, all 10 numerically-labeled holograms are randomly generated on a 7×6 grid at this depth. Assuming a KSC length of 3, the attacker would need to guess from 344,400 possible combinations of holograms. Even assuming this correct guess, the attacker does not know the IV/Salt values shared between devices, making the knowledge of the shared secret irrelevant.

As mentioned in Section 6.2.1, we assume that the attacker cannot simultaneously have knowledge of the KSC, the location of the numerically-labeled holograms, and the IV/Salt values. If this were not the case, GazePair's security could potentially be compromised.

6.5.5 Limitations of GazePair

We have shown that GazePair is an efficient, effective, and safe AR device pairing technique. Even so, there are certainly areas for improvement. We summarize GazePair's limitations as follows.

- **KSC entropy.** With the 3-digit KSC used during testing, the entropy of the KSC itself is only 9 bits. Even with PBKDF2 creating artificial work and increased computational time requirements to hash all KSC combinations, an attacker could potentially use the knowledge of the salt and initialization vector to test all possible key combinations against the known hologram locations if users do not use the generated symmetric encryption keys to transmit another stronger key. As shown in Section 6.4.5, with an increased KSC length, while the entropy level improves, the pairing time also increases.
- **Out-of-band communication.** GazePair requires a spoken KSC between the host and the clients. As mentioned in Section 6.4, even hearing the KSC without knowledge of the Salt/IV or placement of the numerically-labeled holographic cubes does not give the attacker an advantage in this design. However, not all users may understand these technical details, and the act of speaking the KSC could indeed generate a perception of an insecure system, especially when the KSC is exchanged in public places. We believe that the potential misunderstanding about the safety of the system in the presence of other people/listeners could be alleviated with additional information about the high-level technical details. For example, if the users understand that the KSC cannot be used to compromise their pairing session unless the person overhearing their spoken communication can also intercept the Salt/IV and location of the individual holograms in 3D space, they may be much less concerned about this technique.
- **Expansion of paired group.** GazePair has no way to add participants to an estab-

lished sharing group without completely re-attempting the GazePair protocol. Keys are symmetric and generated uniquely per pairing attempt. If a user were to wish to participate in a given pairing group and was not present for the initial protocol, a new protocol instance would need to be initiated with current and new pairing partners.

6.5.6 Future Work

As noted above, GazePair's design still has areas for improvement. One potential area to improve on the entropy and out-of-band communication limitations is with spatial anchors [122]. Spatial anchors can allow multiple devices to not only see shared holograms but to see them in nearly the exact same location in the physical world, something not required by GazePair. Even so, transferring spatial anchors involves significant overhead as mentioned in Section 6.1. Spatial anchors can be very large, and transferring them locally can be difficult. However, these anchors can allow users to interact with shared holograms at absolute locations, creating an opportunity to use this for new and more intuitive methods to pair devices without using a KSC. Specifically, these anchors can potentially allow users to physically locate the intended pairing partners in a room, but without the wireless localization requirement of HoloPair. This enables new ways to authenticate users by verifying their physical location. However, optimizing the transfer of spatial anchors across a local network should be a subject of future research to make this technique more viable and efficient.

6.6 Summary

In this paper, we designed a novel AR device pairing system, GazePair, which leverages eye gaze tracking, a new and powerful AR technology. GazePair achieves efficient, secure, and

scalable local AR device pairing with minimal user interaction while protecting user gaze data. This requirement has become increasingly self-evident as the prevalence of AR devices increases, and the need to share holographic information locally becomes more pressing. Using Microsoft HoloLens 2 HMDs, we implemented a prototype system of GazePair that allows two or more users, without an internet connection, to locally pair AR devices quickly and intuitively. Through experimental evaluation, we showed that pairing two or more AR devices using entropy created from gaze tracking can be efficient. Remarkably, users with minimal AR experience were able to efficiently and effectively use GazePair to pair two or more AR devices, achieving both a high success rate and a low pairing time that are an improvement on all existing solutions for AR device pairing. Furthermore, all of these can also be done without using techniques that limit the deployability of GazePair to a small subset of MR devices or jeopardizing user biometric data.

With the advent of the Metaverse and future potential applications, such as in the military, healthcare, education, entertainment, and automotive manufacturing, the ubiquity of MR devices is only likely to increase. Techniques like GazePair are the building blocks to enable easy and efficient adoption of these MR devices by the broadest range of potential users in the broadest range of emerging MR applications.

Chapter 7

Conclusion and Future Work

7.1 Summary of Key Findings

In the three core works of this dissertation, we proposed three solutions to individual security and privacy problems in augmented reality. Each of these solutions uses human eye gaze to accomplish their purpose. Below, we present the major findings from each solution.

BystandAR:

- Our work identified eye gaze and the user's voice as an effective indicator of the salience of human faces in the device user's field of view. These insights allowed us to create BystandAR, a first-of-its-kind solution to the AR BPP.
- BystandAR leverages a frame-skipping interval to make this real-time solution feasible on modern AR devices. Similar solutions would burden the system without this method and create frame rate reductions that could reduce user satisfaction or increase motion sickness.
- Our evaluation shows a 98.14% bystander protection rate for BystandAR. This improved on existing offline methods by 12%. This system also allows access to 96.27% of subject faces. This ensures the user has access to salient faces for legitimate use cases.

- BystandAR maintains a frame rate of 52.6 FPS, showing a minimal impact on system resources.

ShouldAR:

- ShouldAR is a first-of-its-kind system that uses AR and multimodal eye gaze tracking to detect an SSA against the user. While most existing solutions can only protect against a single form of attack (e.g., on a mobile device or at an ATM PIN pad), ShouldAR can detect a wide variety of attacks in multiple scenarios.
- This system improves user comfort and can detect 87.28% of SSAs in writing and typing scenarios. This is a two-fold increase in the success rate of an ATM mirror, a commonly used solution to the SSA problem.
- ShouldAR works in real-time and has no detectable impact on the performance of the AR system used to build the prototype.
- Evaluation participants determined this system to be usable and intuitive. Participants also stated they would use this system in future AR devices.

GazePair:

- GazePair is the first local pairing solution to use AR and eye gaze. The system produces 60+ bits of entropy in pairing, creating a secure pairing method.
- This system improves on existing local pairing methods by successfully pairing 98.3% of devices in two-user tests in an average of 9.02 seconds.
- GazePair also can pair more than two devices, in a first-of-its-kind evaluation. These pairings were completed with a 96.6% success rate in an average of 12.58 seconds.

- The system is also usable across the breadth of mixed reality devices that include eye tracking.

7.2 Future Work

While the contributions above represent advances in the state-of-the-art, more work is required to adequately address the breadth of security and privacy challenges in AR. We provide an overall vision for these issues in the future and suggest additional work to move the state of the art toward this vision.

7.2.1 A Vision of the Future of Security and Privacy in AR

As AR devices become more usable and ubiquitous, it appears inevitable that they will become more common in public places. Just like cell phones [154], as these devices become better understood and more widely used, the social resistance to their use in public places is expected to lessen. In a world where these devices are expected in public places, we can imagine a future where the danger and the benefits of these devices increase exponentially.

Future privacy solutions cannot be designed as standalone solutions. These solutions must be integrated into the device at the system level. System designers cannot allow information that can compromise the user's privacy to be used without first carefully understanding the context and potential dangers of its use. Future security solutions must be able to distinguish between SSAs and legitimate "over the shoulder" collaboration.

7.2.2 The Future of Bystander Privacy in AR

In addition to addressing the lack of a technical solution to address bystander privacy concerns, research must also address the more nuanced areas of public education and perceptions. As shown in previous work, there is simultaneously a lack of understanding of the threats posed to bystanders and unnecessary apprehension towards devices that do not pose a threat [149]. To make future MR devices viable and publicly acceptable, we must address these interrelated problems holistically.

