Security of Mixed Reality Systems: Authenticating Users, Devices, and Data

Ivo Sluganovic

St Cross College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2018

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. Ivan Martinovic. I am grateful for both the freedom and the support that you gave me during my D.Phil degree, which allowed me to contribute to a range of different topics in system security. I absolutely enjoyed our numerous open-hearted discussions, in which you were honestly enthusiastic not only to give academic guidance, but most importantly, to share your personal experiences and insightful life lessons. Thank you for being a true mentor.

I am very grateful to have had a chance during my D.Phil. to collaborate with amazing researchers from several different institutions. These research collaborations have allowed me to broaden my horizons, meet new inspiring colleagues and tackle challenging problems. I would especially like to thank Prof Srdjan Capkun and Prof Ante Derek for all the challenging and open discussions that we led. Srdjan, thank you for hosting me at ETH Zurich. Ante, thank you for making me feel like I've never left the University of Zagreb.

I wish to also thank the reviewers of my published research. Your constructive feedback has improved my research and transformed it into this thesis.

I am most grateful and humbled by the unconditional love and support that I have always received from my family. Thank you for nurturing such a warm, encouraging, and stable home, which has allowed me to calmly develop into a person that I am today. You truly are my greatest inspiration.

It is a wonderful and peaceful feeling to know you've found someone who you belong with, and to know that they feel the same for you. Viktorija, thank you for all your kindness and all the beautiful moments that we shared. Thank you for staying close even when we were far apart. And now that the distance is gone, I can not wait to see what lies ahead :)

Finally, this journey would not be as engaging and fun without the amazing friends that kept me company along the way. The fun, but also the serious discussions that we had in RHB 101, the ambitious projects that we started at Vibby and in Penkala, and the happy moments that we spent together have all given me the enthusiasm to proceed even when research did not always go as I hoped for. Thank you!

Abstract

Mixed reality devices continuously scan their environment in order to naturally blend the virtual objects with the user's real-time view of their physical environment. Given the potential of these technologies to profoundly change how individuals interact with their environments, many of the largest technology companies are releasing their mixed reality systems and devoting significant resources towards achieving technological leadership in this field.

However, due to the recency of the first commercially available mixed reality devices and their specific interaction channels, existing research has yet to provide practical proposals to achieve many of the core security objectives. Furthermore, given that these devices continuously analyze their environment using multiple front-facing cameras, when designing secure system it becomes necessary to reassess the commonly assumed threat models.

In this thesis, we aim to bridge this gap by focusing on secure authentication on mixed reality headsets. Taking into account the stronger assumed adversary models and the interface capabilities of existing mixed reality devices, we propose methods for user and device authentication, as well as show how these devices can be used to secure one's interactions with legacy systems.

Considering that mixed reality headsets are starting to support gaze tracking, in this thesis we propose, build a prototype and experimentally evaluate the use of reflexive eye movements as a biometric modality that is well suited as an authentication method on MR headsets. As an added benefit, the reflexiveness and predictability of one's eye movement responses makes it possible to incorporate the biometric measurements into challenge-response protocols. This allows the system to prevent replay attacks, one of the most common attack vectors on biometrics.

Furthermore, given the many multi-user applications of mixed reality technologies that rely on direct communication between users' devices, in this thesis we research secure and usable methods to mixed reality headsets. We propose a practical pairing protocol, implement a system prototype using two commercially available mixed reality headsets and evaluate its security and usability.

Finally, we show that front-facing cameras of mixed reality headsets can also serve as the means of securing legacy electronic systems. We therefore build and evaluate a prototype of a system that uses a trusted device with video capture and analysis capabilities to authenticate the data that the user inputs when using a potentially compromised local client to communicate with a remote server.

Contents

List of Figures

List of Tables

List of Abbreviations

2D . . . . . Two-Dimensional

2FA . . . . . Two Factor Authentication

3D . . . . . Three-Dimensional

AR . . . . . Augmented Reality

CDF . . . . . Cummulative Distribution Function

EER . . . . . Equal Error Rate

EMG . . . . . Electromyography

FAR . . . . . False Accept Rate

FPV . . . . . First Person View

FRR . . . . . False Reject Rate

GLM . . . . . Generalized Linear Model

GPS . . . . . Global Positioning System

HPU . . . . . Holographic Processing Unit

HTER . . . . . Half Total Error Rate

HTTPS . . . . Hyper Text Transfer Protocol Secure

IBAN . . . . . International Bank Account Number

ICC . . . . . Intraclass Correlation

IIR . . . . . Infinite Impulse Response

IMU . . . . . Inertial Measurement Unit

IR . . . . . Infrared

kNN . . . . . k-Nearest Neighbors

LDA . . . . . Linear Discriminant Analysis

LIDAR . . . . Light Detection and Ranging

MITM . . . . . Man-in-the-middle

MR . . . . . Mixed Reality

MS . . . . . Microsoft

OCR . . . . . Optical Character Recognition

PKI . . . . . Public Key Infrastructure

POI . . . . . Proof of Intent

RBF . . . . . Radial Basis Function

RF . . . . . Random Forest

RMI . . . . . Relative Mutual Information

ROC . . . . . Receiver Operating Characteristic

RSA . . . . . Rivest-Shamir-Adleman Algorithm

SDK . . . . . Software Development Kit

SHA . . . . . Secure Hash Algorithm

SMI . . . . . SensoMotoric Instruments

SUS . . . . . System Usability Scale

SVM . . . . . Support Vector Machine

TAN . . . . . Transaction Authorization Number

TAR . . . . . True Accept Rate

TLS . . . . . Transport Layer Security

TRR . . . . . True Reject Rate

UI . . . . . User Interface

UWP . . . . . Universal Windows Platform

VR . . . . . Virtual Reality

WMR . . . . . Windows Mixed Reality

XR . . . . . Extended Reality

In a time not distant, it will be possible to flash any image formed in thought on a screen and render it visible at any place desired. The perfection of this means of reading thought will create a revolution for the better in all our social relations.

— Nikola Tesla, 1915.
The wonder world to be created by electricity [1]

1 Introduction

Contents

1.1 Motivation

Mixed reality (MR) devices continuously scan their environment in order to naturally blend the virtual objects with the user's real-time view of their physical environment. The natural interaction between virtual and physical objects is the core difference of MR device in comparison to virtual reality (VR) devices, which fully immerse their users into an artificial world, and augmented reality (AR) devices, which simply overlay the digital information over one's view of the physical world.

While initial research in VR technologies began as early as the 1960s [2], the computing power of off-the-shelf devices has only recently become sufficient to support augmented and mixed reality experiences on a wider scale. The potential of

these novel technologies to rapidly capture public interest can be best exemplified by the success of the Pokemon Go mobile application. In its first week since launching, this location-based AR version of the popular game was downloaded more than any other application in history and has since then been installed more than 800 million times [3]. The application is the likely explanation for the surge in battery pack sales, which have doubled in the three weeks following its launch [4], while its requirement that players physically visit different real-world locations to play the game prompted the experts to suggest it as a population-level strategy to increase the physical activity levels [5].

Consequently, many of the largest companies are devoting significant resources to achieve technological leadership in this area and placing MR into focus of their near- and mid-term plans[1]. Recent examples include Facebook making MR the central topic of their 2017 developer conference, Apple recruiting top experts [7] and acquiring MR startups [8] to reportedly develop their own MR headsets, and Google investing more than $540 million into an MR startup Magic Leap [9]. In less than a year since September 2017, all three companies have unveiled their mixed reality frameworks for smartphones and tablet devices: Apple released ARKit and ARKit2 [10], Google immediately followed by releasing the core components of its Project Tango [11] as ARCore [12], and Facebook released their SparkAR platform [13] in September 2018.

However, since releasing their HoloLens headset in March 2016, Microsoft (MS) is currently considered to be the leader in mixed reality technologies. HoloLens is a self-contained, fully untethered head-mounted computer, the first publicly available headset that supports many of the core mixed reality concepts. Built upon the fundamental research from the Kinect project, HoloLens's main innovation is its remarkably precise inside-out position tracking that does not rely on any outside sensors or markers. Furthermore, its holographic display enables rendering realistic 3D virtual objects (holograms) that closely interact with the physical environment.

[1] As an example, Apple's CEO Tim Cook recently said in an interview [6]: *"I regard it as a big idea like the smartphone. (...) I think AR is that big, it's huge. I view AR like I view the silicon here in my iPhone, it's not a product per se, it's a core technology."*

Figure 1.1: Example of a shared mixed reality experience. Some of the participants wearing Microsoft HoloLens are co-located, while some are virtually present. As a result of sharing the common anchor in the physical space, all of the co-located participants can interact with the object at precisely the same location. Source: [14]

For example, a virtual ball *thrown* in a living room would bounce off the floor, collide with the chair, and disappear from the user's screen as it rolls under the sofa. While HoloLens is currently the only widely available commercial device with such capabilities, this is rapidly changing, as manufacturers such as Magic Leap [15], Meta 2 [16], Dell, HP, and Lenovo [17] all recently announced or released developer versions of their mixed reality headsets.

Besides individual use, mixed reality headsets open a range of possibilities for collaboration of co-located users, in which they all experience precisely the same virtual objects embedded in their shared environment (Figure 1.1). One such example is NASA's Jet Propulsion Laboratory, which already *"allows their scientists to work on Mars"* by using HoloLens devices to train, plan, and execute their Mars rover missions [18]. Other examples range from doctors using HoloLens during spinal chord and brain surgeries [19], to defense forces using the device for training and mission planning [20]. In a recent confirmation of this technology's commercial potential, Microsoft won a multi-year $480 million contract with the US Army [21], which requires the company to develop new capabilities and produce upwards to 100,000 headsets for both training and field deployment. Finally, basic mixed reality capabilities are already shipping with latest Windows 10 operating

system, while MS has recently released *SharePoint Spaces*, an extension of their collaborative suite specifically geared towards mixed reality [22], which further increases the number of business users of this technology.

The recency of the first mixed reality headsets becoming commercially available results in a very limited scope of academic research on the challenges related to their security and privacy. The existing research efforts have thus been mainly limited to positional papers that laid out general topics of interest [23] and discussed how to ensure that MR output on multi-app devices does not become malicious [24]. Furthermore, several researchers proposed solutions to ensure the privacy of bystanders despite the necessity of MR headsets to use front-facing cameras to understand their surrounding [25, 26, 27, 28].

Given the differences in the interaction channels that MR headsets rely on, e.g., having a private display and lacking the physical keyboard input, it is therefore necessary to consider novel ways of addressing challenges such as authenticating the user of an MR headset or establishing a secure communication channel between devices operated by legitimate users (device pairing).

In our research, we are thus inspired by the mixed reality technologies as a platform with novel user interaction capabilities, which in turn require that fresh approaches are considered in order to achieve specific security guarantees. Furthermore, we are also motivated by the changes to the security landscape brought by the proliferation of mixed reality headsets. For example, given that these devices require multiple always-on front-facing cameras to scan their environment, it becomes necessary to re-evaluate the commonly used threat models and design systems with the assumption that many of the user's sensitive interactions are happening under visual surveillance.

Consequently, the main focus of this thesis is in proposing novel ways of authentication on mixed reality systems under stronger adversary models than those which are currently typically assumed. However, as we show in the last part of this thesis, these technological advances can also provide ways of securing existing user interactions that include their traditional devices, such as laptops. We show

how visual supervision can be used to protect the authenticity of user's input as its being sent to the remote server, even in the case when the local client (such as a laptop) is compromised and fully under the adversary's control.

1.2 Contributions of Our Research

This section describes the contributions made by our research and outlines the publications that provide the foundation of this thesis.

The work presented in this thesis focuses on the security challenges of mixed reality headsets. More specifically, we are motivated by the need to find usable and secure methods of authentication on MR systems, as well as the possibilities that such devices provide to secure existing, legacy systems. Consequently, the core of this thesis comes from several of our system security publications [29, 30, 31, 32], all centered around the shared research topic: using various challenge-response protocols to achieve usable authentication of users, devices, and data in the stronger adversary models than the ones typically assumed in an authentication scenario. The used methodology consists of proposing the system designs, implementing their prototypes, and evaluating their security guarantees and usability performance in a series of user experiments, usability evaluations, and experimental measurements.

User authentication. The security and usability challenges related to stronger adversary models are particularly evident in the case of user authentication. User authentication is commonly achieved by having the user prove their knowledge of some secret to the target device, for instance by typing in their password. However, if the secret is leaked during the authentication attempt, for instance by visual observation or device compromise, this subsequently allows the adversary to bypass the authentication mechanism. Furthermore, besides security, a crucial consideration for user authentication is usability, commonly measured by the time it takes for an individual to prove their identity and the likelihood that they will be falsely rejected.

Considering the stronger threat model and the capabilities of existing and announced devices that we describe in the next chapter, in Chapter 4 we propose the use of eye movement biometrics for fast user authentication on mixed reality headsets. This chapter is largely based on our original research [29, 31] that was the first to propose the use of interactive visual stimulus to specifically elicit reflexive eye movements for user authentication. This enables the authentication system to quickly extract stable biometric features, which results in shorter authentication times and lower error rates than existing gaze-based authentication systems. The second contribution of this chapter is in proposing to integrate biometrics with a challenge-response type of protocol. This allows the system to verify the freshness of the received biometric response and thus prevent replay attacks, arguably the most accessible attack vector against biometrics.

Device Authentication. The need for secure authentication goes beyond users. In this thesis we also investigate the methods to mutually authenticate mixed reality devices (headsets) that have not previously established a secure communication and do not have a trusted third party that could aid them in achieving this goal. While device pairing is a topic that has been extensively researched in the past for various combinations of devices, pairing two mixed reality headsets brings about specific challenges and constraints. For example, users must not be required to take their headsets off during the procedure, but the device has no way of outputting any information to protocol participants other than its owner.

Chapter 5 is therefore based on our original research [30] that proposes a new protocol and a usable shared secret confirmation method for device pairing that does not require the users to take their MR devices off their heads during the procedure. Furthermore, in contrast to earlier research on security of mixed reality systems, the research presented in this chapter was the first security research that actually used two Microsoft HoloLens mixed reality headsets to build and evaluate

a fully functional MR system prototype.

Data Authentication. Finally, the new capabilities of mixed reality devices do not only raise the threat assumptions and require new means to achieve existing security objectives; they also allow improving the security guarantees of existing, legacy systems. In Chapter 6 we therefore describe our original research [32] on using mixed reality devices to authenticate user's data despite a potentially compromised local client. This is done by having a camera of the trusted mixed reality device, such as a mobile phone or a headset, observe one's interaction with the client device (e.g. a laptop), extract the contents of user's input on the client's screen, and provide an independent proof to the remote server that the data submitted by the client has indeed been input by the legitimate user.

To the best of our knowledge, this is the first research which aims to achieve authentication of user input via visual observation of another device's screen. As we discuss in the following chapters, while seemingly straightforward, providing security guarantees using visual observation requires that several subtle details are taken into account. To show the feasibility of this novel approach of continuous visual supervision, we build a prototype of the proposed system and evaluate it by running experimental and user studies.

To summarize, the core contributions of this thesis are in proposing, building, and evaluating prototypes of cyber-physical systems that achieve authentication of users, devices, and data despite the changes brought by, and with the help of novel capabilities of mixed reality systems. More specifically, for user authentication, we propose a novel method of incorporating a challenge-response protocol with eye movement biometrics, which allows us to prevent replay attacks. For device authentication, the presented work is the first published security research that proposed and evaluated a prototype for practical device pairing scheme on two mixed reality headsets. For data authentication, we introduce the novel concept

of having a trusted device serve as a continuous second factor for data that user inputs by performing visual supervision of another device's screen.

1.3 Published Work

This section gives an overview of the scientific publications and other outcomes of the doctoral work presented in this thesis.

1.3.1 Main Publications

The foundation of this thesis comes from the following published peer-reviewed publications in which Ivo Sluganovic was the leading author:

[29] **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 (ACM CCS 2016), Oct. 2016.

[30] **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), Dec. 2017.

[31] **Ivo Sluganovic**, Marc Roeschlin, Kasper B. Rasmussen, and Ivan Martinovic. *Analysis of reflexive eye movements for fast replay-resistant biometric authentication.* In: ACM Transactions on Privacy and Security (ACM TOPS), Nov. 2018.

Furthermore, the core ideas of the research presented in this thesis have been described in a publication and a related patent application that are currently under review:

[32] **Ivo Sluganovic**, Enis Ulqinaku, Aritra Dhar, Daniele Lain, Ivan Martinovic, and Srdan Capkun. *IntegriScreen: Visually Supervising Clients to*

Continuously Authenticate User Input. Under review for the IEEE European Symposium on Security and Privacy. IEEE, 2019.

[33] Ivan Martinovic and **Ivo Sluganovic**. *Augmented Reality Security Assistant.* United Kingdom Patent Application No. 1802739.1. Jan. 2018.

1.3.2 Research Collaborations

Ivo Sluganovic has also been a co-author on several other published peer-reviewed papers, which partially inform this thesis:

[34] Marc Roeschlin, **Ivo Sluganovic**, Ivan Martinovic, Gene Tsudik, and Kasper Bone Rasmussen. *Generating Secret Keys from Biometric Body Impedance Measurements.* In: ACM CCS Workshop on Privacy in Electronic Society, 2016.

[35] Mika Juuti, Christian Vaas, **Ivo Sluganovic**, Hans Liljestrand, N. Asokan, and Ivan Martinovic. *STASH: Securing transparent authentication schemes using prover-side proximity verification.* In: Sensing, Communication, and Networking, 2017 14th Annual IEEE International Conference on, IEEE, 2017.

[36] Mario Frank, Tiffany Hwu, Sakshi Jain, Robert Knight, Ivan Martinovic, Prateek Mittal, Daniele Perito, **Ivo Sluganovic**, and Dawn Song. *Using EEG-Based BCI Devices to Subliminally Probe for Private Information.* In: ACM CCS Workshop on Privacy in Electronic Society, ACM, 2017.

[37] Mika Juuti, Christian Vaas, Hans Liljestrand, **Ivo Sluganovic**, N Asokan, and Ivan Martinovic. *Implementing Prover-Side Proximity Verification for Strengthening Transparent Authentication.* In: Sensing, Communication, and Networking, 2017 14th Annual IEEE International Conference on, IEEE, 2017.

[38] Giulio Lovisotto, Raghav Malik, **Ivo Sluganovic**, Marc Roeschlin, Paul Trueman, and Ivan Martinovic. *Mobile Biometrics in Financial Services: A Five Factor Framework.* Oxford University Technical Report CS-RR-17-03, 2017.

[39] Ivan Martinovic, Lukas Kello, **Ivo Sluganovic**. *Blockchains for Governmental Services: Design Principles, Applications, and Case Studies.* In: Working Paper Series, Oxford Centre for Technology and Global Affairs, 2018.

1.4 Thesis Outline

This thesis is structured as follows:

- **Chapter 2** introduces the reader with the necessary technical background and terminology related to mixed reality systems. The chapter continues by introducing several trends that challenge the conventional threat models and introducing the shared system design goals assumed in the remainder of this thesis.

- **Chapter 3** provides an overview of the existing research on the security and privacy of mixed reality headsets and concludes by discussing the open research challenges, which we address in detail in the subsequent chapters.

- **Chapter 4** proposes and evaluates the use of reflexive eye movements for biometric user authentication. Due to their speed and resistance to replay attacks, tracking one's eye movements is particularly well suited for user authentication on mixed reality headsets.

- **Chapter 5** describes the research on securely pairing two reality headsets in the presence of a strong adversary who can fully observe all users' interactions. We design, build a prototype using two Microsoft HoloLens devices and evaluate it in a series of experiments.

- **Chapter 6** presents our research on using front-facing cameras on smartphones and mixed reality headsets for visual supervision in order to secure users of legacy systems against client compromise. We show that the trusted device can record and analyze the untrusted device's screen to continuously authenticate data that the user inputs while they communicate with a remote server.

- **Chapter 7** finally concludes this thesis by summarizing the results, discussing the future work and providing the final remarks.

1.5 Work Done in Collaboration

The research outputs presented in this thesis have been conducted in collaboration with several other researchers, whose specific contributions we now acknowledge. Apart from the exceptions listed below, all other contributions and research efforts presented in this thesis are Ivo Sluganovic's own work, done under guidance of his supervisor, Prof. Ivan Martinovic.

For the work in Chapter 4, Prof. Kasper B. Rasmussen contributed with valuable discussions and provided constructive feedback on the experimental evaluation and the publication write-up. Marc Roeschlin contributed during many discussions about the idea of using reflexive eye movements for challenge-response biometric authentication. Marc also helped design the experimental setup and run the eye tracking experiments, as well as adapted the saccade and fixation detection algorithm used for data analysis.

For the work in Chapter 5, Prof. Ante Derek contributed with valuable discussions and suggestions related the idea of device pairing for mixed reality headsets and helped with the protocol design and the security evaluation. Matej Serbec contributed during prototype implementation, helped design and run the user study, and made sure that the publicly available source code is structured according to best development practices.

For the work in Chapter 6, Prof. Srdjan Capkun suggested that we start working on authenticating the user input via visual observation and provided valuable critical feedback and insights during all stages of the project. Enis Ulqinaku helped during implementation of the prototype smartphone application, contributed in many discussions of the system design and helped set-up and run the user study and the experimental evaluation. Aritra Dhar contributed during the implementation of the prototype's server component, helped structure the initial drafts of the publication, and contributed in discussions about the system design. Daniele Lain

contributed with the specifications and the implementation of the client-side web forms, helped design and implement the attack simulations, and contributed in discussions about the evaluation and system design.

I'm excited about augmented reality because unlike virtual reality, which closes the world out, AR allows individuals to be present in the world but hopefully allows an improvement on what's happening presently.

— Tim Cook, Apple [6]

2 Background and Design Goals

Contents

We start this chapter by providing the necessary technical background: defining the used terminology, describing the novel capabilities unique to mixed reality headsets, and providing a brief overview of Microsoft HoloLens, the first and currently the most widely used MR headset available on the market.

The second part of this chapter provides an overview of the emerging threats that mandate the need to change our assumptions about the adversary's capabilities when designing secure systems. Finally, we conclude by describing the shared

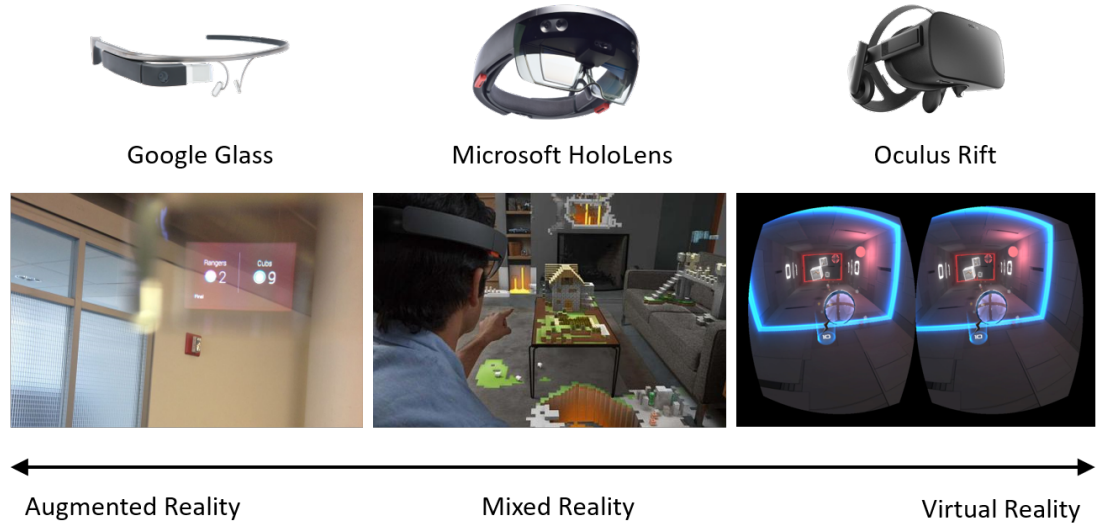


Figure 2.1: Overview of the AR-MR-VR continuum that shows several currently available headsets together with their typical user experiences. Google Glass [40] and other AR devices augment their user's view of the world with digital information, but with limited integration with the physical world, while VR devices (e.g., Oculus Rift [41]) fully immerse the user into a virtual environment, possibly augmenting it by integrating the camera feed of the real world. In between are MR devices like Microsoft HoloLens [14], which seamlessly integrate virtual objects with the physical world by taking into account, e.g., the shape and location of physical objects and the strength and direction of existing lighting.

design goals that we use throughout this thesis.

2.1 Terminology and Scope: VR, AR, and MR

We start by explaining the difference between three terms that are sometimes referred by the umbrella term Extended Realities (XR): augmented, mixed, and virtual reality.

In contrast to virtual reality, where one's environment is completely replaced with a virtual one, augmented and mixed reality technologies *augment* their users' natural view of the physical world by overlaying information, virtual objects, sounds, and other sensory stimulations over it. However, while the difference between VR and MR/AR is clear, the terms MR and AR are often used interchangeably, and even given slightly different meanings by different authors and organizations: for instance, while Apple insists on using the term AR in their communication, Microsoft emphasizes the term mixed reality, even for some devices that do not have true

MR capabilities. The main difference between the MR and AR devices (such as headsets, smartphones, or car windshields) is the level of integration between the virtual objects and their physical surrounding.

An overview of devices belonging to each category, together with the screenshots of their user experience is shown in Figure 2.1. We now describe each of the technologies and some of their core characteristics to define their meaning in this thesis.

Virtual reality (VR) technologies fully immerse the user into an artificial digital environment, which is usually decoupled from their physical surrounding. The VR headsets respond to users' head movements with perspective changes, ideally with a minimal lag in order to avoid any nausea. To the extent to which they are not limited by tethering and the physical obstacles, users are also able to freely move in space, usually holding specific controllers to interact with the environment. Example devices include Oculus Rift [41] and HTC Vive [42].

Augmented reality (AR) is a term mainly used to denote devices that overlay digital information over user's direct view of the physical worlds, but with limited integration between these two. Due to the lack of precise spatial mapping and positioning capabilities, they mainly display information in the form of heads-up-displays, or as virtual objects which do not interact with the environment. Given that virtual objects form a relatively smaller part of the user's view, these devices usually require less computational power and rarely cause lag-related discomfort or nausea, as can be the case with VR. Example headsets include Google Glass [40] and Epson Moverio BT-300 [43].

Mixed reality (MR) is a term used to describe those systems that continuously scan their environment in order to naturally overlay virtual objects onto user's view of the physical world such that they co-exist and interact in real time. An example experience would be having virtual objects, such as balls that collide with surrounding walls or game characters that can jump on desks and sit on chairs in a room. Other examples of MR interactions include being able to move virtual

objects using hand gestures or to precisely determine their location based on their spatial sound that the headsets generates.

In this thesis, we consider Microsoft HoloLens [14] as a typical MR device, which we describe in more detail in the remainder of this section, and which we also use to build a prototype of the MR system in Chapter 5. Other MR headsets that have recently become available include Magic Leap [15] and Meta 2 [16]. We continue this section by describing the main building blocks of MR experiences in more detail.

Finally, we note that in order to keep the scope of this thesis limited, we mainly focus on the security challenges brought by use of mixed reality headsets, as they best exemplify the novel components of mixed reality technology. However, we note that augmented and mixed reality technologies are also being implemented on other platforms, such as smartphones [44] and cars [45]. While we don't explicitly discuss these applications, the core concepts from this thesis can be translated to those platforms. For instance, the authentication method that we propose can also be implemented in cars that support gaze tracking, where tracking one's eyes is particularly useful to detect distraction and drowsiness [46]). Similarly, the system proposed in Chapter 6 was conceptualized with MR headsets in mind. However, we implement it on a smartphone due to the fact that today's high-end mobile phones still provide higher-quality cameras and more processing power than currently available MR headsets.

2.2 Core Capabilities of MR Headsets

In mid-2016, Microsoft released HoloLens [14], the first consumer-ready mixed reality headset that requires no outside markers to achieve precise inside-out spatial mapping and allows centimeter-scale positioning of high-resolution holograms. While several companies have since then released developer versions of their mixed reality headsets, such as Meta2 [16] and Magic Leap One [15], Microsoft HoloLens remains by far the most popular and arguably the most powerful device in the category. In this thesis, we therefore use HoloLens, both as a model of typical capabilities

Figure 2.2: Internal design of the HoloLens' environment understanding sensors: a depth camera, four gray-scale cameras, and a color video camera. The device also includes a 4-microphone array, an ambient light sensor and an inertial measurement unit. Source: [47]

of mixed reality headsets, but also to build an experimental prototype to evaluate the research in Chapter 5.

We now discuss some of its specific the capabilities that allow for truly immersive blend of virtual objects and the physical world.

2.2.1 Inside-Out Tracking and Spatial Mapping

The main innovation of Microsoft HoloLens over previously available augmented reality devices is considered to be precise inside-out position tracking and spatial mapping of the surrounding environment. The inputs from four *"environment understanding cameras"*, a depth camera, an ambient light sensor, and a 2MP video camera are combined in the custom made *Holographic Processing Unit* (HPU), shown in Figure 2.2. The HPU measurements allow the device to build and maintain a model of the surrounding objects. This model is the used to determine the device's location and is combined with the inertial measurement unit (IMU) measurements to achieve precise head and object tracking. The positions of virtual

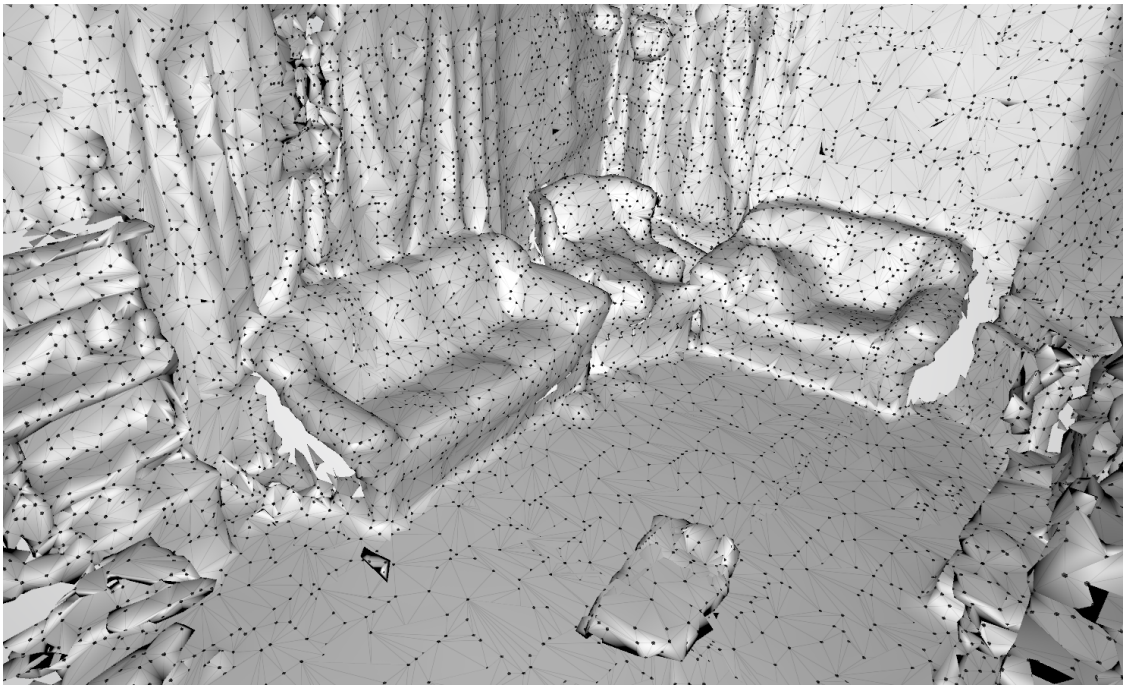


Figure 2.3: Example of HoloLens' spatial mapping capture in a typical living room. The headset uses multiple depth perception cameras to continuously scan its environment and build a 3D model of its surrounding world. This allows precise indoor localization of the device and ensures that holograms remain statically positioned in space. A mixed-reality view of the same room is shown in Figure 2.4.

objects (holograms) are thus fixed in space with centimeter scale precision, leaving an impression that they are completely static despite the user's movement.

In contrast to most other VR and MR headsets, all processing happens on the device, while the inside-out tracking does not require any tethered connection to another computer, thus allowing users to freely move in their environment.

Given that the device continuously builds and updates precise 3D models of its physical surrounding (such as the one shown in Figure 2.3), it is able to quickly distinguish between different indoor spaces, detect if it had been previously used in the current space, and load previously displayed holograms, which thus remain at their location over multiple sessions or even months. Finally, a crucial component of the immersiveness of the experience is that the developers of mixed reality applications easily access the continuously updated model of the physical surrounding. This allows them to create experiences in which holograms naturally interact with existing objects, such as walls, furniture, or humans. For instance,

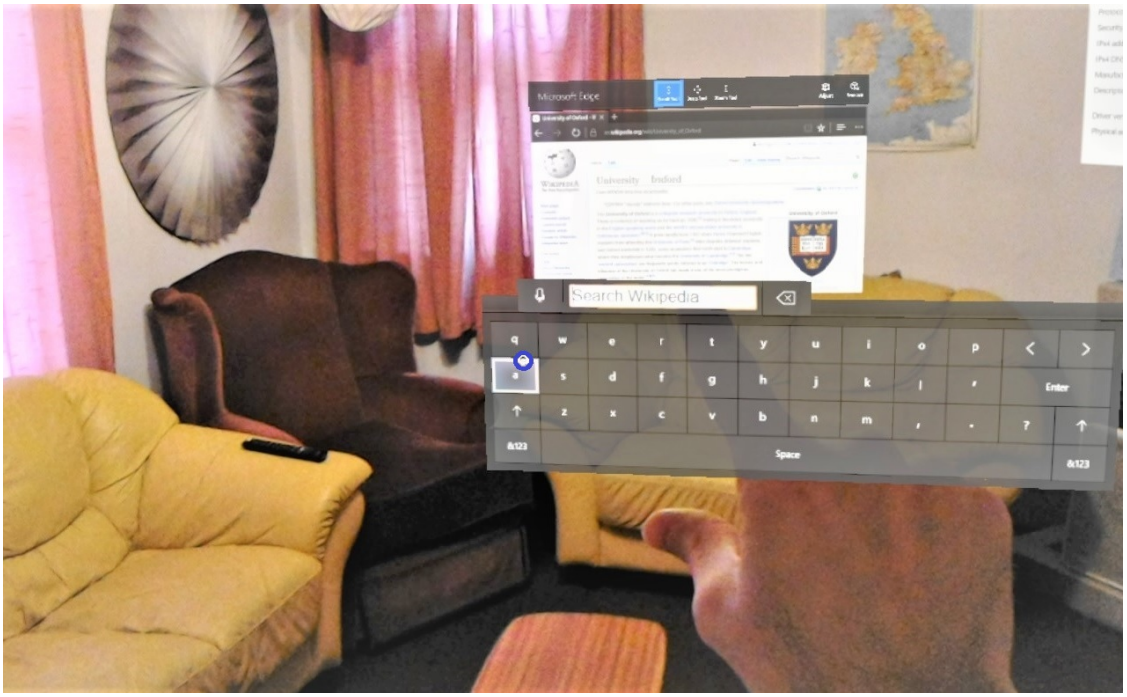


Figure 2.4: Interaction methods on Microsoft HoloLens. The user chooses the interaction object (i.e., a specific hologram, a user interface button, or a browser link) by directing their gaze at it, and using a hand gesture to specify *clicking*. The direction of user's gaze is indicated by the blue ring, currently positioned on the virtual keyboard's "a" key. The keyboard has been displayed to allow the user to search a Wikipedia page, which is loaded in a virtual windows that is displaying the Microsoft Edge browser. The text can also be input using speech recognition.

the immersiveness of an experience is highly increased if holograms disappear when they become occluded by physical objects or other holograms. As an example, when the user walks around a corner, their headset detects that holograms should not be displayed as the line of sight is now broken, further strengthening the feeling that the hologram is indeed located at a specific position in space.