Education

At the root of the perception problem facing AR (or more widely, MR) devices, education provides an understanding of the true threat of such devices. A lack of either user or bystander education is the cause of the demise of past devices and a lack of understanding of the actual threats posed by modern ones. For instance, Google's Glass suffered from a misunderstanding of the device's capability to record and exploit data [55]. In contrast, modern devices that pose far more risk are overlooked [149]. From this, it stands to reason that a more thorough understanding of the capabilities of modern MR devices would ameliorate both extremes. Understanding that modern MR devices are capable of facial recognition, weight estimation, and voice recording (among other exploitation avenues) would create the wariness that we believe would be commensurate with the threat. This would also direct the public's focus to existing risks, direct future technical and policy solutions towards more refined language, and prevent overly restrictive limits.

The mechanism to convey this understanding, however, is not as precise. Existing methods have failed to impress a complete understanding of these threats on the public outside of research circles. There is much room for future research investigating how best to bridge

this understanding gap. Future work should explore what data should be conveyed, at which times, and on what mediums. When properly conveyed, this information could simultaneously reduce undue scrutiny on unlikely or infeasible threats while increasing scrutiny on viable ones.

Bystander Perceptions

Even when provided with the correct information about the viability of a threat, we must better understand the psychological impact of MR devices in public areas. Existing work on this topic focuses on protection techniques and consent mechanisms. Still, it does little to evaluate an implemented, real-time system that seeks to provide the proper (and only the proper) information to third-party applications. Currently published surveys on bystander perceptions and apprehensions shed light on this by informing us of bystander tolerances in certain contexts, including when friends, strangers, and coworkers use the device. A wider familiarization with MR devices may be necessary to fully realize any solution [51]. Other surveys explore new "body-based gestures" like those available as input mechanisms on some modern MR devices. These studies find that the new, MR-friendly methods are less acceptable than others to the general public [167]. Using implemented and tested solutions, such as BystandAR, future work can explore the impact of technical solutions on bystander perceptions of privacy.

Experience

Certain technologies have always created distrust at the onset of their proliferation in the public. Even the cellphone, now ubiquitous, was viewed warily by a public concerned with its ability to record personal data as recently as 2000 [154]. However, these technologies have

become more widely used and accepted over time. It also follows that as MR technology matures and potentially becomes more ubiquitous, the context of their acceptable use changes as well [144].

Nuance in Labeling Subjects and Bystanders

Existing solutions generally label subjects and bystanders according to a simple dichotomy. However, more detailed and nuanced definitions exist. Potential future solutions could sub-classify subjects as "co-users" and subjects, giving more protection to those whom the MR user intends to interact with, as opposed to "co-users" who are fully aware and can control the application and data collection [160]. Additionally, future solutions can protect persons who potentially have permitted data collection in the past but are not relevant to the current interaction context [160].

7.2.3 Shoulder Surfing

ShouldAR was tested assuming the system could observe the attacker's eyes and face with an attacker positioned no more than 2 meters to the victim's rear. More work is needed on longer-range solutions, especially when attackers have tried to obscure their gaze. Additionally, such solutions should also be able to distinguish attackers from legitimate collaborators.

Maybe more interestingly, we must acknowledge the bystander privacy concerns above. ShouldAR does not save any uniquely identifying information about the potential attackers to the user's rear. However, knowing that this system constantly evaluates eye gaze could concern those surrounding the device. For instance, other diners observing the user's plate would be considered shoulder surfing if the user were at a restaurant. Future systems must acknowledge the diversity of potential situations, especially when these devices are worn in

public places.

7.3 Conclusion

This dissertation has presented three new methods for improving security and privacy expectations in AR, specifically by using human eye gaze. BystandAR, ShouldAR, and GazePair represent advances in the state-of-the-art in their respective areas. These methods, while not without limitations, can provide unique protections and benefits to AR users. This work can also aid other research areas, such as privacy policymaking. As AR devices become more ubiquitous and powerful, solutions such as these are needed to ensure the viability and social acceptability of these devices and ensure a future where augmented reality is widely accepted and used. Without them, these devices may languish in obscurity created by misunderstanding and misuse. I sincerely hope this work prevents such an occurrence and aids in the widespread understanding and adoption of these unique and powerful devices.

Bibliography

[1] Ahmed Abdou, Ekta Sood, Philipp Müller, and Andreas Bulling. Gaze-enhanced crossmodal embeddings for emotion recognition. *Proc. ACM Hum.-Comput. Interact.*, 6(ETRA), may 2022. doi: 10.1145/3530879. URL https://doi.org/10.1145/3530879.

[2] Yasmeen Abdrabou, Mohamed Khamis, Rana Mohamed Eisa, Sherif Ismail, and Amrl Elmougy. Just gaze and wave: Exploring the use of gaze and gestures for shoulder-surfing resilient authentication. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367097. doi: 10.1145/3314111.3319837.

[3] Paarijaat Aditya, Rijurekha Sen, Peter Druschel, Seong Joon Oh, Rodrigo Benenson, Mario Fritz, Bernt Schiele, Bobby Bhattacharjee, and Tong Tong Wu. I-pic: A platform for privacy-compliant image capture. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '16, page 235–248, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342698. doi: 10.1145/2906388.2906412.

[4] Open AI. Dall · e: Creating images from text, 2024. URL https://openai.com/research/dall-e. https://openai.com/research/dall-e.

[5] Hironori Akechi, Atsushi Senju, Helen Uibo, Yukiko Kikuchi, Toshikazu Hasegawa, and Jari K Hietanen. Attention to eye contact in the west and east: autonomic responses and evaluative ratings. *PLoS One*, 8(3):e59312, 2013.

[6] Taslima Akter, Tousif Ahmed, Apu Kapadia, and Manohar Swaminathan. Shared

privacy concerns of the visually impaired and sighted bystanders with camera-based assistive technologies. *ACM Trans. Access. Comput.*, 15(2), 2022. ISSN 1936-7228. doi: 10.1145/3506857.

[7] Mohammed Eunus Ali, Anika Anwar, Ishrat Ahmed, Tanzima Hashem, Lars Kulik, and Egemen Tanin. Protecting mobile users from visual privacy attacks. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, page 1–4, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330473. doi: 10.1145/2638728.2638788.

[8] Mohammed Eunus Ali, Anika Anwar, Ishrat Ahmed, Tanzima Hashem, Lars Kulik, and Egemen Tanin. Protecting mobile users from visual privacy attacks. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, page 1–4, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330473. doi: 10.1145/2638728.2638788.

[9] Wi-Fi Alliance. Wi-fi direct, August 2022. URL https://www.wi-fi.org/discover-wi-fi/wi-fi-direct.

[10] Maraim Alnefaie and Srinivas Sampalli. Braillepassword: accessible web authentication technique on touchscreen devices. *Journal of Ambient Intelligence and Humanized Computing*, 10, 06 2019. doi: 10.1007/s12652-018-0860-x.

[11] Thomas Alsop. Augmented (ar), virtual reality (vr), and mixed reality (mr) market size 2021-2024, November 2021. URL https://www.statista.com/statistics/591181/global-augmented-virtual-reality-market-size/#statisticContainer.

[12] Thomas Alsop. Global mobile augmented reality (ar) user devices 2024, 2022. URL https://www.statista.com/statistics/1098630/global-mobile-augmented-reality-ar-users/.

[13] Suliman A. Alsuhibany. Vibration-based pattern password approach for visually impaired people. *Comput. Syst. Sci. Eng.*, 40(1):341–356, 2022. doi: 10.32604/csse.2022.018563. URL https://doi.org/10.32604/csse.2022.018563.

[14] Amazon. Treedix ov2640 camera module 140 degree wide angle cmos 2mp camera mini camera module, 2023. URL https://www.amazon.com/Treedix-OV2640-Camera-Module-Degree/dp/B0894KKXHX/ref=asc_df_B0894KKXHX/?tag=hyprod-20&linkCode=df0&hvadid=658532215676&hvpos=&hvnetw=g&hvrand=13097930281404645000&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9008695&hvtargid=pla-1489482268684&psc=1&mcid=6741e1d20ca03b7ba63f7457dfe91ada. https://www.amazon.com/Treedix-OV2640-Camera-Module-Degree/dp/B0894KKXHX/ref=asc_df_B0894KKXHX/?tag=hyprod-20.

[15] Apple. Apple vision pro - cameras and sensors, 2023. URL https://www.apple.com/apple-vision-pro/?afid=p238%7CsTHXpRf18-dc_mtid_%5Btracker_id%5D_pcrid_664829837701_pgrid_150826790596_pexid__&cid=wwa-us-kwgo-VisionPro-slid---productid--Brand-Avalanche-announceBrand-#product-tour-gallery-item-1.

[16] Abdullah Al Arafat, Zhishan Guo, and Amro Awad. Vr-spy: A side-channel attack on virtual key-logging in vr headsets. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 564–572, 2021. doi: 10.1109/VR50410.2021.00081.

[17] M. Argyle. *The Psychology of Interpersonal Behaviour*. Penguin Books Limited, 1994. ISBN 9780141927121. URL https://books.google.com/books?id=VQOzdOxJFZAC.

[18] Somnath Arjun, Archana Hebbar, Sanjana, and Pradipta Biswas. Vr cognitive load dashboard for flight simulator. In *2022 Symposium on Eye Tracking Research and Applications*, ETRA '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392525. doi: 10.1145/3517031.3529777. URL https://doi-org.ezproxy.lib.vt.edu/10.1145/3517031.3529777.

[19] Deborah Bach. U.s. army to use hololens technology in high-tech headsets for soldiers, June 2021. URL https://news.microsoft.com/transform/u-s-army-to-use-hololens-technology-in-high-tech-headsets-for-soldiers/.

[20] M. Bâce, A. Saad, S. Khamis, M. andSchneegass, and A. Bulling. Privacyscout: Assessing vulnerability to shoulder surfing on mobile devices. In *Privacy Enhancing Technologies Symposium 2022 (PETS 2022)*, 2022. doi: 10.56553/popets-2022-0090.

[21] Divya Bhati. Atm card scams: What is shoulder surfing and how to protect your atm pin from scammers, 2023. URL https://www.indiatoday.in/technology/news/story/atm-card-scams-what-is-shoulder-surfing-and-how-to-protect-your-atm-pin-from-scamm https://www.indiatoday.in/technology/news/story/atm-card-scams-what-is-shoulder-surfing-and-how-to-protect-your-atm-pin-from-scamm

[22] Andrea Bianchi, Ian Oakley, Vassilis Kostakos, and Dong Soo Kwon. The phone lock: audio and haptic shoulder-surfing resistant pin entry methods for mobile devices. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, pages 197–200, 2010.

[23] Robert Biddle, Sonia Chiasson, and P.C. Van Oorschot. Graphical passwords: Learning from the first twelve years. *ACM Comput. Surv.*, 44(4), sep 2012. ISSN 0360-0300. doi: 10.1145/2333112.2333114. URL https://doi.org/10.1145/2333112.2333114.

[24] Basak Bilgi and Bulent Tugrul. A shoulder-surfing resistant graphical authentication method. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, pages 1–4, 2018. doi: 10.1109/IDAP.2018.8620934.

[25] Nicola Binetti, Charlotte Harrison, Antoine Coutrot, and Isabelle Johnston, Alan ad Mareschal. Pupil dilation as an index of preferred mutual gaze duration. *R. Soc. open sci.*, 3(7), 2016. ISSN 2054-5703. doi: 10.1098/rsos.160086.

[26] Mylene Bolmont, John T. Cacioppo, and Stephanie Cacioppo. Love is in the gaze: An eye-tracking study of love and sexual desire. *Psychological Science*, 25(9):1748–1756, 2014. doi: 10.1177/0956797614539706. URL https://doi.org/10.1177/0956797614539706.

[27] Ricard Borràs, Àgata Lapedriza, and Laura Igual. Depth information in human gait analysis: An experimental study on gender recognition. In Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, pages 98–105, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-31298-4.

[28] Andrew Lim Chee Yeung, Bryan Lee Weng Wai, Cheng Hao Fung, Fiza Mughal, and Vahab Iranmanesh. Graphical password: Shoulder-surfing resistant using falsification. In *2015 9th Malaysian Software Engineering Conference (MySEC)*, pages 145–148, 2015. doi: 10.1109/MySEC.2015.7475211.

[29] Chun-Yu (Daniel) Chen, Bo-Yao Lin, Junding Wang, and Kang G. Shin. Keep others from peeking at your mobile device screen! In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, New York,

NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361699. doi: 10.1145/3300061.3300119.

[30] Shudi Chen and Youwen Zhu. A textual password entry method resistant to human shoulder-surfing attack. In Jaideep Vaidya, Xiao Zhang, and Jin Li, editors, *Cyberspace Safety and Security*, pages 409–420, Cham, 2019. Springer International Publishing. ISBN 978-3-030-37352-8.

[31] Zijun Cheng, Tianwei Shi, Wenhua Cui, Yunqi Dong, and Xuehan Fang. 3d face recognition based on kinect depth data. In *2017 4th International Conference on Systems and Informatics (ICSAI)*, pages 555–559, 445 Hoes Lane, Piscataway, NJ 08854., 2017. IEEE. doi: 10.1109/ICSAI.2017.8248353.

[32] Edward Chou, Matthew Tan, Cherry Zou, Michelle Guo, Albert Haque, Arnold Milstein, and Li Fei-Fei. Privacy-preserving action recognition for smart hospitals using low-resolution depth images. *CoRR*, abs/1811.09950, 2018.

[33] James Clayton. Facial recognition beats the covid-mask challenge, Nov 2022. URL https://www.bbc.com/news/technology-56517033.

[34] Matthew Corbett. Gazepair github repository, 2022. URL https://github.com/matthewc84/GazePair.

[35] Apple Corporation. How to use airdrop on your iphone, ipad, or ipod touch, May 2022. URL https://support.apple.com/en-us/HT204144.

[36] Ioane Corporation. Talking passwords and entropy, May 2022. URL https://iocane.com.au/talking-passwords-and-entropy/.

[37] Meta Corporation. Welcome to meta, August 2022. URL https://about.facebook.com/technologies/.

[38] Asadullah Dal. Eyes-position-estimator-mediapipe, Feb 2023. URL https://github.com/Asadullah-Dal17/Eyes-Position-Estimator-Mediapipe. https://github.com/Asadullah-Dal17/Eyes-Position-Estimator-Mediapipe.

[39] Lyle Daly. What is proof of work (pow) in crypto?, December 2021. URL https://www.fool.com/investing/stock-market/market-sectors/financials/cryptocurrency-stocks/proof-of-work/.

[40] David Darling. Automated privacy protection for mobile device users and bystanders in public spaces. Master's thesis, University of Arkansas, 2021. Retrieved from https://scholarworks.uark.edu/etd/4218.

[41] David Darling, Ang Li, and Qinghua Li. Identification of subjects and bystanders in photos with feature-based machine learning. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6, 445 Hoes Lane, Piscataway, NJ 08854., 2019. IEEE.

[42] David Darling, Ang Li, and Qinghua Li. Automated bystander detection and anonymization in mobile photography. In *International Conference on Security and Privacy in Communication Systems*, pages 402–424. Springer, 2020.

[43] Brendan David-John, Diane Hosfelt, Kevin Butler, and Eakta Jain. A privacy-preserving approach to streaming eye-tracking data. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2555–2565, 2021. doi: 10.1109/TVCG.2021.3067787.

[44] Brendan David-John, Diane Hosfelt, Kevin Butler, and Eakta Jain. Let's soup up xr: Collected thoughts from an ieee vr workshop on privacy in mixed reality. In *VR4Sec: Security for VR and VR for Security, SOUPS 2021 Workshop*, 2021.

[45] Jaybie A. De Guzman, Kanchana Thilakarathna, and Aruna Seneviratne. Security and privacy approaches in mixed reality: A literature survey. *ACM Comput. Surv.*, 52 (6), 2019. ISSN 0360-0300. doi: 10.1145/3359626.

[46] Alexander De Luca, Roman Weiss, and Heiko Drewes. Evaluation of eye-gaze interaction methods for security enhanced pin-entry. In *Proceedings of the 19th Australasian Conference on Computer-Human Interaction: Entertaining User Interfaces*, OZCHI '07, page 199–202, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595938725. doi: 10.1145/1324892.1324932.