2.2.2 Realistic 3D Holograms

In this thesis, we use the term *hologram* to denote a virtual three-dimensional object that the headset projects at either a specific location in space or at a specified position relative to user's position.

Besides displaying realistic 3D objects, the operating system (Windows 10 Holographic) allows running arbitrary applications built on the Universal Windows Platform (UWP) [48]. Each application is displayed in their *virtual window* that

the user can arbitrarily position, orient, and resize, such as the browser window rendering a Wikipedia page that is shown in Figure 2.4.

The immersiveness is further increased by the information from multiple cameras and the ambient light sensors that generate a model of the outside lighting and thus adapt the holograms' brightness and illumination accordingly. Furthermore, the device uses two speakers near user's ears to simulate the effect of binaurial audio, allowing each ear to receive slightly different sounds and thus simulate *spatial sound*, the sounds originating from holograms positioned in a location in space. Positioning the speakers close to one's ears allows for a natural blend between the sounds from the environment and those generated by the device.

The speakers are largely inaudible to the observers, while the content shown on the holographic display is only visible to the individual wearing the headset. Besides wireless networking, the device has no other output channel that could support even a low-bandwidth communication with bystanders or other devices.

2.2.3 Shared Mixed Realities: *World Anchors*

Shared mixed reality experiences, in which multiple users in the same physical space interact with identical holograms, are at the core of many current and future mixed reality applications. These range from collaboration during 3D modeling and planning [18], surgical procedures [19], to gaming and even military training [21, 20].

When multiple devices first start establishing a shared experience, they independently generate a spatial mapping of their environment and wirelessly exchange their models. One of the devices chooses an arbitrary location as an anchor and communicates its position within its model to all other devices, which compute a translation between spatial models received from the other devices. This anchor is then used as a reference point for the coordinate system, in which the shared holographic objects get synchronized to appear at the identical location to all participants.

2.2.4 Gesture Recognition

Due to the lack of keyboard or mouse input, gesture recognition is the main control channel in mixed reality experiences, used to manipulate and interact with objects, navigate menus, and even model or draw 3D objects in the space. Gesture recognition is achieved using the close-range depth perception cameras, which precisely track the position and orientation of users' hands and recognize several different hand gestures.

The standard interaction in mixed reality consists of the user gazing at a specific object and using their hand to perform the *click* gesture, pinching with their fingers to zoom, or spreading their fingers from a closed first to indicate the generic "back" command.

Textual input is achieved by either gazing and gesturing *clicks* on a virtual keyboard (shown in Figure 2.4) or using speech recognition, both of which we describe next.

2.2.5 Gaze Tracking

Gaze tracking is the core targeting interface within HoloLens' mixed reality experiences. The headset currently estimates the direction of one's gaze by tracking the orientation and position of their head. This vector is then raycast onto the continuously updated model of the environment (Figure 2.3) in order to determine the physical or virtual object with which the user intends to interact. As an example, in Figure 2.4, the position of user's gaze is shown by the blue ring, currently positioned on top of virtual keyboard's "a" key.

While the current version of Microsoft HoloLens uses one's head direction as an indication of interaction focus, the official developer documentation [49] and several technology trends strongly indicate that the majority of MR headsets will include a dedicated eye tracking module for this purpose. As a control channel, eye tracking greatly increases the speed and immersiveness of interactions while reducing neck strain and allowing more natural interactions, such as the ones where virtual avatars respond to one's gaze. Most importantly, the ability to precisely track where the user is looking is a necessary requirement for foveated rendering [50]. This rendering

technique relies on the fact that human visual system only perceives the full detail in the 5 degrees area around the gaze point and only renders this part in full resolution. Given that foveated rendering allows the remainder of the screen to be shown in lower resolution, it can significantly reduce the processing requirements of MR headsets.

Consequently, several companies are introducing prototypes of XR headsets that include eye tracking capabilities [51, 52], or announcing products that fit existing devices with this technology [53]. The recently released (August 2018) Magic Leap One MR headset [15] already includes eye tracking [54] as one of their main advantages over the other existing MR headsets. Furthermore, Microsoft has recently been granted a patent that proposes a new eye tracking method specifically suitable for head-mounted displays such as Microsoft HoloLens [55]. The innovation of the patent is in using a cluster of capacitive sensors to directly track the movement of the eye by detecting shifts of the cornea using its shape.

The next version of Microsoft HoloLens, and most future MR headsets are thus widely expected to include full eye tracking capabilities [56].

2.2.6 Speech Recognition

The last of the three major input channels supported on Microsoft HoloLens and other available MR headsets is speech recognition. It allows one to specify commands to its OS-level virtual assistant (Cortana) or to provide arbitrary commands defined by each of the applications running on the device. Using voice commands, users manipulate holograms, interact with the application interfaces, or dictate text instead of inputting it using the virtual keyboard and gesture recognition.

2.2.7 Operating System

Microsoft HoloLens runs a modified version of the Windows 10 operating system, which is part of the Windows Mixed Reality (WMR) platform [57].

Users are able to write and run arbitrary Universal Windows Platform (UWP) [48] applications: the standard, 2D applications are run as virtual windows (such as the browser shown in Figure 2.4), while the 3D applications, usually developed

in the Unity Framework [58], currently cause all other applications to move into the background and completely take over the MR output.

2.3 Emerging Threats

In this section, we cover the emerging trends that challenge the threat model that is conventionally assumed when designing and discussing cyber-physical systems security.

2.3.1 Conventional Threat Models

The threat model assumed in designing and securing many conventional systems aims to protect the legitimate user against a remote adversary who does not have physical access at the time of the legitimate user's interaction with the system. Security guarantees that such systems provide are thus evaluated under the assumption that the adversary is forced to guess the secret credentials involved in the interaction.

However, the security of such systems is compromised if the adversary gains physical co-presence or is able to observe the legitimate user. For instance, when using PIN and graphical authentication schemes to authenticate users to mobile phones, a co-present adversary can *shouldersurf* the secret information that is being input [59] and subsequently compromise the system at will. Additionally, previous research has shown that by recording and analyzing a video of user's authentication attempt, adversaries can extract the credentials even if only the backside motion of the device is visible [60].

As the result of proliferation of mixed and augmented reality headsets, as well as technologies such as CCTV cameras, home assistants, first person view drones, and computer-vision based smartphone applications, it becomes necessary to assume that many of our sensitive interactions with computer systems are likely to be visually observed.

Furthermore, a growing number of sensitive systems that users access from their local clients are remote (cloud-based). In such cases, the threat model usually assumes the possibility that user's secret credential might be leaked. To prevent

compromise if only one of the credentials is leaked, such systems often recommend or mandate that users use two or more different authentication factors (2FA) in order to authenticate themselves through their local client.

However, such threat models do not provide security guarantees the the client is compromised (as we discuss in Chapter 6), since the adversary can simply wait for successful user authentication (regardless of the number of factors) and subsequently arbitrary modify the communication with the remote client.

Considering the large number of applications that users non-critically install on their local devices, which rely on numerous external libraries, and the various zero-day vulnerabilities that are discovered every day, it is becoming necessary to assume that most electronic devices are likely to have been or will be compromised at some point.

2.3.2 Visual Observation

While the threat of visual observation and related compromise of one's sensitive data has been previously researched and argued for [61], we emphasize the several recent technology trends that make this a necessary assumption when designing secure systems.

As the number of different augmented and mixed reality headsets keeps rapidly increasing, we emphasize that multiple front-facing cameras are the core components of these devices. The input from these front-facing cameras (shown in Figure 2.2) is necessary to allow the precise spatial mapping, inside out positioning and realistic rendering of virtual objects. Consequently, the cameras are always actively scanning one's environment and potentially capturing sensitive bystander data in the process. Similarly, future self driving cars require not only LIDAR systems, but also multiple cameras that help recognize objects and pedestrians [62].

However, the threat of visual observation does not only come from these advanced technologies. As an example, drivers and cyclists are already increasingly installing dashcams on car dashboards and their helmets in order to continuously record evidence in case of accidents [63]. Similarly, the number of cameras in public spaces

is surging: the total market for CCTV and traffic light cameras has been growing by 25% annually for the past 3 years [64], which is in large part a result of raising security concerns, but also of different customer tracking and crowd analysis systems that are being installed in shopping centers, airports, and stadiums [65].

Neither public nor private spaces remain free from visual observation since an increasing number of individuals install *"smart cameras"*, which detect intruders while at the same time claiming to use face recognition to stop recording when legitimate tenants are indoors. Among many others, examples include Nest Cam [66] and Picollo Home Assistant [67]. However, the security of those devices is often lacking. The problem is only exacerbated by users not changing their default settings, allowing adversaries to use them to their advantage [68].

Finally, another trend that requires reevaluation of our expectations of visual privacy is the growing number of drones that are reaching the consumer markets, most of which support First Person View (FPV) flying and often include another higher-resolution camera for video capture. While their numbers and capabilities keep increasing, the prices and sizes of commercially available camera-equipped drones keep rapidly shrinking. For instance, Aerix Vidius, which can be ordered for as little as $70, fits between on the palm of one's hand with the dimensions of 4.3x4.3 cm [69], making the concept of insect-sized video-surveillance drones a not-so-distant possibility.

Together, these near-term trends and the existing threats significantly increase the likelihood that one's interaction with their electronic devices is being captured by a camera, not only as soon as they leave their home, but even while they inhabit their private spaces.

2.3.3 Device Compromise

Another important assumption for today's threat modeling is device compromise. Considering the number of applications that users install and the number of unpatched security exploits [70, 71], a compromise of a local client, such as

smartphone or a laptop is a realistic assumption that must be considered during system design [72].

This is particularly true for scenarios such as online banking [73], accessing private customer data [74], voting [75] or configuring industrial [76] or medical implants and devices [77], where a compromised local client likely results in direct financial, reputational, or even physical harm.

As a result of device compromise, user's secret credentials can be stolen not only by visual observation, but also by directly accessing them on a compromised device. This not only includes passwords but also biometric data, which is a larger problem due to its irrevocability. As such, it becomes crucial to design systems which can prevent an adversary from directly exploiting such a compromise in the future. For instance, it should not be possible to directly replay an obtained secret credential, be it a password or a biometric measurement, nor should adversaries be able to use the compromised clients for further, more damaging attacks.

2.4 System Design Goals

The aforementioned trends require that cyber-physical systems are designed with assumptions of potential visual observation of human-computer interactions and the possibility of compromise of existing devices.

Following the conclusions from the previous section, we now state the design goals that the systems must achieve in order to be considered secure under the stronger threat assumptions:

- **Resistance to impersonation.** In the context of biometric authentication systems, impersonation attacks happen when the adversary claims the identity of the legitimate user and aims to impersonate them by providing a biometric measurement. While the resistance to impersonation is the typical metric used to evaluate the security guarantees of biometric authentication systems, we argue that it is crucial to also design systems that are resilient to more advanced attacks.

- **Resistance to visual observation.** The developed schemes should assume the possibility that the adversary gains one or more recording of one's interaction with their electronic device. Unless system design prevents this, the adversary can use such observations to uncover secret knowledge, such as PINs, passwords, or shared keys. These attacks are particularly relevant in the context of user and device authentication (pairing).

- **Resistance to replay.** In order to be considered secure, systems must assume the possibility that the adversary has previously compromised some of the devices and obtained one's secret credentials, such as their password. In the context of user authentication, the adversary targets a specific user and replays their previously recorded authentication attempt to the authentication system. A replay attack can, for instance, happen as a result of the user being tricked into authenticating to a compromised device.

Rules for passwords:

1. *A good password should be hard to remember.*
2. *You should never write your password down.*
3. *No password should ever be reused.*

— anonymous

3 Security and Privacy of Mixed Reality Systems

Contents

In this chapter, we provide a summary and a comparative overview of the existing research that focuses on the security and privacy of mixed reality headsets. The research that we cover in this section is related, but not directly comparable with the core contributions that we describe in subsequent chapters. We provide the overview of directly related work in each of the subsequent chapters.

3.1 Overview

The topic of augmented and mixed reality has been the focus of a wide range of fundamental research in sensors, displays, location tracking, and user interfaces [78, 79] for more than 50 years [2]. However, it is only in the last decade that these technologies have matured and came sufficiently close to commercialization to warrant the active interest of security and privacy researchers.

An initial survey of challenges and directions in safely designing mixed reality systems was given by Roesner et al. [80]. They emphasize several security and privacy challenges raised by the application of AR technologies: from a large set of always-on sensors, to the complexity of controlling the mixed reality output, especially given the interplay between the physical world and multiple applications running simultaneously. Authors only briefly mention the challenges of authenticating the users to their devices and establishing wireless communication between AR-enabled technology.

The most recent and comprehensive overview of the growing body of security and privacy research for MR and AR is given in a literature survey by de Guzman et al. [81]. The survey covers a wide range of topics: from input, data, and output *protection*, to protecting user interactions and devices themselves.

We continue this chapter by providing an overview of the most related work in several categories, which range from ensuring that the always-on sensors do not infringe on one's privacy, ensuring that the MR output does not harm users, to using MR devices as an attack tool. We overview the research that shows MR technologies being used to improve security by improving on password managers, enabling visual cryptography schemes, and functioning as crucial components of various observation resistant authentication schemes.

Finally, we overview the existing research on biometric authentication in the context of mixed reality headsets. We note that, while some of the presented research in this chapter has been carried out on AR or VR headsets, we include those approaches that should also be directly applicable to mixed reality headsets.

3.2 Privacy

When discussing the privacy-related research challenges posed by mixed reality devices, the main motivation stems from the fact that these devices include multiple always-on sensors, such as RGB cameras, depth perception IR sensors, GPS, and microphones. While many other mobile devices also include environment sensors such as microphones and cameras, their operation typically does not mandate that they are continuously kept on. On the other hand, MR headsets rely on perpetual video capture and analysis, which potentially poses significant privacy concerns for bystanders.

The early research emphasized the privacy threats as a result of potential automatic recognition of bystander's faces [82], but offered little suggestions for improvements. Similarly, largely motivated by the backlash against Google Glass, Denning et al. [83] evaluated bystander's perspectives on the possibility of being recorded in public by other users' devices and offered suggestions that the devices should, for example, automatically blur all detected faces at the operating system level.

This line of reasoning has been closely followed by Zareopur et al. [26], who propose a system that automatically detects and blurs sensitive objects such as faces and license plates. A seemingly simple, yet challenging problem of detecting screens has been researched by Korayem et al. [27, 84]. *Darkly* [85] extends this approach by automatically sanitizing the details in the input video feed by applying various levels of visual filtering that preserves sufficient detail for underlying context-aware applications to still operate successfully.

On a similar note, solutions have been proposed that aim to automatically sanitize the video feed or prevent video capture in sensitive places [86, 87], or respect previously placed markers that specify what kind of input processing is allowed [88, 89].

Finally, a more specific application of the access policies is related to ensuring that applications operate at the least privilege model during gesture recognition [90], object recognition and depth perception necessary to project 3D objects in a given space, without being able to learn their location or even confirm their existence.

3.3 Security of MR Output

We start by discussing an important user interface capability that mixed reality headsets provide: the possibility of privately showing information only to their user and thus providing an important tool in designing systems that are resistant to continuous visual observation. The section continues with a discussion of security challenges related to mixed reality headsets: controlling the output capabilities of the devices in a multi-application environment to prevent user harm.

3.3.1 Observability of Displays

The displays on mixed reality headsets are designed to show a stereo depiction of virtual objects very close to or even directly on one's retina [91, 92]. The privacy advantages of wearing a display close to one's eyes are obvious in many cases where the sensitive content could be shoulder-surfed, such as during flights, in busy conference and lecture halls, or in other public spaces. Finally, despite the threat of continuous observation, assuming that users have a private personal display during sensitive interactions allows us to design protocols that require a secret channel between the user and their device.

However, the observation resistance of mixed reality displays should not be automatically assumed, as has been shown in research by Kohno et al. [93]. Their work has shown that it is indeed possible to eavesdrop the screen contents of two wearable displays: Silicon Micro Display ST-1080 [94] and Google Glass [40]. Authors also propose relatively simple measures to increase the privacy of their screens, such as adding a polarization filter.