[47] Alexander De Luca, Martin Denzel, and Heinrich Hussmann. Look into my eyes! can you guess my password? In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605587363. doi: 10.1145/1572532.1572542.

[48] Alexander De Luca, Katja Hertzschuch, and Heinrich Hussmann. Colorpin: Securing pin entry through indirect input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1103–1106, 2010.

[49] Ziedune Degutyte and Arlene Astell. The role of eye gaze in regulating turn taking in conversations: A systematized review of methods and findings. *Frontiers in Psychology*, 12, 2021. ISSN 1664-1078. doi: 10.3389/fpsyg.2021.616471. URL https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2021.616471.

[50] Ziedune Degutyte and Arlene Astell. The role of eye gaze in regulating turn taking in conversations: A systematized review of methods and findings. *Frontiers in Psychology*, 12, 2021. ISSN 1664-1078. doi: 10.3389/fpsyg.2021.616471.

[51] Tamara Denning, Zakariya Dehlawi, and Tadayoshi Kohno. In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating tech-

nologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 2377–2386, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450324731. doi: 10.1145/2556288.2557352.

[52] Yanko Design. Every single sensor inside the apple vision pro and what it's individually designed to do, 2023. URL https://www.yankodesign.com/2023/06/07/every-single-sensor-in-the-apple-vision-pro-and-what-its-individually-designed-to-

[53] Mariella Dimiccoli, Juan Marín, and Edison Thomaz. Mitigating bystander privacy concerns in egocentric activity recognition with deep learning and intentional image degradation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(4), 2018. doi: 10.1145/3161190.

[54] Luke Dormehl. Forget ar glasses. augmented reality is headed to your windshield, December 2020. URL https://www.digitaltrends.com/features/envisics-ar-windshield-technology/.

[55] Sarah Downey. Google glass cons: How the camera-embedded eyeglasses could shatter privacy, 2013. URL https://variety.com/2013/biz/news/google-glass-cons-how-the-camera-embedded-eyeglasses-could-shatter-privacy-1200563 https://variety.com/2013/biz/news/google-glass-cons-how-the-camera-embedded-eyegla

[56] Drewbatgit. App capability declarations - uwp applications, 2021. URL https://learn.microsoft.com/en-us/windows/uwp/packaging/app-capability-declarations.

[57] Reyhan Düzgün, Peter Mayer, and Melanie Volkamer. Shoulder-surfing resistant authentication for augmented reality. In *Nordic Human-Computer Interaction Conference*, NordiCHI '22, New York, NY, USA, 2022. Association for Computing Machin-

ery. ISBN 9781450396998. doi: 10.1145/3546155.3546663. URL https://doi.org/10.1145/3546155.3546663.

[58] Malin Eiband, Mohamed Khamis, Emanuel von Zezschwitz, Heinrich Hussmann, and Florian Alt. Understanding shoulder surfing in the wild: Stories from users and observers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 4254–4265, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346559. doi: 10.1145/3025453.3025636.

[59] Passant Elagrousy, Mohammed Khamis, Florian Mathis, Diana Irmscher, Ekta Sood, Andreas Bulling, and Albrecht Schmidt. Impact of privacy protection methods of lifelogs on remembered memories. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394215. doi: 10.1145/3544548.3581565.

[60] Rob Enderle. Apple's vision pro: The pros and cons of head-mounted hardware, 2024. URL https://www.computerworld.com/article/3699672/apples-vision-pro-the-pros-and-cons-of-head-mounted-hardware.html#:~:text=Real%2Dworld%20benefits%3A%20privacy%20and,and%20potentially%20greater%20battery%20life. https://www.computerworld.com/article/3699672/apples-vision-pro-the-pros-and-cons-of-head-mounted-hardware.html#:~:text=Real%2Dworld%20benefits%3A%20privacy%20and,and%20potentially%20greater%20battery%20life.

[61] Espressif. Espressif - esp32, 2023. URL https://www.espressif.com/en/products/socs/esp32. https://www.espressif.com/en/products/socs/esp32.

[62] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. Android permissions: User attention, comprehension, and behavior. In

Proceedings of the Eighth Symposium on Usable Privacy and Security, SOUPS '12, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450315326. doi: 10.1145/2335356.2335360.

[63] Ethan Gaebel, Ning Zhang, Wenjing Lou, and Y. Thomas Hou. Looks good to me: Authentication for augmented reality. In *Proceedings of the 6th International Workshop on Trustworthy Embedded Devices*, TrustED '16, page 57–67, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450345675. doi: 10.1145/2995289.2995295.

[64] Ceenu George, Mohamed Khamis, Daniel Buschek, and Heinrich Hussmann. Investigating the third dimension for authentication in immersive virtual reality and in the real world. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 277–285, 2019. doi: 10.1109/VR.2019.8797862.

[65] Dylan Gilbert and Michael Fagan. Journey into the immersive frontier: Preliminary nist research on cybersecurity and privacy standards for immersive technologies, 2024. URL https://www.nist.gov/blogs/cybersecurity-insights/journey-immersive-frontier-preliminary-nist-research-cybersecurity-and. https://www.nist.gov/blogs/cybersecurity-insights/journey-immersive-frontier-preliminary-nist-research-cybersecurity-and.

[66] Sarwesh Giri, Gurchetan Singh, Babul Kumar, Mehakpreet Singh, Deepanker Vashisht, Sonu Sharma, and Prince Jain. Emotion detection with facial feature recognition using cnn and opencv. In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pages 230–232, 445 Hoes Lane, Piscataway, NJ 08854., 2022. IEEE. doi: 10.1109/ICACITE53722.2022.9823786.

[67] Google. Image of the google glass, 2023. URL https://g.co/kgs/Hu7WJE.

[68] Google. Why on-device machine learning?, 2023. URL https://developers.google.com/learn/topics/on-device-ml/learn-more.

[69] Google. Mediapipe, 2023. URL https://developers.google.com/mediapipe. https://developers.google.com/mediapipe.

[70] Google. Google scholar search, 2024. URL https://scholar.google.com/scholar?hl=en&as_sdt=0%2C47&as_vis=1&q=%22shoulder+surfing%22&btnG=. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C47&as_vis=1&q=%22shoulder+surfing%22&btnG=.

[71] Yongjie Guan, Xueyu Hou, Nan Wu, Bo Han, and Tao Han. Realtime 3d object detection for headsets. *CoRR*, abs/2201.08812, 2022.

[72] Matthew Allan Hamilton, Anthony Paul Beug, Howard John Hamilton, and Wil James Norton. Augmented reality technology for people living with dementia and their care partners. In *2021 the 5th International Conference on Virtual and Augmented Reality Simulations*, ICVARS 2021, page 21–30, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450389327. doi: 10.1145/3463914.3463918.

[73] Katie Terrell Hanna. What is shoulder surfing?, 2024. URL https://www.techtarget.com/searchsecurity/definition/shoulder-surfing. https://www.techtarget.com/searchsecurity/definition/shoulder-surfing.

[74] Rakibul Hasan, David Crandall, Mario Fritz, and Apu Kapadia. Automatically detecting bystanders in photos to reduce privacy risks. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 318–335, 445 Hoes Lane, Piscataway, NJ 08854., 2020. IEEE. doi: 10.1109/SP40000.2020.00097.

[75] Oliver Haslam. Apple vision pro sales are already 4x the number of iconic next cubes steve jobs ever sold — and it's just getting started, 2024. URL https://www.imore.com/vision-pro/apple-vision-pro-sales-are-already-4x-the-number-of-iconic-next-cubes-steve-jobs-e https://www.imore.com/vision-pro/apple-vision-pro-sales-are-already-4x-the-number-

[76] Scott Hayden. Zuckerberg: Oculus quest pro could bring face & eye tracking, September 2021. URL https://www.roadtovr.com/mark-zuckerberg-oculus-quest-pro/.

[77] Bogdan Hoanca and Kenrick Mock. Secure graphical password system for high traffic public areas. In *Proceedings of the 2006 Symposium on Eye Tracking Research and Applications*, ETRA '06, page 35, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933050. doi: 10.1145/1117309.1117319.