3.3.2 Preventing Malicious Output

If one of the applications on an MR device is malicious or contains bugs, it could overlay important information over the other applications or the user's view of the real-world objects. Given the immersiveness of mixed reality technologies, it becomes possible that a malicious application directly causes physical harm by, e.g., changing the speed limit on an important traffic sign, or even fully hiding an

obstacle, such as a stair or a pothole [95]. Similarly, it has been argued that, unless controlled, malicious applications could cause physical harm by directly targeting the visual system of the user, e.g. by generating a high-speed flicker [96].

Initial research on the challenge of controlling MR output [23, 97] proposed a framework that enforces one of the several proposed output policies at the level of the operating system. Authors thus designed, built a prototype, and evaluated *Arya*, an AR platform that controls application output according to specified policies for complex multi-application MR operating systems. The prototype was, however, built using a VR system due to the lack of commercially available mixed reality headsets at the time of publication.

The ideas proposed by the Arya platform have recently been extended to allow deriving the policies that control AR output by supervised learning [98]. This is achieved by relying on simulations to train a deep learning model and thus automatically learning appropriate strategies for filtering potentially distracting or malicious content.

3.4 MR Headsets as System Components

After reviewing related research on privacy and security challenges of mixed reality headsets, we now present research in which mixed reality headsets are used as parts of larger systems, mainly to achieve security objectives on other devices. The research reviewed in this section relies mainly on the observation resistance of the mixed reality display, as well as the front-facing cameras of the headsets and their usability in conveying information with a spatial component.

3.4.1 Visual Cryptography

The basic idea of visual cryptography is to transmit an encrypted secret without it ever being decrypted by an electronic device. This can be achieved if the secret is presented in a visual form, usually a grid of black and white squares, and overlaid with a visual key in some form of a transparent layer. As a result, our visual system essentially performs a *visual OR operation*, thus decrypting the underlying secret

as the combination of the two grids. Since the MR device only overlays the key, it does not necessarily ever get access to the combined, decrypted data. However, this requires precise spatial alignment between the cipher and the key, which is hard to achieve. While the original idea was proposed by Shamir and Naor in 1994 [99], the advent of MR and AR headsets make its implementation practically possible on cyber-physical systems, spurring a renewed interest in this area of research [100, 101].

3.4.2 Augmenting Interactions with Other Devices

A promising application of mixed reality technologies is in augmentation and authentication of users' interactions with other electronic devices.

For example, if the mixed reality headset stores one's passwords, it can serve as a very usable password manager [80], given its ability to privately show the secret credential to the user. Furthermore, such an MR password management system could automatically detect which credential to use (based on the website URL or the visual interface of the application) and even provide input guidance on the user's keyboard, similar to existing MR approaches to teaching users how to play piano [102].

However, the adversary can still eavesdrop on the user as they are typing on the physical keyboard. This challenge was recently addressed by Maiti et al. [103], who propose using the mixed reality display to show shuffled key labels on a physical keyboard. In the case of mobile devices, several authors [104] and patents [105] have similarly proposed and evaluated systems that use an augmented reality headset to show the layout of a shuffled PIN-pad.

A similar concept was recently applied in an offensive scenario [106], where authors built a prototype and evaluated the applicability of an AR-based system to serve as a visual guide for attacking keystroke dynamics biometric authentication on touchscreen-based devices.

Finally, MR headsets could be used to detect potential compromise and malicious behavior on other electronic devices. Proposed examples include automatic detection of skimming on ATM devices by analyzing visual clues [107] or detecting unauthorized Bluetooth communication [108].

Our work in Chapter 6 broadly falls in this category: using a mixed reality device to guide and secure user's interactions with another electronic device. We comparatively overview other (non-mixed reality) approaches to achieving data authentication despite a compromised device in Section 6.10.

3.4.3 Shared MR Experiences

Given the advantage of mixed reality in supporting co-located collaboration between multiple users who all see the same virtual holograms, ensuring privacy and security guarantees during one's participation in shared MR experiences is an important area of research. Lebeck et al. [109] have recently investigated users' perception of security and privacy threats during single and multi-user experiences in several Microsoft HoloLens applications. They emphasize several future research challenges that arose from the user experiments, such as providing privacy for bystanders and controlling ownership of virtual objects.

Multi-user experiences are also a challenge from a system security perspective: when the users want to start a shared MR experience, how do the two or more headsets authenticate each other to establish a shared key? The challenge comes from the possibility that an adversary that controls the wireless communication positions himself as a man-in-the-middle (MITM), thus being able to eavesdrop on all subsequent communication. Gaebel et al. [110] were the first to focus on the problem of device pairing of mixed reality headsets. They propose a solution that relies on precise wireless localization and human verification of faces in order to prevent the adversary from impersonating another user. If each of the communicating devices can be precisely localized in space, then the headset can request the legitimate user to confirm the identity of the person wearing the device at a specific coordinate. Authors develop a prototype using two laptops and evaluate the provided guarantees using state of the art wireless localization algorithms. The achieved results mandate that devices are not more than 1 m apart in order to achieve at least 60% success rates. Given the shown limitations of wireless localization, we consider the problem of secure pairing to still be an important and open research challenge.

In Chapter 5 we therefore present another approach to achieving secure device authentication for mixed reality headsets. In our approach, users are guided by the precisely positioned holograms, which are identical only in the absence of attacks.

3.5 Observation Resistant Authentication

With the growing volume of sensitive data that is accessible on electronic devices, achieving secure user authentication is always an important research challenge. However, secure authentication often contrasts with usability: surveys show that more than half of users choose convenience over security and do not use any authentication to lock their devices [111]. While some users might not be fully aware of the security and privacy implications of their choices, such behavior is primarily influenced by the fact that traditional secret-based authentication methods, namely passwords and PINs, are hard to (re)create and remember [112], take a long time to input [113] and are easy to forget [114, 115]. For attackers, they are simple to guess [116] or exploit after a system breach [117], and can be shoulder-surfed during authentication attempts [118]. Furthermore, they are easily transferable to others, which makes them vulnerable to insider threats, as witnessed in some of the most prominent recent data leaks [119].

The need for novel authentication methods is especially evident in the case of mixed reality headsets, which do not have keyboard access, making any textual or numerical input slow, imprecise and easily observable.

Given the lack of research that focuses specifically on mixed reality headsets, and the applicability of various methods that were originally proposed for smart glasses or virtual reality headsets, in this section, our definition of mixed reality headsets is slightly relaxed, i.e., we also include related research that could be translated to mixed reality headsets, despite being carried out on AR and VR headsets. We now discuss two orthogonal approaches to user authentication on mixed reality headsets.

3.5.1 Secret-based Authentication

Unless specific care is taken in designing the authentication systems, secrets are commonly leaked, either by observation or by the victim using a compromised device. Therefore, it is important to incorporate observation resistance and, ideally, also resistance to device compromise. We now overview several secret-based observation-resistant authentication schemes that have been proposed and adapted for mixed reality headsets. The proposed schemes rely on using a (potentially scrambled) PIN-pad and mainly differ in the choice of the input channel.

In order to evaluate the applicability of established PIN and pattern unlock mechanisms in virtual reality, George et al. [120] ran a user study with 25 participants and measured the success rates and authentication times for several variants of the PIN and pattern unlock schemes in VR. The secrets were input using dedicated position-based controllers held in users' hands.

The approach of having a secret display that scrambles a PIN-pad has recently also been adapted for the context of MR headsets by proposing to use an EMG (electromyography) armband as an input device [121]. However, despite the use of a dedicated input device, the authors report a relatively low recognition rate of 80% - 93%, which would result in an impractically high FRR for users.

Finally, Yadav et al. [122] designed and evaluated an observation resistant authentication scheme for Google Glass that relies on the unobservability of its display in order to show a scrambled keyboard. Users then input their PINs, either by using touch gestures or speaking out numbers shown next to the desired digit. The results show that speech recognition achieved authentication times comparable to the default authentication on the device (6.4 vs. 5.6 seconds) while being observation resistant and achieving higher, but still relatively low success rates (83% vs. 68%).

While achieving observation resistance, secret-based authentication for mixed reality headsets still requires users to remember their PINs or passwords and input them using methods that are arguably not usable. However, the largest threat to secret-based schemes comes from their transferability: if we assume the possibility

of a device compromise, then as soon as the user authenticates to a compromised device, their authentication credentials are leaked.

In the next subsection, we thus overview proposed methods of biometric authentication, as they are inherently less transferable than secret-based authentication, and some could also be made resilient to replay after device compromise.

3.5.2 Biometric Authentication

Instead of relying on the proof of secret knowledge to verify user's identity, biometric authentication systems use unique physical or behavioral traits to distinguish between different individuals and verify one's identity. Physiological traits, such as fingerprints [123] or iris recognition [124, 125], usually remain more stable over time, while many behavioral characteristics, such as touchscreen usage patterns [126] or gait recognition [127], have an advantage of being less obtrusive and allow for continuous identity verification. Regarding usability, biometrics offer many benefits over other authentication methods, since they do not require the user to learn a new procedure or remember additional information like passwords, they can not get lost and they are not transferable to other users [128].

However, one of the most serious drawbacks of biometric authentication systems is their susceptibility to replay attacks and the presentation of artificial data or objects to the sensor. This is especially true for certain types of biometrics, such as fingerprints, where copies can be acquired from traces left unconsciously on numerous occasions in everyday life [129]. Unfortunately, more dynamic biometrics are not invulnerable to these types of attacks either: voice recognition, for instance, can be circumvented with the help of audio recordings [130], while many video-based face recognition systems with built-in liveness detection can be bypassed by simply showing a low-resolution 3D model [131].

Consequently, when deciding to use biometric authentication under the threat model in which continuous observation is expected, it is crucial to consider the likelihood that a biometric measurement can be collected and used for subsequent impersonation or replay attacks.

We now overview related work in several biometric modalities that could support secure and usable authentication of users to their MR headsets:

Sound recognition. Given the availability of microphones on mixed reality headsets, a relatively obvious approach to user authentication could be to analyze whether they *sound like them.* The traditional speaker recognition, while heavily researched [132, 133], is still prone to impersonation [134] and replay attacks [135]. However, Chauhan et al. argue that using the acoustics of one's breathing as a biometric [136] could result in a reliable and hard-to-replay biometric that supports continuous authentication. The initial pilot evaluation with a limited set of users shows the potential of this approach. However, the authors emphasize that more studies are needed to truly asses the performance of this biometric.

Skull conductance. Another potentially interesting biometric uses differences in skull conductance [137] between individuals. Authors propose using a bone conduction speaker on one ear and measuring the resulting sound by a microphone on the headset. A preliminary evaluation with 10 users shows that the system achieves an equal error rate (EER) of 6.9%. While this authentication method is inherently less observable, more work is needed to evaluate the applicability of this approach to devices that do not include conduction speakers.

Palm biometrics. As an example of static, physiological features being used for user authentication for mixed reality headsets, Epson has recently announced that their flagship headsets will allow authenticating users by capturing an image of their hand [43]. However, no research about the achieved error rates, or the possibility of spoofing and replay attacks has yet been published.

Movement characteristics. Finally, several approaches suggested using the idiosyncrasies of one's movement as a biometric trait.

Two recent papers proposed using head movements as a biometric: in the first one, users are required to move between two points in the virtual space [138],

while in the other, conveniently called *Headbanger*, users are required to perform a "music induced" head movement [139].

Furthermore, gait biometrics, which rely on analysis of user's movements, have also been recently proposed in the context of mixed reality headsets, considering their accelerometer and gyroscope sensors [140]. Authors report a high success rate of 98% after requiring users to make 5 steps. However, similarly to the proposed Headbanger system, it should be noted that one's movement characteristics are inherently susceptible to visual observation. It is interesting to note that previous research has indeed shown that, armed with the victim's recordings, even non-expert users can be trained to achieve significantly high rates of successful impersonation attacks against gate-based systems [141].

3.5.3 Gestural Interactions as a Biometric

Real-time tracking of users' hand movements while they perform various gestures in mid-air is becoming an important input channel for a range of applications, spanning from design, embedded systems control, to interactions in various forms of extended realities. Given that the majority of available mixed reality headsets already support a degree of hand tracking and gesture recognition, we now overview related work that focuses on using one's hand movements as a biometric.

Touchscreen gestures. Research on using gestures as a biometric initially started by focusing user authentication to on smartphones and tablet devices with touchscreens. Besides unlocking the device, touchscreen gestures are also analyzed in a continuous authentication scenario [126], in which the system aims to detect intrusion during general device usage (swiping). Reported results show that by analyzing only two of user's consecutive touch strokes, it is possible to achieve equal error rates between 5 and 15%, while the error rates stabilize around 5% when more than 20 strokes are taken into account before making the authentication decision.

When used as an unlock mechanism, users are required to perform one or more single- or multi-finger gestures on the touchscreen. Previous work achieved equal

error rates between 3 and 20%, depending on the number of required gestures, and whether they are user chosen [142] or provided as system defaults [143].

In-air gestures. Early research has shown that camera capture and analysis of whole body silhouettes as users perform various system-defined gestures [144] might be a potential authentication method, yielding EER rates between 5 and 6%. This approach was extended by the use of depth sensors such as Microsoft Kinect [145], which analyzed user's handwriting as they are required to *write* their passwords in the air [146]. While authors report very low error rates in case of random attacks, such a system was not particularly resistant to shoulder-surfing, allowing attackers to succeed in 23% of simulated attacks.

In other approaches that require users to perform a specific in-air gesture, authors have proposed authenticating users based on how they make different lower-scale hand gestures, tracked by a Leap Motion [147] sensor [148] or a short-range depth camera [149], and achieving low error rates between 3.4 and 13%. Another group of authors [150] analyzed the usability of such systems, finding a strong positive correlation between users' perception of usability of a gesture (based on the System Usability Scale [151]) and its authentication error rates.

Moving away from external sensors, a relatively small study proposed authenticating users by analyzing gyroscope and accelerometer readings as they perform a gesture while holding a smartphone in their hand [152]. However, the developed prototype did not achieve practical error rates: for all tested scenarios, the false reject rates remained higher than 15%.

Finally, when addressing the challenge of device pairing for mixed reality headsets, the approach that we take in Chapter 5 is most similar to previous work by Ahmed et al. [153], who looked at using a gesture to automatically pair e.g. a smartwatch with a screen by asking users to perform hand gestures while holding the device and comparing whether the gestures match the expectation.

3.5.4 Eye Movements as a Biometric

Considering the importance of gaze and eye tracking for the immersiveness of mixed reality headsets, we now discuss the possibility of using eye movements as a biometric for user authentication on these devices.

Even when one's gaze is firmly fixated on a single position, human eyes are never completely still. They are constantly making hundreds of micro-movements per second, which are interlaced with more than 100,000 larger movements during the course of one day. During visual tasks, such as search or scene perception, our eyes alternate between fixations and saccades. Fixations are used to maintain the visual focus on a single stimulus, while saccades reorient the eye to focus the gaze on a next desired position. Saccades are considered to be the fastest rotational movement of any external part of our body, reaching angular velocities of up to 900 degrees per second, and usually lasting between 20 ms and 100 ms. Eye movements are both quick and responsive, and although most movements happen subconsciously, some can be actively triggered, which allows the possibility of developing novel biometric authentication protocols [154, 155, 156].

The responsiveness and versatility of eye tracking motivate its use as an input channel in a broad range of devices and tasks, particularly those that do not support touch or mouse input. Consequently, as a result of technological advances and significant reductions in average retail prices [157], eye tracking devices are now being integrated into laptops [158], mobile phones [159], and cars [160].

As already discussed, foveated rendering and acquisitions of eye tracking companies and related patents filed by the largest tech giants such as Apple [8], Google [9], and Microsoft [55] give strong indications that eye tracking is becoming an important computer-human interface on mixed and virtual reality headsets and various other platforms.

Consequently, in Chapter 4 we propose and evaluate the use of reflexive eye behavior and unique low-level characteristics of individuals' gaze patterns to design a replay-resistant biometric that is suitable for mixed reality headsets.

3.6 Summary and Research Challenges

This chapter provided an overview of existing research that focuses on the challenges related to privacy and security of mixed reality systems.

We now overview the main research challenges that stem from the related work and the novel threat models and capabilities of mixed reality headsets, which we address in the following chapters of this thesis.

User Authentication. As personal devices that have access to one's sensitive data, it is crucial that MR headsets authenticate their users in a secure and usable way. We've thus overviewed the related work on secret-based and biometric authentication methods that have previously been proposed for mixed reality headsets, and we conclude that the majority are not resistant to visual observation or replay attacks after a device compromise.

Given the expectations that most future mixed reality headsets will include eye tracking capabilities, and the fact that some already do, tracking eye movements is likely to allow for fast, usable, and secure biometric authentication. Consequently, in Chapter 4 we propose and evaluate a system that uses reflexive eye movements to not only authenticate users quickly but also to prevent the adversary from directly using a biometric reading even in the case of device compromise.

Device Pairing. An important benefit of mixed reality headsets is that they allow multi-user shared experiences, in which the virtual objects are precisely aligned and synchronized across a group of users of this technology. However, if the group has not previously established a shared key for communication, it is crucial that they avoid becoming victim to man-in-the-middle attacks by confirming that the established encryption keys have not been modified by the adversary. However, since the first mixed reality headsets have only recently become commercially available, there are no practical proposals for usable device pairing of MR headsets. In Chapter 5 we therefore propose a novel method to authenticate two mixed reality headsets, implement the proposed system on two Microsoft HoloLens devices and experimentally evaluate its security and usability in a user study.