[78] I.P. Howard and B.J. Rogers. *Binocular Vision and Stereopsis*. Oxford psychology series. Oxford University Press, 1995. ISBN 9780195084764. URL https://books.google.com/books?id=I8vqITdETe0C.

[79] Jinhan Hu, Andrei Iosifescu, and Robert LiKamWa. Lenscap: Split-process framework for fine-grained visual privacy control for augmented reality apps. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '21, page 14–27, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384438. doi: 10.1145/3458864.3467676.

[80] Ahmad Jalal, Shaharyar Kamal, and Daijin Kim. Human depth sensors-based activity recognition using spatiotemporal features and hidden markov model for smart environments. *Journal of Computer Networks and Communications*, 2016:8087545, 2016. ISSN 2090-7141. doi: 10.1155/2016/8087545.

[81] Suman Jana, Arvind Narayanan, and Vitaly Shmatikov. A scanner darkly: Protecting user privacy from perceptual applications. In *2013 IEEE Symposium on Security and Privacy*, pages 349–363, 445 Hoes Lane, Piscataway, NJ 08854., 2013. IEEE. doi: 10.1109/SP.2013.31.

[82] Fucheng Jia, Deyu Zhang, Ting Cao, Shiqi Jiang, Yunxin Liu, Ju Ren, and Yaoxue Zhang. Codl: Efficient cpu-gpu co-execution for deep learning inference on mobile devices. In *The 20th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'22)*. Microsoft, ACM, 2022. URL https://www.microsoft.com/en-us/research/publication/codl-efficient-cpu-gpu-co-execution-for-deep-learning-inference-on-mobile-devices/

[83] Wenqiang Jin, Ming Li, Srinivasan Murali, and Linke Guo. Harnessing the ambient radio frequency noise for wearable device pairing. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 1135–1148, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370899. doi: 10.1145/3372297.3417288.

[84] Ronald Kainda, Ivan Flechais, and A. W. Roscoe. Usability and security of out-of-band channels in secure device pairing protocols. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605587363. doi: 10.1145/1572532.1572547.

[85] Sebastian Kapp, Michael Barz, Sergey Mukhametov, Daniel Sonntag, and Jochen Kuhn. Arett: Augmented reality eye tracking toolkit for head mounted displays. *Sensors*, 21(6), 2021. ISSN 1424-8220. doi: 10.3390/s21062234.

[86] Christina Katsini, Yasmeen Abdrabou, George E. Raptis, Mohamed Khamis, and Florian Alt. The role of eye gaze in security and privacy applications: Survey and future

hci research directions. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–21, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376840.

[87] Mohamed Khamis, Florian Alt, Mariam Hassib, Emanuel von Zezschwitz, Regina Hasholzner, and Andreas Bulling. Gazetouchpass: Multimodal authentication using gaze and touch on mobile devices. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, page 2156–2164, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340823. doi: 10.1145/2851581.2892314.

[88] Mohamed Khamis, Mariam Hassib, Emanuel von Zezschwitz, Andreas Bulling, and Florian Alt. Gazetouchpin: Protecting sensitive data on mobile devices using secure multimodal authentication. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI '17, page 446–450, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450355438. doi: 10.1145/3136755.3136809.

[89] Marion Koelle, Matthias Kranz, and Andreas Möller. Don't look at me that way! understanding user attitudes towards data glasses usage. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '15, page 362–372, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336529. doi: 10.1145/2785830.2785842.

[90] Marion Koelle, Swamy Ananthanarayan, Simon Czupalla, Wilko Heuten, and Susanne Boll. Your smart glasses' camera bothers me! exploring opt-in and opt-out gestures for privacy mediation. In *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*, NordiCHI '18, page 473–481, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450364379. doi: 10.1145/3240167.3240174.

[91] Marcin Kopaczka, Justus Schock, Jan Nestler, Kevin Kielholz, and Dorit Merhof. A combined modular system for face detection, head pose estimation, face tracking and emotion recognition in thermal infrared images. In *2018 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6. IEEE, 2018.

[92] Katharina Krombholz, Heidelinde Hobel, Markus Huber, and Edgar Weippl. Advanced social engineering attacks. *Journal of Information Security and Applications*, 22:113–122, 2015. ISSN 2214-2126. doi: https://doi.org/10.1016/j.jisa.2014.09.005. Special Issue on Security of Information and Networks.

[93] Romina Kühn, Mandy Korzetz, and Thomas Schlegel. User strategies for mobile device-based interactions to prevent shoulder surfing. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia*, MUM '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450376242. doi: 10.1145/3365610.3368412. URL https://doi-org.ezproxy.lib.vt.edu/10.1145/3365610.3368412.

[94] Arun Kumar, Nitesh Saxena, and Ersin Uzun. Alice meets bob: A comparative usability study of wireless device pairing methods for a "two-user" setting. *ArXiv*, abs/0907.4743, 2009.

[95] Manu Kumar, Tal Garfinkel, Dan Boneh, and Terry Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd Symposium on Usable Privacy and Security*, SOUPS '07, page 13–19, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595938015. doi: 10.1145/1280680.1280683.

[96] Manu Kumar, Tal Garfinkel, Dan Boneh, and Terry Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd Symposium on*

Usable Privacy and Security, SOUPS '07, page 13–19, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595938015. doi: 10.1145/1280680.1280683.

[97] Pavan Kunchala. Real-time age gender detection using opencv, 2021. URL https://medium.com/analytics-vidhya/real-time-age-gender-detection-using-opencv-fa705fe0e1fa.

[98] Taekyoung Kwon and Jin Hong. Analysis and improvement of a pin-entry method resilient to shoulder-surfing and recording attacks. *Ieee transactions on information forensics and security*, 10(2):278–292, 2014.

[99] Magic Leap. Magic leap 2, 2023. URL https://www.magicleap.com/magic-leap-2.

[100] Michelle Ye Hee Lee. Seoul wants to build a metaverse. a virtual new year's eve ceremony will kick it off, November 2021. URL https://www.washingtonpost.com/world/asia_pacific/metaverse-seoul-virtual/2021/11/27/03928120-4248-11ec-9404-50a28a88b9cd_story.html.

[101] The Florida Legislature. The 2023 florida statutes (including special session c). *OJ*, Title XXXVII, 2023. URL http://www.leg.state.fl.us/Statutes/index.cfm?App_mode=Display_Statute&Search_String=&URL=0600-0699/0655/Sections/0655.962.html. http://www.leg.state.fl.us/Statutes/index.cfm?App_mode=Display_Statute&Search_String=&URL=0600-0699/0655/Sections/0655.962.html.

[102] Sarah M. Lehman, Abrar S. Alrumayh, Kunal Kolhe, Haibin Ling, and Chiu C. Tan. Hidden in plain sight: Exploring privacy risks of mobile augmented reality applications. *ACM Trans. Priv. Secur.*, 25(4), jul 2022. ISSN 2471-2566. doi: 10.1145/3524020. URL https://doi.org/10.1145/3524020.

[103] Ang Li, Qinghua Li, and Wei Gao. Privacycamera: Cooperative privacy-aware photographing with mobile phones. In *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9, 445 Hoes Lane, Piscataway, NJ 08854., 2016. IEEE. doi: 10.1109/SAHCN.2016.7733008.

[104] Ming Li, Shucheng Yu, Joshua D. Guttman, Wenjing Lou, and Kui Ren. Secure ad hoc trust initialization and key management in wireless body area networks. *ACM Trans. Sen. Netw.*, 9(2), apr 2013. ISSN 1550-4859. doi: 10.1145/2422966.2422975.

[105] Jonathan Liebers and Stefan Schneegass. Gaze-based authentication in virtual reality. In *ACM Symposium on Eye Tracking Research and Applications*, ETRA '20 Adjunct, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371353. doi: 10.1145/3379157.3391421.

[106] Jonathan Liebers, Patrick Horn, Christian Burschik, Uwe Gruenefeld, and Stefan Schneegass. Using gaze behavior and head orientation for implicit identification in virtual reality. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, VRST '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450390927. doi: 10.1145/3489849.3489880.

[107] Linzaer. Ultra-light-fast-generic-face-detector-1mb, 2022. URL https://github.com/Linzaer/Ultra-Light-Fast-Generic-Face-Detector-1MB.

[108] Luyang Liu, Hongyu Li, and Marco Gruteser. Edge assisted real-time object detection for mobile augmented reality. In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361699. doi: 10.1145/3300061.3300116.

[109] Dillon Lohr, Samuel-Hunter Berndt, and Oleg Komogortsev. An implementation of eye movement-driven biometrics in virtual reality. In *Proceedings of the 2018 ACM*

Symposium on Eye Tracking Research and Applications, ETRA '18, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450357067. doi: 10.1145/3204493.3208333.

[110] O. Lorenz and Ulrike Thomas. Real time eye gaze tracking system using cnn-based facial features for human attention measurement. In *VISIGRAPP*, 2019. URL https://api.semanticscholar.org/CorpusID:88489425.

[111] Feiyu Lu, Leonardo Pavanatto, and Doug A. Bowman. In-the-wild experiences with an interactive glanceable ar system for everyday use. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction*, SUI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702815. doi: 10.1145/3607822.3614515. URL https://doi.org/10.1145/3607822.3614515.

[112] Ye Luo, Zehai Su, Wei Zheng, Zhaobin Chen, Fuqin Wang, Zhemin Zhang, and Jinjun Chen. A novel memory-hard password hashing scheme for blockchain-based cyber-physical systems. *ACM Trans. Internet Technol.*, 21(2), mar 2021. ISSN 1533-5399. doi: 10.1145/3408310.

[113] MagicLeap. Eye tracking - unity, 2022. URL https://ml1-developer.magicleap.com/en-us/learn/guides/eye-tracking-tutorial-unity.

[114] Bhanuka Mahanama, Yasith Jayawardana, Sundararaman Rengarajan, Gavindya Jayawardena, Leanne Chukoskie, Joseph Snider, and Sampath Jayarathna. Eye movement and pupil measures: A review. *Frontiers in Computer Science*, 3, 2022. ISSN 2624-9898. doi: 10.3389/fcomp.2021.733531. URL https://www.frontiersin.org/articles/10.3389/fcomp.2021.733531.

[115] Florian Mathis, John H. Williamson, Kami Vaniea, and Mohamed Khamis. Fast and secure authentication in virtual reality using coordinated 3d manipulation and

pointing. *ACM Trans. Comput.-Hum. Interact.*, 28(1), 2021. ISSN 1073-0516. doi: 10.1145/3428121.

[116] Suhas Mathur, Robert Miller, Alexander Varshavsky, Wade Trappe, and Narayan Mandayam. Proximate: Proximity-based secure pairing using ambient wireless signals. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, page 211–224, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450306430. doi: 10.1145/1999995.2000016.

[117] Saul McLeod. Likert scale definition, examples and analysis, Nov 2019. URL https://www.simplypsychology.org/likert-scale.html.

[118] Tushar Mehta. How ar glasses are going from niche gadget to smartphone replacement, 2023. URL https://www.digitaltrends.com/mobile/ar-glasses-replace-smartphones-future-how/. https://www.digitaltrends.com/mobile/ar-glasses-replace-smartphones-future-how/.

[119] Meta. Introducing meta quest pro, an advanced vr device for collaboration and creation, 2022. URL https://www.oculus.com/blog/meta-quest-pro-price-release-date/.

[120] Meta. Meta quest pro, 2023. URL https://www.meta.com/quest/quest-pro/.

[121] Meta. Expand your world with meta quest 3, 2024. URL https://www.meta.com/quest/quest-3. https://www.meta.com/quest/quest-3.

[122] Microsoft. Azure spatial anchors overview, 2021. URL https://docs.microsoft.com/en-us/azure/spatial-anchors/overview.

[123] Microsoft. Eye tracking on hololens 2, 2021. URL https://docs.microsoft.com/en-us/windows/mixed-reality/design/eye-tracking.

[124] Microsoft. Hololens 2 - overview, features, and specs, December 2021. URL https://www.microsoft.com/en-us/hololens/hardware.

[125] Microsoft. Hololens 2, 2021. URL https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/hologram-stability.

[126] Microsoft. Getting started with eye tracking in mrtk, May 2021. URL https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/input/eye-tracking/eye-tracking-basic-setup?view=mrtkunity-2021-05.

[127] Microsoft. Randomnumbergenerator class, December 2021. URL https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.randomnumbergenerator?view=net-6.0.

[128] Microsoft. Hololens 2 gestures for navigating a guide in dynamics 365 guides, February 2022. URL https://docs.microsoft.com/en-us/dynamics365/mixed-reality/guides/operator-gestures-hl2#gestures.

[129] Microsoft. Facedetector class, oct 2022. URL https://learn.microsoft.com/en-us/uwp/api/windows.media.faceanalysis.facedetector?view=winrt-22621.

[130] Microsoft. Eye tracking on hololens 2, 2022. URL https://learn.microsoft.com/en-us/windows/mixed-reality/design/eye-tracking.

[131] Microsoft. Point and commit with hands, February 2022. URL https://docs.microsoft.com/en-us/windows/mixed-reality/design/point-and-commit.

[132] Microsoft. Hologram stability, 2022. URL https://www.microsoft.com/en-us/hololens/hardware.

[133] Microsoft. Holographic face tracking sample, oct 2022. URL https://learn.microsoft.com/en-us/samples/microsoft/windows-universal-samples/holographicfacetracking/.

[134] Microsoft. Mr in your classroom, February 2022. URL https://www.microsoft.com/en-us/education/mixed-reality.

[135] Microsoft. What is mixed reality toolkit 2?, oct 2022. URL https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05.

[136] Microsoft. Mediacapture class, oct 2022. URL https://learn.microsoft.com/en-us/uwp/api/windows.media.capture.mediacapture?view=winrt-22621.

[137] Microsoft. Mixed reality capture overview, Nov 2022. URL https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/mixed-reality-capture-overview.

[138] Microsoft. World locking and spatial anchors in unity, 2022. URL https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/spatial-anchors-in-unity?tabs=wlt.

[139] Microsoft. Create images from words with ai, Feb 2024. URL https://www.bing.com/images/create/a-person-working-at-a-desk-on-a-laptop-with-a-pers/1-65cb93bd21814fdd927fa22c3e8d0aa7?id=EowoJwMdjsIQj%2fjndAM3uw%3d%3d&view=detailv2&idpp=genimg&idpbck=1&form=BICREC&idpview=singleimage&thid=OIG4.VXWiwzf4YiCerq6DxWjb. https://www.bing.com/images/create/a-person-working-at-a-desk-on-a-laptop-with-a-pers/1-65cb93bd21814fdd927fa22c3e8d0aa7?id=EowoJwMdjsIQj%2fjndAM3uw%3d%3d&

view=detailv2&idpp=genimg&idpbck=1&form=BICREC&idpview=singleimage&thid=OIG4.VXWiwzf4YiCerq6DxWjb.

[140] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. In *IEICE TRANSACTIONS on Information and Systems*, volume E77-D, pages 1321–1329. IEICE, IEICE, December 1994.

[141] Kathleen Moriarty, Burt Kaliski, and Andreas Rusch. PKCS #5: Password-Based Cryptography Specification Version 2.1. RFC 8018, January 2017. URL https://rfc-editor.org/rfc/rfc8018.txt.

[142] NanoNets. Introduction to motion estimation with optical flow, 2022. URL https://nanonets.com/blog/optical-flow/.

[143] Shivsevak Negi and Ritayan Mitra. Fixation duration and the learning process: an eye tracking study with subtitled videos. *Journal of Eye Movement Research*, 13, 08 2020. doi: 10.16910/jemr.13.6.1.

[144] Helen Nissembaum. Privacy as contextual integrity, 2004. URL https://digitalcommons.law.uw.edu/wlr/vol79/iss1/10/. https://digitalcommons.law.uw.edu/wlr/vol79/iss1/10/.

[145] NMSBA. What eye-tracking can and can't tell you about attention, 2023. URL https://nmsba.com/neuromarketing-companies/neuromarketing-technologies-explained/what-eye-tracking-can-and-cant-tell-you-about-attention.