Authenticating User Input. Finally, as we argue in more detail in Chapter 6, in the case of using a local client to connect to a remote server (such as the case with most online services), authenticating the user is not sufficient if the client is compromised. The adversary in control of the local client can simply wait for the user to authenticate, and then forge or modify subsequent requests to the remote server without the user noticing.

Given the front-facing cameras on mixed reality headsets, and the realization that during input, users data is shown on the client's screen, it is an interesting research challenge to design a system which ensures that any request received from a client was indeed typed in by the legitimate user. We tackle this challenge in Chapter 6 by having a trusted device visually capture user's intended input from the potentially compromised client's screen, thus forcing it to behave honestly.

Right then, let's begin!

Don't underestimate the power of your vision to change the world. Whether that world is your office, your community, an industry or a global movement, you need to have a core belief that what you contribute can fundamentally change the paradigm or way of thinking about problems.

— Leroy Hood

4

Using Reflexive Eye Movements for User Authentication

Contents

After discussing the interfaces available on mixed reality systems in Chapter 2, and the existing related research on user authentication in Chapter 3, we now evaluate the feasibility of using eye movements for fast, replay-resilient biometric authentication on MR headsets. We thus propose an interactive stimulus and a corresponding challenge-response protocol, build a prototype using a static eye tracking device, run a user study and analyze its performance in a range of different attack scenarios. While the prototype that we build and evaluate in this chapter uses a static eye tracking device, we emphasize that due to its simplicity and speed, the proposed stimulus is well suited to be displayed on the mixed reality headset's display.

4.1 Introduction

Eye tracking devices capture precise position and movement of the human cornea on a millisecond scale. This, in turn, allows determining the exact location of one's gaze on a screen or on the surrounding objects. Since analyzing the eye behavior can give insight into our internal cognitive processes and even predict conditions such as autism [161], eye trackers have been used in neurophysiological research for over a century, but until recently their use in everyday life was limited due to the prohibitive equipment costs.

However, the speed and responsiveness of eye movements strongly motivate their use as an attractive input channel for human-computer interaction; as a result, recent years have brought a sharp reduction in retail prices of eye tracking devices. While dedicated trackers can now be purchased for as little as $100 [157], eye tracking capabilities are also being added to consumer products such as laptops [158], cars [162], tablets, mobile phones [159], and virtual [52] and mixed

reality headsets [50]. Given the diverse advantages and applications of eye tracking, its widespread expansion into our everyday lives is only likely to continue.

As we demonstrate in the following sections, tracking the user's gaze is particularly suitable for fast and low-effort user authentication, especially in scenarios where keyboard input is not available. Eye movements exhibit traits distinctive enough that classification algorithms (e.g., [156]) can reliably discern among a group of individuals. However, despite their advantages, exploiting eye movements for user authentication remains a challenging topic. As we summarize in Section 4.3, most previous work on gaze-based authentication achieves either high error rates (e.g., EER above 15%) or long authentication times (e.g., above 20 seconds). One likely explanation for some of these outcomes are the overly complex visual stimuli that result in voluntarily triggered eye movements, which are highly dependent on a user's current cognitive state.

In this chapter, we show how the reflexive physiological behavior of human eyes can be used to build fast and reliable biometric authentication systems. We utilize the fact that, even though most eye movements are elicited voluntarily, specific *reflexive* movements can be actively triggered using a simple visual stimulus. Measuring and analyzing millisecond-scale characteristics of reflexive eye movements provides several important benefits. Users' eyes naturally and spontaneously react to the shown stimulus so they do not need to follow any instructions or memorize additional information. As a result, elicitation of reflexive behavior requires lower cognitive load and is very fast. This, in turn, enables keeping authentication times short while at the same time extracting large amounts of useful biometric data and achieving low error rates.

Finally, we show another crucial advantage of exploiting reflexive eye movements for authentication: by employing a challenge-response type of protocol, such systems can provide security even under a stronger adversary model than the ones usually considered for biometrics. One of the obstacles for widespread use of biometric authentication in our daily lives is the fact that most biometrics can be captured and replayed relatively easily. Examples include spoofing image recognition systems

with photographs from social media [131] and spoofing fingerprint recognition using photos or traces of fingerprints left on everyday items [129]. If the visual stimulus could be made unique for each authentication attempt, then the elicited responses should accordingly be different, but still include user-specific characteristics. By always choosing a new *challenge* (randomly generated stimulus) and verifying if the *gaze-response* (measured eye movements) corresponds to it, our authentication system can assert that the biometric sample is indeed fresh. Other biometric systems have to make special provisions to achieve a level of spoofing and replay protection. For example, sophisticated fingerprint readers measure additional attributes like temperature and moisture in order to determine liveness. Our gaze-based authentication system achieves these guarantees practically for free, without requiring any other information besides the recording of the user's eye movements.

4.2 Background on Eye Movements

We start by giving a short background of the human visual system and describe the necessary terminology related to eye movements; this allows us to introduce the main concepts that motivate our research and guide the design of the system in the following sections.

Even when one's gaze is firmly fixated on a single stimulus, human eyes are never completely still, as they hey are constantly making hundreds of micro-movements per second. These micromovements are interlaced with about 3-5 larger movements every second, which amount to more than 100,000 eye movements during the course of one day [163]. During standard visual tasks, such as object search or scene perception, our eyes alternate between *fixations* and *saccades*. Fixations are used to maintain the visual focus on a single stimulus, while saccades reorient the eye to focus the gaze on a next desired position. Saccades are rapid eye movements that are considered to be the fastest rotational movement of any external part of our body, reaching angular velocities of up to 900 degrees per second, and usually lasting between 20 ms and 100 ms [164]. In Figure 4.1, fixations can be seen as

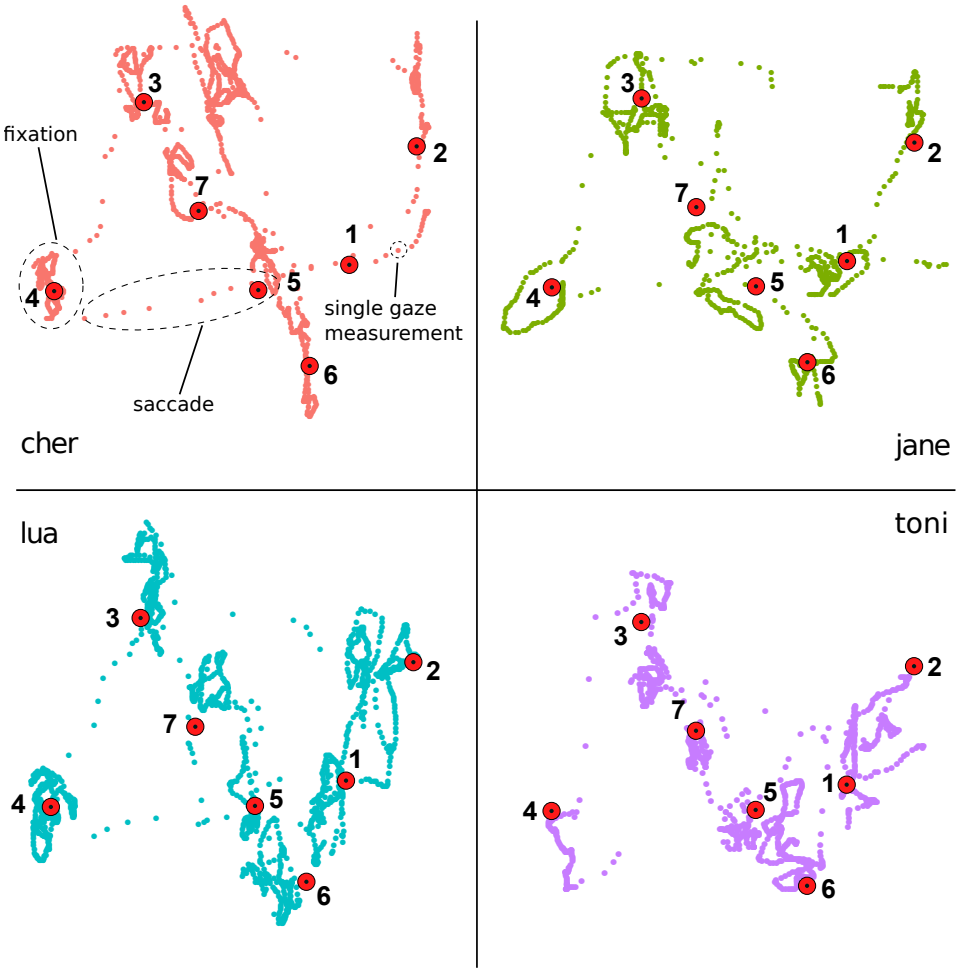


Figure 4.1: Eye movements of four users as the response to the same visual stimulus. Fixations are visible as clustered areas, while saccades consist of series of dots that depict paths. Larger red dots show the positions at which the visual stimulus was shown. Despite their distinct characteristics, all four gaze paths closely match the positions of the stimulus.

areas of large numbers of closely grouped points, while saccades consist of series of more spread recordings that depict fairly straight paths.

Reflexive vs Voluntary Saccades. When a salient change happens in our field of vision, our eyes naturally reorient on the target, since this is a necessary first step to provide information for further higher-level cognitive processes [165]. These externally elicited saccades happen reflexively and are considered to be an effortless neuronal response, requiring very low cognitive load from the user. After the stimulus onset, a corresponding *reflexive* saccade is initiated rapidly, with usual latencies of less than 250 ms [166]. In contrast, voluntary saccadic movements

were shown to have larger mean latencies (above 300 ms) which are additionally influenced by different internal and external factors [166].

The analysis of eye movements has been part of medical research for more than a century since it offers valuable information of our cognitive and visual processing [165], [167], [168]. Keeping the goal of reliable biometric authentication in mind, we are interested in extracting and combining multiple characteristics of human eye movements for which there exists supporting research that they offer stable individual differences between users. For example, Castelhano et al. [169] examine stable individual differences in characteristics of both saccades and fixations and provide support for their stable use in biometric authentication. Saccades were also used in [156] to enable stable authentication and identification. Furthermore, several researchers have analyzed eye behavior features of trained shooters [170], professional baseball players [171] and other specific groups of individuals [172], and reported measurable differences between their eye movements characteristics.

Given that reflexive reactions are less dependent on momentary conscious states of an individual than conscious actions, it is expected that biometrics based on reflexive characteristics offer more stable authentication. Furthermore, taking into account the advantage in faster elicitation times, the goal of our research is to design a stimulus that supports the use of reflexive saccades for biometric authentication. For example, prior research has shown that saccade latencies depend on the dominant eye [173] of the individual, which is a stable characteristic and provides strong motivation for using saccade latencies for classification. Finally, it was shown that *saccade latency* varies if anticipation (temporal expectancy) is present [174]. This provides an argument for randomizing the stimulus that is shown to users.

4.3 Related Work

While different eye tracking methods have been used in medical research for over a century, their use in security research is fairly recent. A review paper by Zhang et. al. [175] provides an overview of authentication methods and systems proposed before 2010, while Saeed [176] gives a more recent comparison of methods and results

of gaze-based authentication systems proposed up to the year 2013. According to Zhang et. al. [175], existing work in user identification and authentication can be roughly divided into two categories: **1)** using gaze tracking as a human-computer interface (control channel) to support various forms of non-biometric user authentication and **2)** using characteristics of the gaze patterns to extract individual biometric traits that enable distinguishing between different users.

In the first line of research, individuals use their eyes to prove their identity by naturally and covertly inputting secret information such as passwords [177], [178], [179] or specific patterns on the screen [131], [180], [181]. Using eyes as a control channel has several advantages, such as prevention of shoulder-surfing and smudge attacks. Unfortunately, these approaches usually share the negative characteristics of passwords, such as requiring the users to learn a procedure or remember and recall different pieces of information, as well as still being susceptible to eavesdropping and replay attacks.

Our work belongs to the second, biometric approach, which uses the characteristics of individual's gaze patterns to discriminate between different users. Such authentication systems usually come with the general benefits, but also challenges typical to biometrics: they usually require no memorization, prevent sharing of credentials, and offer high usability. At the same time, they suffer from irrevocability, which renders replay attacks a serious threat if even a single user's biometric sample is acquired by an attacker.

Biometric approaches to gaze-based authentication can be further divided into two subcategories: those that rely on high-level characteristics of user's gaze patterns (*where* and *what* the user is looking at), and those that analyze the low-level traits of *how* the user's eyes are moving.

High-level characteristics. The first approach is motivated by hypotheses that users exhibit individual behavior during certain tasks. Authors thus extract high-level characteristics of users' responses while they are instructed to freely look at videos, photos of faces, or other specific types of stimuli. Prior work includes

Table 4.1: Comparison to existing biometric authentication systems based on eye-movements

Analysis of	Stimulus	Ref.	Time [s]	EER [%]	Notes
High-level Features					
Scan paths + arch densities	Human faces	[182]	17	25	
Distribution of areas of interest	Human faces	[183]	10	36.1	
Graph matching	Human faces	[184]	4	30	
Fixation density maps	Movie trailer	[185]	60	14	
Low-level Features					
Cepstrum transform of raw signal	Dot, fixed inter-stimulus	[154]	8	N/A	FAR 2%, FRR 22%
Oculomotor plant model	Dot, horizontal sequence	[186]	21	N/A	FAR 5.4%, FRR 56.6%
Scan paths and fixation features	Read section of text	[187]	60	23	
Fixation and saccade features	Read section of text	[188]	60	16.5	
Liveness detection	Dot, horizontal sequence	[189]	100	18	Focus on liveness detection
Fixation and saccade features	**Dot, interactive**	**this thesis**	**5**	**6.3**	**Replay: FAR 0.06%**

analysis of scan paths and arch densities [182], areas of interest on human faces [183], graph matching [184] and fixation density maps [185].

As summarized in Table 4.1, existing work in this category mostly achieves equal error rates higher than 15%, which is likely due to complex features being more dependent on varying cognitive and physiological states of the user. Furthermore, in order to acquire sufficient data to extract complex features, these systems often require long authentication times (measured in tens of seconds!), so further improvements are needed before they can be applied to real-world systems.

Low-level characteristics. On the other hand, motivated by psychological and neurophysiological research [169] that suggests stable differences between users [190], several authors researched systems that use low-level characteristics of users' eye movements as features for discrimination, such as eye movement velocity profiles, sizes of fixation areas, and saccade latencies.

Kasprowski is one of the first authors to start systematically researching the low-level characteristics of user's gaze for authentication. In his initial paper [154] and corresponding Ph.D. thesis [191], he proposes using features such as the distance between the left and right eye-gaze, Fourier and wavelet transforms of the raw gaze signal and average velocity directions. The used stimulus consists of 9 LED lights arranged in a 3x3 grid, where the position of the single active light changes according to a fixed, equally timed sequence, regardless of the user's gaze. An

experimental study showed half total error rates (HTER) of close to 12%, but with relatively high false reject rates of 22%. In relation to our design goals, such stimulus also leads to eliciting some reflexive saccades, but as Table 4.1 shows, it results in longer authentication times and higher error rates. This is likely due to periods of time where the user has already gazed at the light but is still waiting for the position of the active LED to change. The authors also propose, organize and describe two yearly competitions in eye movements verification and identification using their datasets [192], [155], which have further increased the research interest in gaze-based authentication and its fusion with other biometric modalities [193].

Komogortsev et al. propose modeling the physiological properties of individuals' oculomotor plant [186] during multiple horizontal saccades and using the estimated model parameters as features for classification. Related work by Holland et al. [187] provides an insight into performances of multiple features such as fixation counts and durations during text reading, combining these two approaches to achieve an EER of 23%. Furthermore, their newer research [188] provides an additional analysis of 13 classification features based on fixations and saccades and achieves an EER of 16.5%.

Finally, while most previous research on eye-movement biometrics has been performed using statically positioned devices, head-mounted trackers are starting to achieve similar results. As an example, recent work by Zhang et al. [194] analyzed the application of eye tracking to authenticate users of VR headsets, used a dot-based stimuli, and reported an EER of 6.9% for their prototype headset.

Continuous Authentication. In contrast to point-of-entry authentication, in which the classifier makes a single decision about the user's identity at the beginning of the session, Eberz et al. [156] propose using 21 low-level characteristics of eye movements to continuously re-authenticate users, regardless of their current task, and thus detect intruders whose eye movements differ from the legitimate user over a period of time. For one parameter combination, the authors achieve equal error rates of 7.8% when 40 seconds are chosen as a period before making the first decision. Furthermore, using one-class SVM classification, they are able to detect all but 1% of attackers when the classifier is allowed to make decisions across

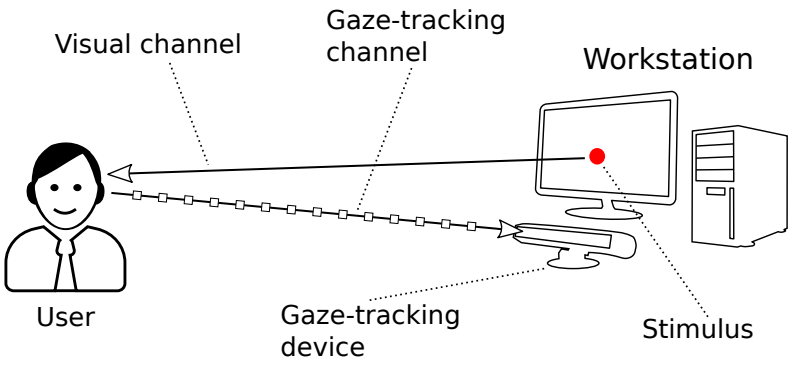


Figure 4.2: System model. The workstation uses data acquired by the gaze tracker and user's biometric template to make the authentication decision. The adversary has read-write access to the gaze channel. The visual channel is authenticated and therefore read-only.

the span of 30 s [195]. However, due to the requirement of task independence in the continuous authentication scenario, potential replay attacks remain a serious vulnerability. If the attacker is able to capture even a short recording of legitimate user's gaze, he can continuously rewind and replay it back to the gaze tracking device. This causes the system to (correctly!) accept the received eye movements as coming from a legitimate user.