[146] Syeda Noor. Why do atms have convex mirror?, 2024. URL https://www.boldsky.com/insync/pulse/2016/why-do-atms-have-a-convex-mirror/articlecontent-pf137288-107799.html. https://www.boldsky.

com/insync/pulse/2016/why-do-atms-have-a-convex-mirror/articlecontent-pf137288-107799.html.

[147] J. O'Dwyer, N. Murray, and R. Flynn. Eye-based continuous affect prediction. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 137–143, Los Alamitos, CA, USA, sep 2019. IEEE Computer Society. doi: 10.1109/ACII.2019.8925470. URL https://doi.ieeecomputersociety.org/10.1109/ACII.2019.8925470.

[148] Last 365 Days of SSA Research. Google scholar search, Feb 2019. URL https://scholar.google.com/scholar?start=0&q=%22shoulder+surfing%22&hl=en&scisbd=1&as_sdt=0,47&as_vis=1. https://scholar.google.com/scholar?start=0&q=%22shoulder+surfing%22&hl=en&scisbd=1&as_sdt=0,47&as_vis=1.

[149] Joseph O'Hagan, Pejman Saeghe, Jan Gugenheimer, Daniel Medeiros, Karola Marky, Mohamed Khamis, and Mark McGill. Privacy-enhancing technology and everyday augmented reality: Understanding bystanders' varying needs for awareness and consent. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 6(4), jan 2023. doi: 10.1145/3569501. URL https://doi.org/10.1145/3569501.

[150] Tomomi Okano and Minoru Nakayama. Feasibility of evaluating temporal changes in cognitive load factors using ocular features. In *ACM Symposium on Eye Tracking Research and Applications*, ETRA '21 Short Papers, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383455. doi: 10.1145/3448018.3458019. URL https://doi-org.ezproxy.lib.vt.edu/10.1145/3448018.3458019.

[151] OpenCV.org. Mediapipe, 2023. URL https://opencv.org/. https://opencv.org/.

[152] OpenVINO. gaze-estimation-adas-0002, Feb 2023. URL https://docs.openvino.

ai/2022.3/omz_models_model_gaze_estimation_adas_0002.html. https://docs.openvino.ai/2022.3/omz_models_model_gaze_estimation_adas_0002.html.

[153] JAMES PAINE. 10 real use cases for augmented reality, February 2022. URL https://www.inc.com/james-paine/10-real-use-cases-for-augmented-reality.html.

[154] Leysia Palen, Marilyn Salzman, and Ed Youngs. Going wireless: Behavior & practice of new mobile phone users. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, CSCW '00, page 201–210, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132220. doi: 10.1145/358916.358991.

[155] Athanasios Papadopoulos, Toan Nguyen, Emre Durmus, and Nasir Memon. Illusionpin: Shoulder-surfing resistant authentication using hybrid images. *IEEE Transactions on Information Forensics and Security*, 12(12):2875–2889, 2017. doi: 10.1109/TIFS.2017.2725199.

[156] Alfredo J Perez, Sherali Zeadally, and Scott Griffith. Bystanders' privacy. *IT Professional*, 19(3):61–65, 2017.

[157] Alfredo J. Perez, Sherali Zeadally, Scott Griffith, Luis Y. Matos Garcia, and Jaouad A. Mouloud. A user study of a wearable system to enhance bystanders' facial privacy. *IoT*, 1(2):198–217, 2020. ISSN 2624-831X. doi: 10.3390/iot1020013.

[158] Veronika Petrenko. Ar/mr/vr for healthcare: Benefits for doctors, nurses, and patients, August 2020. URL https://wear-studio.com/ar-mr-vr-for-healthcare-benefits-for-doctors-nurses-patients/.

[159] Photon. The ease-of-use of unity's networking with the performance & reliability of photon realtime, Nov 2022. URL https://www.photonengine.com/pun.

[160] James Pierce, Claire Weizenegger, Parag Nandi, Isha Agarwal, Gwenna Gram, Jade Hurrle, Hannah Liao, Betty Lo, Aaron Park, Aivy Phan, Mark Shumskiy, and Grace Sturlaugson. Addressing adjacent actor privacy: Designing for bystanders, co-users, and surveilled subjects of smart home cameras. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference*, DIS '22, page 26–40, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393584. doi: 10.1145/3532106.3535195. URL https://doi.org/10.1145/3532106.3535195.

[161] PonemonInstitute. Global visual hacking experimental study: Analysis, 2016. URL multimedia.3m.com/mws/media/12542320/global-visual-hacking-experiment-study-summary.pdf. https://multimedia.3m.com/mws/media/12542320/global-visual-hacking-experiment-study-summary.pdf.

[162] Pasika Ranaweera, Anca Delia Jurcut, and Madhusanka Liyanage. Survey on multi-access edge computing security and privacy. *IEEE Communications Surveys & Tutorials*, 23(2):1078–1124, 2021.

[163] Pei-Luen Patrick Rau, Jian Zheng, and Zhi Guo. Immersive reading in virtual and augmented reality environment. *Information and Learning Sciences*, 122(7/8):464–479, 2021. ISSN 2398-5348. doi: 10.1108/ILS-11-2020-0236.

[164] Nisarg Raval, Animesh Srivastava, Kiron Lebeck, Landon Cox, and Ashwin Machanavajjhala. Markit: Privacy markers for protecting visual secrets. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, page 1289–1295, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330473. doi: 10.1145/2638728.2641707.

[165] Research and Markets. Augmented reality market with covid-19 impact analysis, by device type (head-mounted display, head-up display), offering (hardware, software), application (consumer, commercial, healthcare), technology, and geography - global forecast to 2026, December 2021. URL https://www.researchandmarkets.com/reports/5416605/.

[166] Reydar. The future of augmented reality, 2023. URL https://www.reydar.com/exploring-the-future-potential-of-augmented-reality-trends-technology-and-impact/#:~:text=In%20the%20near%20future%2C%20we,and%20technology%20than%20today's%20smartphones. https://www.reydar.com/exploring-the-future-potential-of-augmented-reality-trends-technology-and-impact/.

[167] Julie Rico and Stephen Brewster. Usable gestures for mobile interfaces: Evaluating social acceptability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 887–896, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589299. doi: 10.1145/1753326.1753458.

[168] Franziska Roesner, David Molnar, Alexander Moshchuk, Tadayoshi Kohno, and Helen J. Wang. World-driven access control for continuous sensing. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 1169–1181, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329576. doi: 10.1145/2660267.2660319.

[169] Sol Rogers. Seven reasons why eye-tracking will fundamentally change vr, February 2019. URL https://www.forbes.com/sites/solrogers/2019/02/05/seven-reasons-why-eye-tracking-will-fundamentally-change-vr/?sh=6cdf4493459a.

[170] Volker Roth, Kai Richter, and Rene Freidinger. A pin-entry method resilient against

shoulder surfing. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 236–245, 2004.

[171] Kimberly Ruth, Tadayoshi Kohno, and Franziska Roesner. Secure Multi-User content sharing for augmented reality applications. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 141–158, Santa Clara, CA, 2019. USENIX Association. ISBN 978-1-939133-06-9. URL https://www.usenix.org/conference/usenixsecurity19/presentation/ruth.

[172] Alia Saad, Dina Hisham Elkafrawy, Slim Abdennadher, and Stefan Schneegass. Are they actually looking? identifying smartphones shoulder surfing through gaze estimation. In *ACM Symposium on Eye Tracking Research and Applications*, ETRA '20 Adjunct, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371353. doi: 10.1145/3379157.3391422.

[173] Joe Saballa. Us army oks acquisition of 5,000 ivas goggles after year-long delay, 2022. URL https://www.thedefensepost.com/2022/09/05/us-army-ivas-goggles-2/.

[174] Eric E. Sabelman and Roger Lam. The real-life dangers of augmented reality. *IEEE Spectrum*, 52(7):48–53, 2015. doi: 10.1109/MSPEC.2015.7131695.

[175] Timothy A. Salthouse and Cecil L. Ellis. Determinants of eye-fixation duration. *The American Journal of Psychology*, 93(2):207–234, 1980. ISSN 00029556. URL http://www.jstor.org/stable/1422228.

[176] Jodi Schulz. Eye contact: An introduction to its role in communication, 20212.

[177] Julia Schwarz. Csvlogger, March 2021. URL https://gist.github.com/julenka/20ece5141e821cb6d0a9cb530d3dc96d.

[178] Monib Sediqi. Eye tracking with mediapipe, Feb 2023. URL https://kh-monib.medium.com/title-gaze-tracking-with-opencv-and-mediapipe-318ac0c9c2c3. https://kh-monib.medium.com/title-gaze-tracking-with-opencv-and-mediapipe-318ac0c9c2c3.

[179] Sefik Ilkin Serengil. Deep face detection with mtcnn in python, 2022. URL sefiks.com/2020/09/09/deep-face-detection-with-mtcnn-in-python/.

[180] Sefik Ilkin Serengil and Alper Ozpinar. Hyperextended lightface: A facial attribute analysis framework. In *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–4, 445 Hoes Lane, Piscataway, NJ 08854., 2021. IEEE, IEEE. doi: 10.1109/ICEET53442.2021.9659697.

[181] Ayushi Shal, Ayan Dey, Rana Majumder, Anirban Mitra, Shilpi Sharma, and Pratik Bhattarcharjee. A review on different techniques for human-computer communication using eye movement. In *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, pages 1225–1230, 2022. doi: 10.1109/IC3I56241.2022.10073174.

[182] Tom Sharman. Is eye tracking the future of virtual reality?, July 2020. URL https://medium.com/virtual-library/is-eye-tracking-the-future-of-virtual-reality-c5af7b0763f3.

[183] Jeff Shepard. What sensors are used in ar/vr systems?, 2022. URL https://www.sensortips.com/featured/what-sensors-are-used-in-ar-vr-systems-faq/.

[184] Rashik Shrestha. Perspective-n-point(pnp), 2023. URL https://medium.com/@rashik.shrestha/perspective-n-point-pnp-f2c7dd4ef1ed. https://medium.com/@rashik.shrestha/perspective-n-point-pnp-f2c7dd4ef1ed.

[185] Jiayu Shu, Rui Zheng, and Pan Hui. Cardea: Context-aware visual privacy protection from pervasive cameras. *CoRR*, abs/1610.00889, 2016.

[186] Ivo Sluganovic. *Security of mixed reality systems: authenticating users, devices, and data*. PhD thesis, University of Oxford, December 2018.

[187] Ivo Sluganovic, Marc Roeschlin, Kasper B. Rasmussen, and Ivan Martinovic. Using reflexive eye movements for fast challenge-response authentication. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 1056–1067, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978311.

[188] Ivo Sluganovic, Matej Serbec, Ante Derek, and Ivan Martinovic. Holopair: Securing shared augmented reality using microsoft hololens. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, ACSAC 2017, page 250–261, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450353458. doi: 10.1145/3134600.3134625.

[189] Ivo Sluganovic, Mihael Liskij, Ante Derek, and Ivan Martinovic. Tap-pair: Using spatial secrets for single-tap device pairing of augmented reality headsets. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, CODASPY '20, page 61–72, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371070. doi: 10.1145/3374664.3375740.

[190] Julian Steil, Marion Koelle, Wilko Heuten, Susanne Boll, and Andreas Bulling. Privaceye: Privacy-preserving head-mounted eye tracking using egocentric scene image and eye movement features. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367097. doi: 10.1145/3314111.3319913.

[191] Scott Stein. Watching me, watching you: How eye tracking is coming to vr and beyond, February 2022. URL https://www.cnet.com/tech/computing/watching-me-watching-you-how-eye-tracking-is-coming-to-vr-and-beyond/.

[192] Brian Jay Tang and Kang G. Shin. Eye-Shield: Real-Time protection of mobile device screen information from shoulder surfing. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5449–5466, Anaheim, CA, 2023. USENIX Association. ISBN 978-1-939133-37-3. URL https://www.usenix.org/conference/usenixsecurity23/presentation/tang.

[193] Techjury. 29+ augmented reality stats to keep you sharp in 2022, 2022. URL https://techjury.net/blog/augmented-reality-stats/.

[194] Tobii. What is eye tracking?, 2023. URL https://www.tobii.com/learn-and-support/get-started/what-is-eye-tracking.

[195] TP-Link. Ac1900 wireless mu-mimo wi-fi 5 router, 2023. URL https://www.tp-link.com/us/home-networking/wifi-router/archer-c80/. https://www.tp-link.com/us/home-networking/wifi-router/archer-c80/.

[196] Khai N. Truong, Shwetak N. Patel, Jay W. Summet, and Gregory D. Abowd. Preventing camera recording by designing a capture-resistant environment. In *Proceedings of the 7th International Conference on Ubiquitous Computing*, UbiComp'05, page 73–86, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 9783540287605. doi: 10.1007/11551201_5.

[197] Dorin Ungureanu, Federica Bogo, Silvano Galliani, Pooja Sama, Xin Duan, Casey Meekhof, Jan Stuhmer, Thomas J. Cashman, Bugra Tekin, Johannes L. Schonberger, Bugra Tekin, Pawel Olszta, and Marc Pollefeys. HoloLens 2 Research Mode as a Tool for Computer Vision Research. *arXiv:2008.11239*, 2020.

[198] Unity. Gameobject, oct 2022. URL https://docs.unity3d.com/ScriptReference/GameObject.html.

[199] Varjo. Varjo xr-3, 2023. URL https://varjo.com/products/xr-3/.

[200] Jonathan Vaughan and Thomas M. Graefe. Delay of stimulus presentation after the saccade in visual search. *Perception & Psychophysics*, 22(2):201–205, Mar 1977. ISSN 1532-5962. doi: 10.3758/BF03198755. URL https://doi.org/10.3758/BF03198755.

[201] Stylianos I. Venieris, Ioannis Panopoulos, and Iakovos S. Venieris. Oodin: An optimised on-device inference framework for heterogeneous mobile devices. In *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 1–8, 2021. doi: 10.1109/SMARTCOMP52413.2021.00021.

[202] Roel Vertegaal, Robert Slagter, Gerrit van der Veer, and Anton Nijholt. Eye gaze patterns in conversations: There is more to conversational agents than meets the eyes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, page 301–308, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581133278. doi: 10.1145/365024.365119.

[203] Wang, Sung, and Ronda Venkateswarlu. Eye gaze estimation from a single image of one eye. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 136–143 vol.1, 2003. doi: 10.1109/ICCV.2003.1238328.

[204] Monica White. What is the metaverse? a deep dive into the 'future of the internet', December 2021. URL https://www.digitaltrends.com/computing/what-is-the-metaverse-the-future-of-the-internet-explained/.

[205] XReal. Xreal air, 2023. URL https://www.xreal.com/air/.

[206] Lin Yang, Wei Wang, and Qian Zhang. Secret from muscle: Enabling secure pairing with electromyography. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, SenSys '16, page 28–41, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342636. doi: 10.1145/2994551.2994556.

[207] Rajeev Yasarla, Federico Perazzi, and Vishal M. Patel. Deblurring face images using uncertainty guided multi-stream semantic networks. *IEEE Transactions on Image Processing*, 29:6251–6263, 2020. doi: 10.1109/TIP.2020.2990354.

[208] Hidir Yüzugüzel, Jari Niemi, Serkan Kiranyaz, Moncef Gabbouj, and Thomas Heinz. Shakeme: Key generation from shared motion. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 2130–2133, 2015. doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.316.

[209] Tengxiang Zhang, Xin Yi, Ruolin Wang, Yuntao Wang, Chun Yu, Yiqin Lu, and Yuanchun Shi. Tap-to-pair: Associating wireless devices with synchronous tapping. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(4), dec 2018. doi: 10.1145/3287079.

[210] Yunfan Zhang, Tim Scargill, Ashutosh Vaishnav, Gopika Premsankar, Mario Di Francesco, and Maria Gorlatova. Indepth: Real-time depth inpainting for mobile augmented reality. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 6 (1), mar 2022. doi: 10.1145/3517260.