4.4 Assumptions and Goals

We start by defining the system and adversary model used throughout this chapter; we then state the design goals for the visual stimulus and the authentication system.

4.4.1 System Model

We assume the general settings of the user authenticating to their device (workstation) throughout the course of a normal workday. A simple visualization of the system model is shown in Figure 4.2. The user authenticates to the workstation using a gaze tracking device by looking at a visual stimulus displayed on the screen. The workstation uses the data acquired by the gaze tracker and the user's biometric template to make the authentication decision.

The legitimate user is the one who is enrolled in the authentication system. The enrollment happens in a secure setting, where the legitimate user authenticates to

the workstation using another authentication method. During enrollment, the user is shown several visual stimuli and the workstation uses the corresponding recordings of the user's gaze to create a biometric template used for identity verification.

The interaction takes place through three different channels. The *visual channel* is an authenticated channel from the workstation to the user that consists of a screen that displays information, and the *gaze tracking channel* from the user to the gaze tracker allows the workstation to determine characteristics about the user's eyes, including where he is looking on the screen, as well as capture the reflexive eye movements described in Section 4.2.

The workstation itself cannot be modified or forced to run unintended code.

4.4.2 Adversary Model

The adversary's goal is to impersonate a legitimate user and successfully authenticate to the workstation. The adversary can freely choose his victim from the set of enrolled users. Since he can observe both the visual and gaze channels, the adversary has access to the biometric data from previous authentication attempts by the victim. Following the discussion in Chapter 2, we focus on two different types of attacks that the adversary can perform:

- **Impersonation attack.** The adversary tries to gain access to the workstation by positioning himself in front of the gaze tracking device. This is the most common way of evaluating biometric authentication systems, and is usually reported in terms of false reject (FRR) and false accept rates (FAR) as well as equal error rates (EER).

- **Replay attack.** The adversary targets a specific user and replays his previously recorded authentication attempt to the authentication system. This can be done either at the sensor level (e.g. by using a mechanical eye replica) or by bypassing the gaze tracking sensor completely and injecting the recorded samples between the workstation and the sensor.

Biometrics are non-revocable, and users are surrounded by sensors that can be used to capture one's biometric data. We therefore believe that modeling an attacker as having access to legitimate user's previous biometric measurements is a realistic and necessary assumption.

Most static biometrics, such as fingerprints or face recognition [131], cannot provide security under such assumptions; the ability to prevent replay attacks is one of the major strengths of our scheme since simply replaying an acquired sample is arguably the most accessible attack vector for most biometrics.

We do not consider a targeted adversary who is able to model and generate arbitrary artificial samples of a user's eye movements in an interactive manner. As we further discuss in Section 4.10, such attacks require significantly higher levels of complexity and effort from the adversary: a level of commitment against which most biometric systems cannot provide security guarantees.

4.4.3 Design Goals

Based on the discussion in Chapter 2 and the previous sections, we state the design goals for the biometric user authentication system as:

- **Low cognitive load:** The system should pose low cognitive load on the users. Ideally, users should not be required to remember credentials, carry tokens, or learn new procedures. Moreover, the cooperation required from the user should be as effortless as possible.

- **Authentication speed:** The duration of a single authentication attempt should be as short as possible.

- **Resistance against replay:** The system should make it difficult for an adversary to replay acquired biometric samples and thereby successfully authenticate.

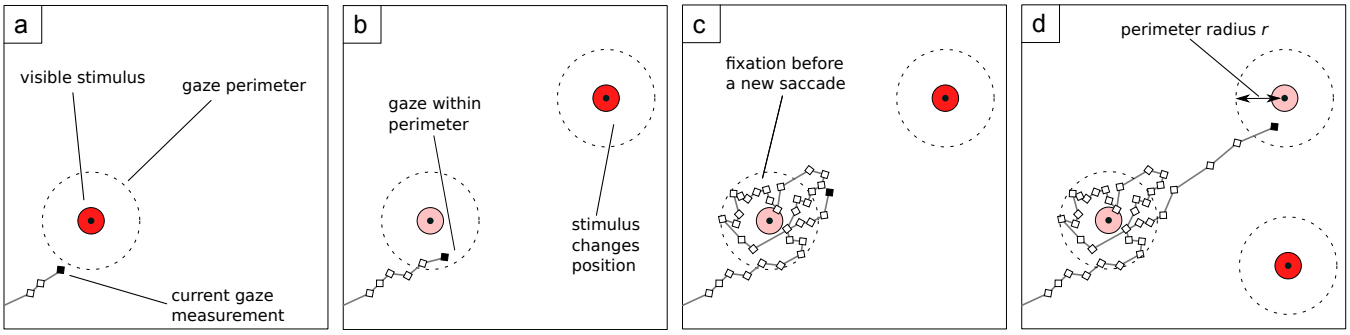


Figure 4.3: Visualization of the stimulus for reflexive saccade elicitation. At any given time, only a single red dot is shown; previous positions of the dot are shown in this figure only to help the reader. Shortly after the red dot appears on the screen (a), a user's visual system starts a reflexive saccade to shift the gaze (dotted path) towards its position. Several milliseconds later, as the user's gaze enters the invisible perimeter around the stimulus (dashed circles), the system detects that the dot was successfully gazed at and momentarily changes its position. Before the new saccade starts, there is usually a 100-250 ms long fixation period, during which the visual system processes new input information (saccade latency). In (d), the presented dot is again successfully gazed, and its position is again changed.

4.5 System Architecture

The proposed authentication system works as follows. The workstation shows an interactive visual stimulus on the screen (we refer to it as *gaze-challenge*). Simultaneously, the gaze tracking device captures eye movements of the user as he watches the screen (*gaze-response*), which the workstation uses to adapt the stimulus in real time. Finally, the workstation makes a decision about the user's identity and verifies if the received gaze-response corresponds to the shown gaze-challenge, asserting that the captured eye movements are indeed fresh.

4.5.1 Stimulus for Reflexive Saccade Elicitation

To achieve the stated design goals, a visual stimulus should satisfy several requirements. It should elicit responses that are sufficiently distinctive to allow discrimination between different users. The response should not require high cognitive effort and should not depend on the user's momentary cognitive state. The stimulus should be *unpredictable* to prevent habituation: seeing an image for the first time will likely result in a different response than seeing it for the second

and each consecutive time [174]. Finally, in order to allow fast authentication, the stimulus duration should be as short as possible.

Design. Considering that the reflexive behavior is more stable and less dependent on the user's transient internal cognitive state than the voluntary behavior, our goal is to design a stimulus which allows eliciting and measuring individual traits of the user's reflexive saccadic responses. Reflexive saccades are triggered by salient objects that appear in one's field of view; thus our stimulus consists of presenting a single red dot on a dark screen and changing its position multiple times. Consequently, as shown in Figure 4.3, a user's eyes respond to the salient change by eliciting a reflexive saccade that reorients the gaze towards the dot. Every time the position of the dot changes, the visual system responds by initiating a new reflexive saccade. Due to the saccade latency, this happens after a period of 100-200 ms during which the visual system processes new information.

Ideally, our stimulus should elicit the maximal number of reflexive saccades in a given period of time, and this highly depends on the frequency with which the position of the dot changes. If this frequency is too high, the user's eyes will not be given sufficient time to perform a full saccade. If it is too low, the user might get tired of looking at a static screen and start voluntary saccadic movements. Furthermore, each user is slightly different, so there might not exist a unique frequency at all. Using an interactive stimulus achieves the optimum between these trade-offs by interactively changing the location of the dot as soon as the user *successfully gazes* the dot, i.e., when a user's gaze enters a perimeter of radius r around the dot's center. This results in eliciting the maximal number of full saccades in any given time interval and ensures that the user's visual system receives an outside stimulus change as often as possible, consequently reducing the elicitation of voluntary saccades, which depend on his current cognitive state. To ensure that the stimulus terminates even if the user is not looking at the screen, the dot is considered to be *unsuccessfully gazed* and moves to the next position after a specific period of $D_{\max}$ milliseconds has passed. This process continues for the N arbitrarily chosen stimulus positions that constitute a gaze-challenge.

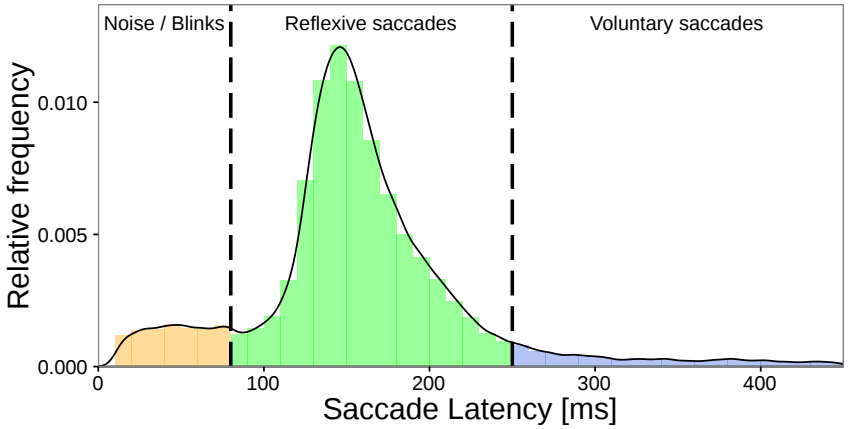


Figure 4.4: Relative frequency of the saccade latencies for gaze-responses in our dataset. Latencies are computed as the duration between the stimulus change and the start of the subsequent saccadic movement. Vertical lines discriminate between reflexive and other types of saccades; the latencies of reflexive saccades are usually lower than 250 ms, in contrast to the latencies of voluntary saccades that are over 250 ms. The values under 80 ms are likely the result of the noise or blinking, or voluntary saccades initiated well before the stimulus change [166].

Basing an authentication system on reflexive movements provides additional benefits: taking into account that reflexive behavior is significantly harder to consciously control, an adversary attempting an impersonation attack is less likely to be able to successfully imitate another user's characteristics. Most importantly, because of the natural and effortless tendency of the human visual system to keep *"catching"* the red dot, the response to such visual stimulus is fully reflexive: users neither need to follow specific instructions nor invest high cognitive effort —*their eyes do the work themselves.*

Effectiveness of the stimulus. In order to evaluate how effectively does our designed stimulus elicit reflexive behavior, we compute saccade latencies for a total of 991 legitimate authentication attempts from the experimental dataset (described in detail in Section 4.7) used throughout this thesis. Since each of the measurements represents a gaze-response to a stimulus with 25 different positions for the dot, in total, this sums up to analyzing close to 25,000 captured saccades.

Figure 4.4 shows the distribution and categorization of the measured saccade latencies, dividing them into reflexive saccades, voluntary saccades and saccadic movement caused by blinks. Given that latencies under 80 ms have only been

recorded in specifically designed conditions, e.g., when the stimulus position and onset are predictable [174], in our setting we consider them to likely be the result of blinks or noise [166]. Remaining latencies predominantly fall below 250 ms, the threshold that characterizes reflexive saccades [166]. This lets us conclude that the stimulus does indeed elicit primarily reflexive behavior.

4.5.2 Authentication Protocol

We now use the proposed stimulus as the building block to design a challenge-response protocol for biometric user authentication that is secure against replay attacks.

At the end of the protocol execution, the workstation must decide if the user whose identity is claimed is at the moment truly present in front of the gaze tracking device. To that goal, the workstation must ensure that two properties hold:

Freshness. Freshness of the received biometric data can be ensured by always showing a different randomly generated visual stimulus (gaze-challenge) to which every response will differ in a verifiable way.

Correct Identity. This property holds if the authenticating user can generate biometric data that corresponds to the claimed user's template, which was created during enrollment.

The protocol for local biometric authentication is shown in Figure 4.5. After the user claims his identity, the workstation generates a fresh visual stimulus, which we refer to as *gaze-challenge* (c_W) in the rest of this chapter. c_W consists of a set of n randomly chosen coordinates, which uniquely define the interactive stimulus described in Section 4.5.1. As the gaze-challenge is presented to the user, his eyes reflexively respond with a series of eye movements, which constitute the *gaze-response* (r_U). Gaze-response is recorded by the gaze tracking device through the gaze channel.

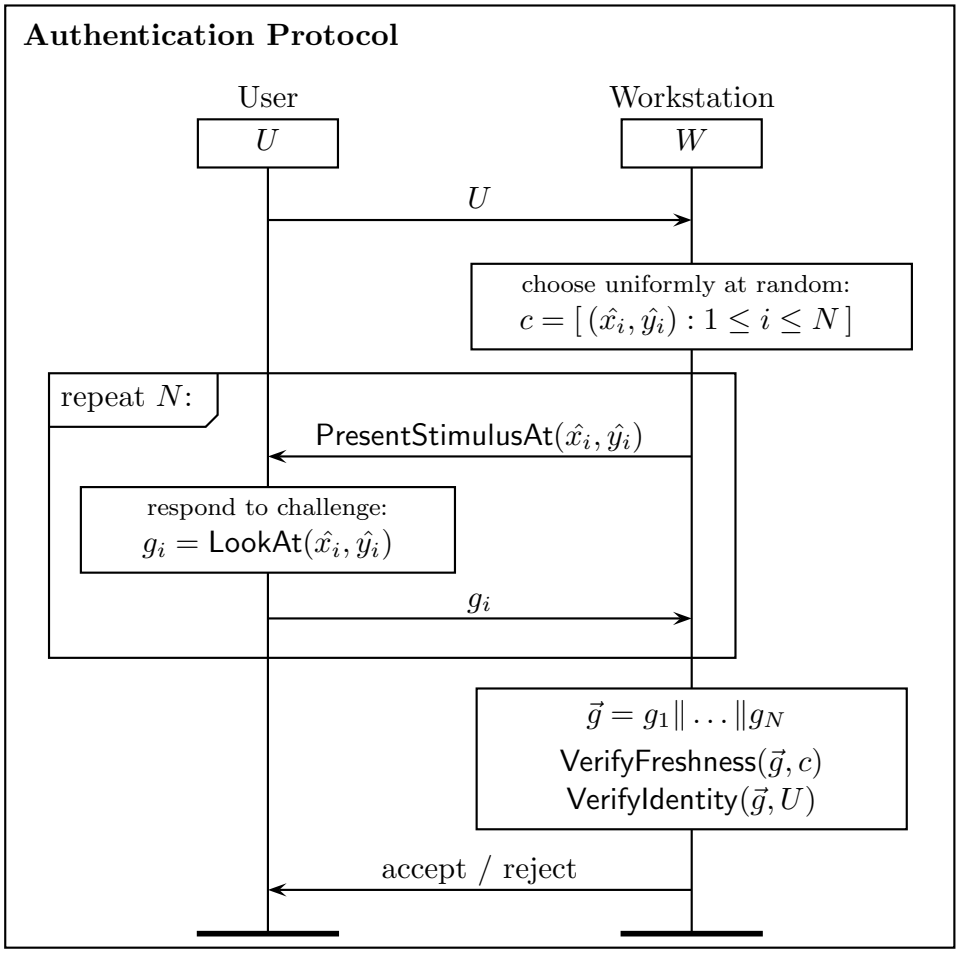


Figure 4.5: Biometric challenge-response authentication protocol. The user first claims his identity, after which the workstation generates a fresh *gaze-challenge* c, an ordered list of positions in which the stimulus is shown. The user then looks at (LookAt) a screen where the stimulus is shown at N positions $\{(\hat{x}_i, \hat{y}_i)\}$. Meanwhile, the gaze tracking device records the user's gaze paths g_i for all stimulus positions that constitute the *gaze-response* $\vec{g}$. The workstation verifies the freshness of $\vec{g}$, and finally verifies that the biometric features extracted from $\vec{g}$ correspond to the claimed identity.

In order to accept or reject the user's authentication request, the workstation performs two verification steps: VerifyFreshness and VerifyIdentity. These are described in detail in Sections 4.5.3 and 4.5.4, respectively.

In the final message, the workstation notifies the user if he has been granted or denied access to the system.

4.5.3 VerifyFreshness

As described in Section 4.5.1, each visual stimulus is uniquely defined by a list of N coordinates; therefore, it is possible to always present a different random gaze-challenge to the user. Since no visual stimulus shown to users is ever reused, in order to verify the freshness of the response, it suffices to verify if the received gaze-response closely corresponds to the freshly presented gaze-challenge. As visualized in Figure 4.3, if some gaze-response was recorded while specific gaze-challenge was shown to the user, then the user's eye movements should closely resemble the order and positions in which the stimulus dot was shown. This is visible in Figure 4.1: despite differences in gaze patterns of different users, all of them correspond to the locations of the stimulus dot.

The system determines if the gaze-response is indeed fresh by ensuring that the user timely gazed at the majority of the stimulus positions. After a stimulus dot is shown in one of the N positions, it is considered *successfully gazed* only if one of the subsequent measurements of the user's gaze position falls within a radius of R pixels from the center of the stimulus dot. Otherwise, if no gaze measurement falls within its radius after $D_{\max}$ milliseconds, a position is considered to be *unsuccessfully gazed* and the dot moves to the next position:

$$g_i \coloneqq [(x_j, y_j) : t_i \leq t_j < t_i + D_{\max}]$$

$$\text{succ. gazed}\,(\hat{x}_i, \hat{y}_i) \iff \exists (x, y) \in g_i \colon \; \|(x, y) - (\hat{x}_i, \hat{y}_i)\|_2 \leq R$$

In order to decide on the freshness of the received gaze-response, the system checks if the ratio of successfully gazed stimulus positions is greater or equal to a chosen percentage threshold T.

As the threshold T increases, the possibility that an adversary successfully replays an old recording of a legitimate user's gaze decreases. On the other hand, this also results in more legitimate attempts failing freshness verification, e.g., because of inaccurate gaze measurements. We evaluate the security guarantees of different thresholds T in Section 4.8.3 and analyze the impact of different

values for the threshold D on the classification performance and authentication times in Section 4.9.3.

4.5.4 VerifyIdentity

If the received gaze-response passes the freshness verification, the system finally validates that it truly originated from the user whose identity was claimed at the beginning of the protocol. The received gaze-response is first used as input to compute a set of specific feature values that are idiosyncratic and support stable classification between users. Next, the computed features are used as an input to a two-class classifier that was created during user enrollment. The classifier determines whether the calculated features more likely belong to the user whose identity was claimed, or to some internal or external attacker. As the last step, the authentication system makes a final decision and notifies the user of acceptance or rejection.

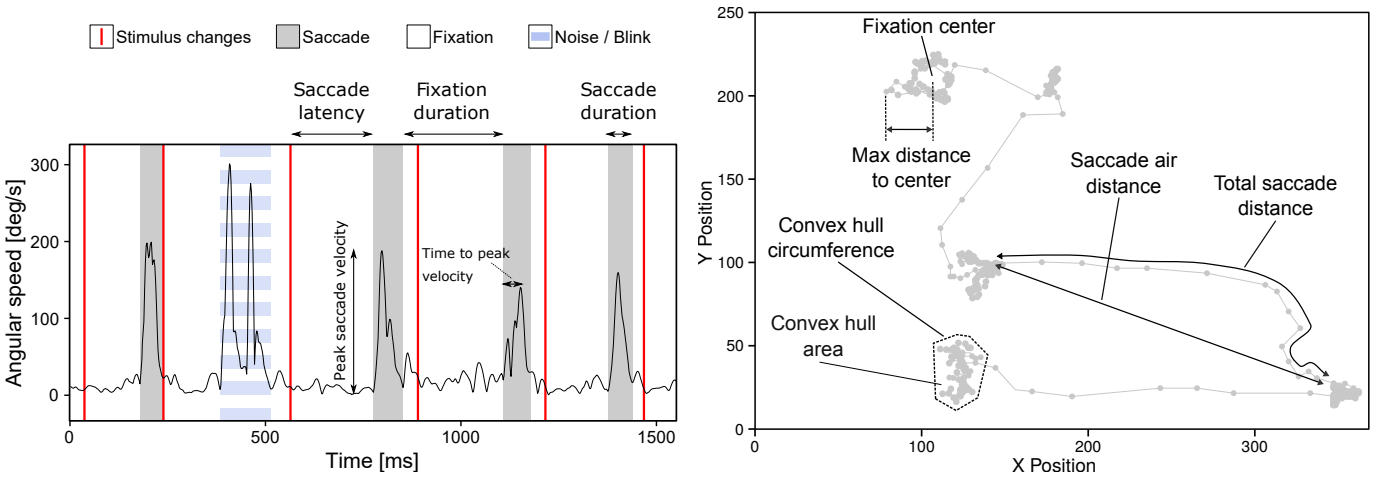
The next section describes the details about the features that we use and how we train the user classifiers.

4.6 Features for Gaze Classification

This section describes the process of extracting individual characteristics from user's gaze-response and training a classifier that can uniquely discriminate between future responses of the same user and any other user's gaze patterns.

4.6.1 Feature Extraction

Feature extraction is the process of converting the raw measurements into a lower dimensional set of meaningful data that retains most of the information useful to distinguish between the different output classes. When considered in the context of eye tracking, the feature extraction process should take as input the time-stamped positions of one's gaze during the authentication attempt, and compute a significantly lower dimensional set of feature values that allow discrimination between different individuals.



(a) Temporal Features **(b)** Spatial Features

Figure 4.6: Visualization of the features on (a) the temporal and (b) the spatial plots of the raw gaze tracking data. In Subfigure (a), the moment when the stimulus changes position is depicted with a vertical red line. The period around 500 ms, highlighted with the horizontal stripes, is physiologically impossible for a human eye to perform and is caused by blinking. We remove such artifacts with methods described in Section 4.6.

Saccade and Fixation Detection. Following the discussion in Section 4.5.1, the expected user's gaze will consist of multiple repetitions of a reflexive saccade, which redirects one's gaze at the new position of the red stimulus dot, followed by a fixation that lasts until one's visual system detects the change in stimuli location (saccade latency). Consequently, the first step in the feature extraction process is to split the raw gaze measurement into intervals of saccades and fixations, which we later use to compute specific characteristics of one's eye movements.

We implement an adaptive algorithm [196] that estimates the level of noise in the data to determine the thresholds used to classify the measurements into periods of fixations and saccades. The detection is mainly based on angular velocities and accelerations, taking into account the known physiological limitations of eye movements. As seen in Figure 4.6a, the algorithm also detects eye movement recordings that could not have been generated by a human eye under known physiological constraints, and are usually the result of blinking. Given that the mean duration of a single blink is close to 200 ms [164], and that head movements and gazes outside of the screen area usually last even longer, it is important to denoise

Table 4.2: Relative Mutual Information ($RMI_{\mathbf{ID}}$) of the features that were considered during system design. Rows in bold numbers show the 16 features that were selected for subsequent classification. Selection was made based on RMI values, except where stated differently. #1, #2, #3, #20, and #33 are included to allow comparison.

#	Feature Description	$\mathbf{RMI_{ID}}$	Comment
1	Mean Y coord. of the corneal reflex position (left eye)	0.3267	Excluded as a static feature
2	Mean the pupil diameter (left eye)	0.2871	Excluded as a static feature
3	Mean X coord. of the corneal reflex position (left eye)	0.2381	Excluded as a static feature
4	**Median air distance vs total distance ratio**	**0.1846**	
5	**Median fixation duration**	**0.1575**	
6	**Median average fixation velocity**	**0.1540**	
7	**Density of fixation convex hulls**	**0.1474**	
8	Mean fixation duration	0.1456	Excluded due to similarity with #5
9	**Median saccade latency**	**0.1453**	
10	Total time of authentication attempt	0.1447	Excluded as similar to 11
11	**Average time per stimulus**	**0.1447**	
12	Mean saccade duration	0.1403	Excluded due to similarity with #15
13	**Mean velocity**	**0.1397**	
14	**Average distance per stimulus**	**0.1382**	
15	**Median saccade duration**	**0.1363**	
16	**Median fixation convex hull area**	**0.1362**	
17	**Median saccade average velocity**	**0.1360**	
18	**Median fixation convex hull perimeter**	**0.1343**	
19	**Median fixation max velocity**	**0.1337**	
20	Ratio of successful gazes	0.1321	Excluded as a static feature
21	**Median saccade max velocity**	**0.1283**	
22	**Median saccade max acceleration**	**0.1257**	
23	Mean saccade latency	0.1227	Excluded due to similarity with #7
24	**Median fixation max distance**	**0.1226**	
25	Median saccade air distance	0.1215	
26	Median fixation Y span	0.1208	
27	Median fixation X span	0.1089	
28	Median fixation X and Y span ratios	0.0983	
29	Median saccade X span	0.0937	
30	Median saccade X and Y span ratios	0.0864	
31	Median saccade Y span	0.0790	
32	Median saccade time to max velocity	0.0698	
33	Random variable	0.0567	Included only for comparison

the raw data before further analysis. These artifacts are filtered based on research that shows the peak angular velocity of the human eye to lie between 700 and 900 deg/sec [164], and the peak angular acceleration to not cross 100000 deg/sec^2.

Having grouped the measurements as belonging either to a fixation or a saccade, we proceed to calculate a set of feature values for each recorded gaze sample, ignoring those measurements that are classified as noise by the procedure. The set of features that we use in this work is motivated and extended from a subset of features used in previous work on eye movement biometrics [156, 195].

Using Median Values. Since the authentication system always shows a fresh visual stimulus (defined by the N positions of the stimulus dot), the computed features should not be influenced by the positions of the dots in the gaze-challenge,

as this would lower the probability that the user reauthenticates with a fresh challenge. As Figure 4.6 shows, each gaze-response consists of intermixed periods of saccades and fixations and each such period allows us to compute multiple features. However, we are interested in computing a single set of identifiable feature values for a given gaze-response as a whole, irrespective of the number of elicited saccades and fixations; to that end, and to reduce the effect of noise, feature values for a single user's gaze-response (authentication attempt) are computed as the median of feature values computed on individual saccades or fixations in that gaze measurement.

4.6.2 Feature Quality

In order to support secure and reliable authentication, the features should ideally be chosen so that they are as varied for different users and as similar as possible when computed for multiple authentication attempts of the same user. Extracting stable and distinctive features from real-world user behavior data, especially when the measurement is noisy, as is the case with eye trackers, is a challenging task.

Since all potential features do not contribute the same amount of distinguishing power, we follow a semi-automated approach to select the optimal set of features for the authentication system. Initially, we explore a broader set of fixation and saccade traits, in addition to a range of other metrics that measure overall characteristics of the gaze path.

Relative Mutual Information. In order to choose the final subset of features that we use for classification, we compute the Relative Mutual Information (RMI), a measure that quantifies the reduction in the entropy of the final outcome (user's identity in the context of biometric authentication) as a result of knowing the value of an individual feature [197]. More precisely, RMI can be expressed as:

$$\mathrm{RMI}_{\mathrm{ID}}(F) \coloneqq \frac{\mathrm{MI}(\mathrm{ID}, F)}{H(\mathrm{ID})} = \frac{H(\mathrm{ID}) - H(\mathrm{ID}|F)}{H(\mathrm{ID})}$$

Mutual Information between two variables A and B is denoted as $\mathrm{MI}(A, B)$, while $H(A)$ represents the entropy of variable A.

Based on RMI, we test the features on randomly chosen subsets of the dataset, measure their classification performance, and exclude those that do not achieve satisfactory results. The chosen features that have a clear spatial or temporal representation are shown in Figure 4.6, while their RMI values can be found in Table 4.2.

Chosen Features. As the RMI values in Table 4.2 show, medians of `average angular speeds` during fixations or saccades, as well as the `duration` of fixations are among the most specific features we tested. This finding is congruent with the feature assessment conducted by Eberz et al. [156, 195], where pairwise speeds exhibit the highest relative mutual information, only outperformed by some of their static features, such as pupil diameter. Contrary to their results, we identify `saccade curviness` (ratio of air distance and total distance of a saccade) and `saccade latency` to be the features that yield the most distinguishing power. Furthermore, we identify several discriminative features based on computing a convex hull of all measurements in a fixation: `convex hull and circumference`, as well as `fixation density`, defined as the ratio of the convex hull area and the number of gaze measurements in that fixation.

Using Dynamic Features. Given the focus on evaluating the feasibility of using reflexive behavior for authentication, this thesis only uses dynamic characteristics of eye movements for classification. We thus consciously forego using several features that most gaze tracking devices provide, such as an estimate of user's pupil size and the distances between the user's eyes. In prior work, pupil size was shown to be one of the more discriminative features for gaze-based authentication systems [195], however, the authors raise valid concerns that an adversary could manipulate his pupil size, e.g., by controlling the lighting conditions. Despite potential classification improvements, in our research we employ only features that can be extracted from raw coordinates of the user's gaze. We further discuss relaxing this assumption in Section 4.10.

4.6.3 User Enrollment

During enrollment, several gaze-responses are used to train a dedicated 2-class classifier that the system will use as user's identity verifier. In any subsequent authentication attempt, the same set of feature values are extracted from any gaze-response and the classifier makes a decision whether the values correspond to the claimed user or not.

Besides legitimate user's gaze-responses, the enrollment procedure requires a similarly sized set of gaze-responses belonging to other users that are labeled as negative samples during classifier training.

Choice of the Classifier. In this analysis, we mainly use a Support Vector Machine (SVM) [198] with Radial Basis Function (RBF) kernel as the classifier, since SVMs are known to provide strong classification results for non-linear data sets. In Section 4.9.4 we also evaluate and discuss several other classification algorithms for multiple test configurations, confirming that SVMs consistently achieve the lowest error rates on our data.

SVMs with RBF kernels are fully defined by two hyper-parameters: **1)** C, which controls the trade-off between the penalty of incorrect classification and the margin of the decision hyperplane, and **2)** σ, which is a parameter that defines the scale of the radial basis function. The optimal pair of hyper-parameter values is chosen from a predetermined set of potential values, based on the evaluation that uses 5-fold cross-validation: for each pair of potential hyperparameters, 80% of the enrollment data is used to train the resulting classifier, while the remaining 20% of the enrollment data is used to evaluate the classification performance; this is repeated five times.

The pair of hyperparameters that results in strongest classification performance is finally used to derive the final user classifier which is used in future authentication.

4.7 Data Acquisition

In order to experimentally evaluate the performance of the proposed system and protocol, we developed a prototype and ran a series of user experiments to gather

data for analysis.

4.7.1 System Prototype

Setup. Our prototype setup is composed of a gaze tracking device (SMI RED 500 [199]), a 24-inch LED screen, and a desktop computer. The generation of the visual stimulus and the gaze sampling was performed by a custom-built software library that controls the gaze tracking device. We implemented procedures that take care of the internal calibration of the gaze tracker, the validation of the calibration accuracy, and the visual presentation of the stimulus, as well as the acquisition of the gaze samples captured by the gaze tracker.

Parameters. For each authentication attempt, the prototype generated a visual challenge consisting of $N = 25$ random positions on which the stimulus will be shown. The distance between users' eyes and the gaze tracking device (positioned directly underneath the screen) was approximately 70 cm. Red stimulus dot is shown on a plain dark background, with a diameter of 0.7 cm (0.95°). In order to detect that a dot was successfully gazed, we used a perimeter radius of $r = 1.4$ cm (1.25°). If not successfully gazed, the dot changed position after $D_{\max} = 1000$ ms.

4.7.2 User Experiments

Experiment Design. For the purpose of assessing feasibility and performance of the proposed system, we conducted a series of user experiments that reflect the scenario described in Section 4.4. We refer to a series of consecutive authentication attempts with the same participant as one session. Each session lasted about 10 minutes and included a briefing and 15 authentication attempts. Before participant's first session we generated a calibration profile that was reused during all subsequent sessions with that participant. To analyze the performance of our system, both from the perspective of a user and an attacker, we divided the participants into two groups: legitimate users who have completed the enrollment procedure, and external attackers, whose gaze characteristics were not known to the system.

In order to show that our system can successfully authenticate users over the course of a normal workday (without re-calibration), we require each enrolled user to take part in a minimum of three (up to four) sessions. The first two sessions are five minutes apart and mimic a legitimate user leaving his desk to take a break or use the restroom. All subsequent sessions are at least 6 hours apart. Participants acting as external attackers are only invited to one session where they are asked to impersonate a legitimate user, i.e., the system uses the calibration profile and biometric template of the chosen legitimate user. Every external attacker tries to authenticate as 5 different legitimate users, at least 3 times per user. In their last session, legitimate users were asked to act as internal attackers and each performed a minimum of 15 attempts of impersonating other users, analogously to external attackers.

Test Population. Experimental data was acquired from a total of 30 participants aged 21 to 58 who were recruited from the general public through public advertisements, email lists, and social media. The only requirement was a minimum age of 18. The test population consists of 7 women and 23 men. Out of the 30 recruited participants, 22 participants were enrolled as legitimate users and 8 participants represented external attackers whose gaze characteristics were not known to the system. The acquired data set consists of a total of 1602 gaze-responses: 1021 authentication attempts by legitimate users and 581 simulated attack attempts by either internal or external attackers.

Participants were told that their eye movements will be recorded for the purpose of evaluating the feasibility of distinguishing individuals based on their behavioral gaze-based biometrics. They were also told that they can decide to withdraw their participation in the experiment at any time. Given that biometric data was captured in the experiment, participants also signed a written consent form in accordance with the approval given by the University of Oxford's research ethics committee, reference number SSD/CUREC1A/14-226. Names have been replaced with pseudonyms.

Participants who did not have normal vision wore contact lenses or were asked to remove their glasses. This was done to remove the possibility that classification relies

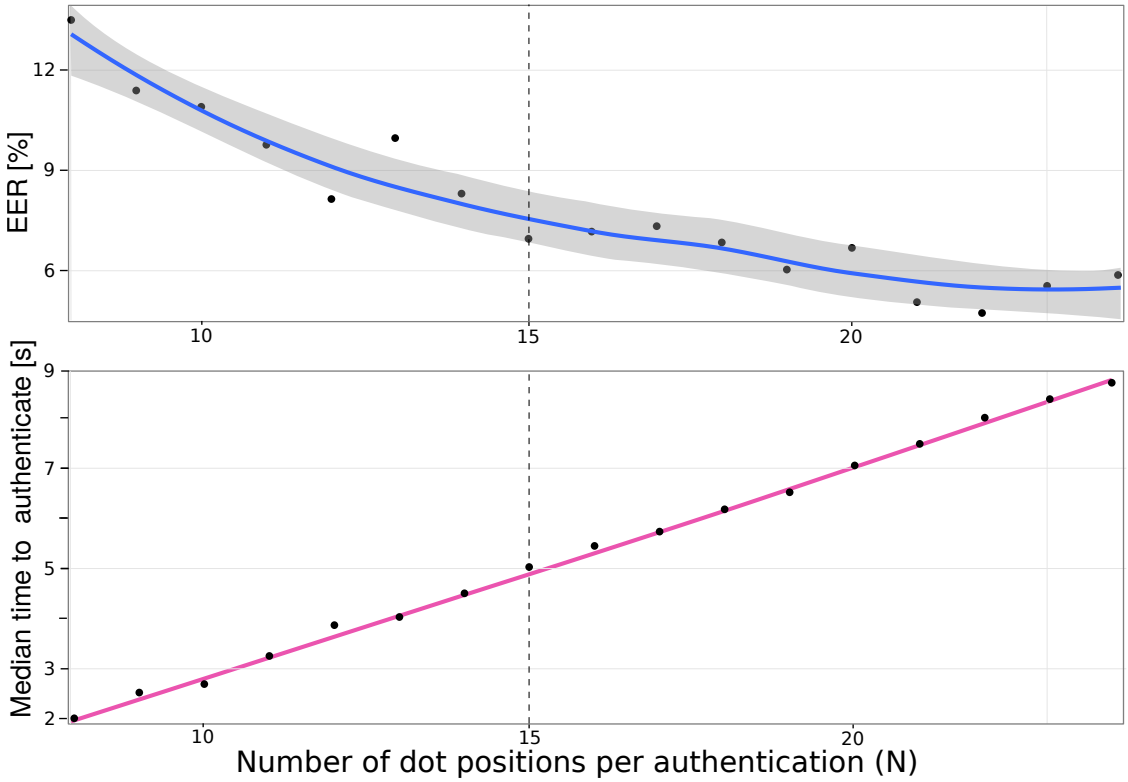


Figure 4.7: Measured authentication time and EER as a function of gaze-challenge complexity N. As N increases from 8 to 24, the EER reduces from above 12% to under 6%, while at the same time, the median time to authenticate grows linearly from 2 seconds to about 9 seconds. The vertical line depicts a scenario where 15 positions are used in a challenge: the median authentication time is around 5 seconds, while the EER is close to 7%.

on potential specific characteristics of recorded gaze when glasses are worn. For the same reason, lighting conditions were not changed during all experiment sessions.

4.8 System Evaluation

We now experimentally evaluate the proposed system with respect to the design goals stated in Section 4.4.

4.8.1 Varying the Challenge Complexity N

One of the defining parameters of the proposed system is N, the number of stimulus positions in a single gaze-challenge. We first analyze the effect that varying N has on authentication time and overall user classification performance. Incrementing N directly increases the complexity of gaze-challenge, thus requiring more time

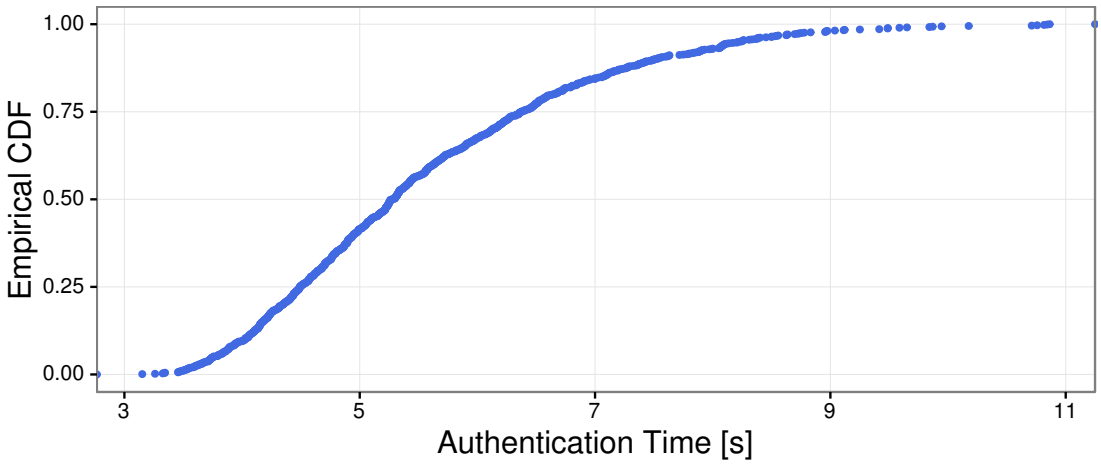


Figure 4.8: Empirical cumulative distribution function for the duration of all measured authentication attempts when $N = 15$. Close to 50% of the attempts took less than 5 seconds, while more than 80% of the attempts lasted less than 7.5 seconds.

to respond to the visual stimulus. At the same time, larger N should allow the system to extract more stable features and thus achieve stronger classification results. On the other hand, as N decreases, both the authentication time and the classification performance are likely to decline.

Setup. Since all user experiments were run with gaze-challenges that had $N = 25$ stimulus dot positions, we can evaluate the classifier performance in a scenario where gaze-challenges consist of $K < N$ positions by simulating that the stimulus presentation and gaze recording stopped after the K-th position was gazed. Such an adapted dataset is constructed by only considering gaze measurements that were recorded before the $(K + 1)$-th stimulus position is shown.

The classification performance for each K and for each user is estimated by computing an equal error rate (EER) while performing a five-fold cross-validation of the individual classifiers as follows. In each of five repetitions, four out of five folds of the legitimate user's authentication attempts are provided as enrollment data for user enrollment that was performed as described in Section 4.6. The remaining fold was used to evaluate classifier performance against other users' authentication attempts as negative samples. The resulting EER for any K is computed as an average across all five folds of all individual users' classifiers for that K.

Results. We show the effect of varying N on authentication time and classification performance in Figure 4.7. The median time for a single authentication attempt grows linearly from 2 seconds for 8 stimulus positions, to about 9 seconds for 24 stimulus positions. At the same time, the overall EER of the classification falls from around 12% when only 8 stimulus positions are used, to a level of 6% when 24 stimulus positions are used in a challenge.

Since $N = 15$ shows a balanced trade-off between classification performance and median authentication time, we use this value to report results in the remainder of the analysis. In order to provide a more comprehensive estimate of the time required for the majority of users to authenticate than just median, in Figure 4.8 we show a cumulative density function of the authentication times for all users when $N = 15$. The figure shows that half of the users authenticate in 5 seconds or less, while the authentication for more than 80% of the users takes less than 7.5 seconds. As we discuss in Section 4.10, these times are favorable to previous related work in gaze-based authentication.

4.8.2 Impersonation Attacks

Recall that, in an impersonation attack, the attacker targets a specific user with the goal of responding to the gaze-challenge posed by the system, and successfully impersonating the legitimate user in order to gain access. The attacker is permitted to use the gaze-based authentication system in any way he wishes, such as purposely moving or altering the angle of his head to try to increase the chance of gaining access.

As described in Section 4.7.2, we purposely design the user experiments to simulate this type of attack as closely as possible: all participants were asked to perform multiple "attack attempts", in which they falsely claimed some other user's identity and tried to authenticate with the gaze calibration profile of the legitimate user loaded by the system.

Setup. For each user, we perform a five-fold cross-validation to estimate the performance of the system under such attacks. We enroll the user as described in Section 4.6, using four out of five folds of legitimate user's samples, and then

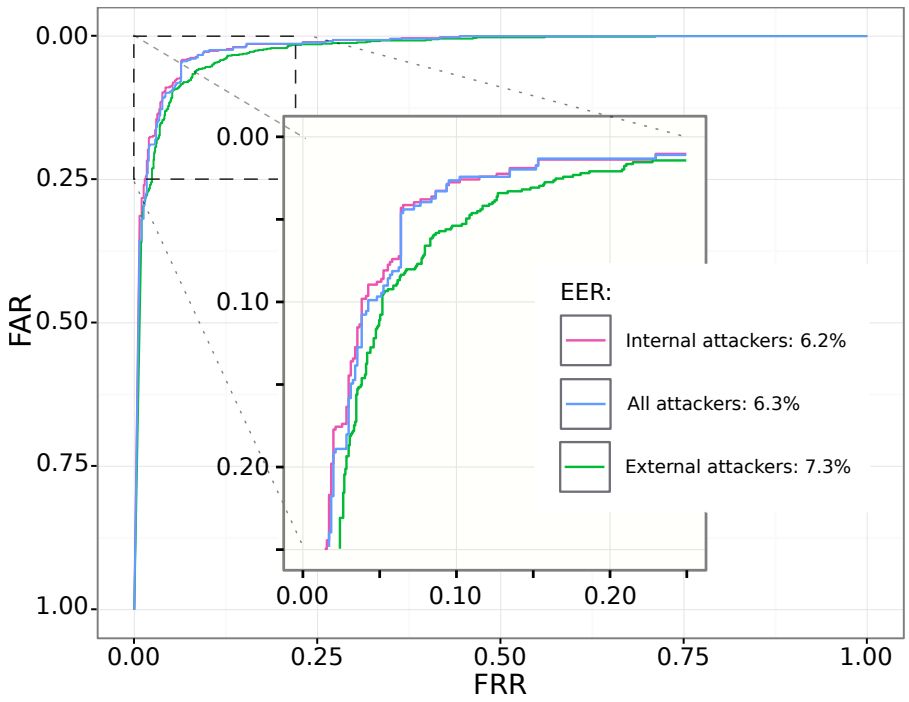


Figure 4.9: ROC curves present authentication trade-offs against impersonation attacks. Red and green curves represent only internal and external attackers, while blue curve shows the overall combined performance. The EER for internal attackers equals 6.2%, while for external attackers it is expectedly slightly higher and amounts to 7.3%. The overall EER for all attackers is 6.3%.

evaluate the performance of the whole authentication system on the remaining one fifth of the legitimate user's gaze-responses that were not used for enrollment. During the evaluation, legitimate user's samples are labeled as positive, while all attack attempts that other users made while pretending to be the legitimate user are labeled as negative. We consider an authentication attempt accepted by the system only if it passes both the identity verification and the freshness verification. For freshness verification, we use a threshold $T = 50\%$.

Besides overall performance, we also separately evaluate two disjunct subsets of the attack attempts: those originating from external attackers, who are unknown to the system, and those originating from internal attackers, whose previous authentication attempts might have been used as negative samples during enrollment.

Results. We show the system performance against impersonation attacks as an ROC curve in Figure 4.9. Since individual user classifiers output a probability that a given sample belongs to the respective legitimate user, we can achieve

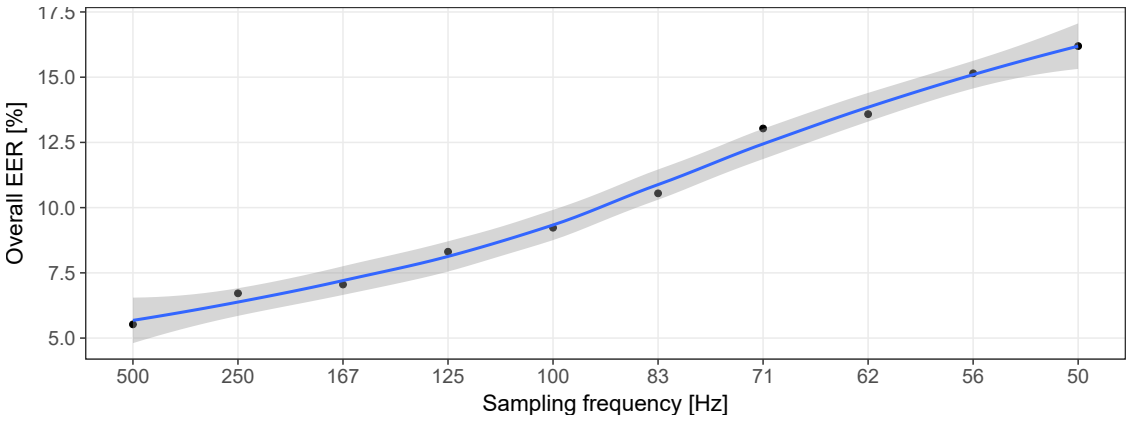


Figure 4.10: Effect of the sampling frequency on the overall EER. Sampling frequencies between 50 and 250 Hz were simulated by decimation: applying a low-pass filter before subsampling the original 500 Hz measurements. As the sampling rate reduces from 500 Hz to 50 Hz, the EER increases by about 11%. However, for frequencies close to 120 Hz, which are supported by a range of affordable eye tracking devices, the error rates are still well below 10%.

different classification performance by varying the threshold above which a sample is considered legitimate. As this threshold increases, so does the likelihood of falsely rejecting a legitimate user (FRR) increase, but at the same time, the likelihood of falsely accepting an attacker (FAR) decreases. Different combinations of FAR and FRR values for three attack scenarios (internal, external, and all attackers) are shown in Figure 4.9. For all three scenarios, it is possible to achieve low FAR values (under 5%) if FRR is increased closer to 10% and vice-versa.

The equal error rate (EER) is defined as the rate at which FRR and FAR equalize. Given that EER is the single measure most commonly used to compare classification performance, we also use it throughout the rest of the chapter. The reported overall system EER is computed using a single, shared, decision threshold for all classification decisions across all users' classifiers.

As expected, in terms of EER, the system achieves slightly stronger performance against internal attackers (6.2% EER) than external attackers (7.3% EER). Overall, the system achieves an EER of 6.3% for impersonation attacks; as we discuss in Section 4.3, this result is preferable to any previously reported performance of gaze-based authentication systems.

4.8.3 Replay Attacks

Recall from Section 4.5.3 that in order to prevent reuse of biometric data, the system verifies that the received gaze-response corresponds to the presented gaze-challenge, i.e., that the user successfully gazed at no less than a chosen percentage T of the stimulus positions presented during authentication.

The result of verifying the freshness of a received response does not depend on the claimed identity during authentication, but only on the positions of the dot in the visual stimulus. Therefore, in order to provide a more comprehensive estimate of the distinctiveness of a challenge-response pair, we report the results for a scenario in which identity verification always returns a positive answer.

Setup. In order to evaluate the probability of success of a replay attack, for each gaze-challenge c_i, we simulate a "replay" of all other gaze-responses r_j to the VerifyFreshness function of the system. We calculate the success rate of replaying r_j to c_i as the percentage of stimulus positions from c_i that would be considered successfully gazed if a user's response was r_j.

Since our dataset consists of 1021 legitimate authentication attempts, each recorded with a unique gaze-challenge, we are able to simulate more than 10^6 potential replay attempts in order to estimate the true reject rate. Furthermore, in order to estimate the true accept rates, we use the same procedure to simulate a total of 1021 legitimate authentication attempts, in which the gaze-response was indeed generated as the user was presented with the matching gaze-challenge.

Results. Figure 4.11 shows achieved performance of the challenge-response verification for different values of T, which we vary from 0% to 100%. As T increases, so does true reject rate (TRR), the ratio of replay attempts that are correctly rejected. On the other hand, this also causes a decrease of the true accept rate (TAR), the ratio of legitimate, fresh attempts that are correctly accepted.

The desired threshold is the one that detects all replay attempts while accepting all legitimate authentication attempts as fresh. Figure 4.11 shows a wide range of potential threshold values that lie between 40% and 60% and almost perfectly

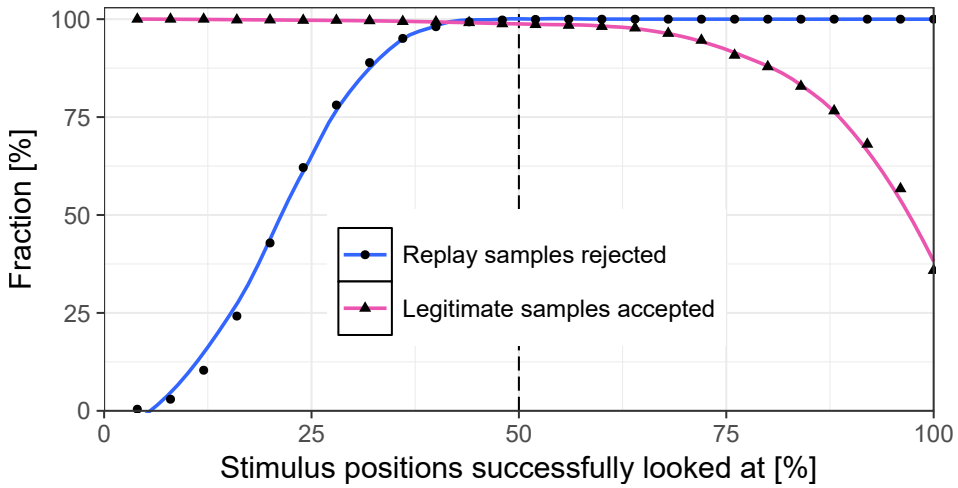


Figure 4.11: Performance of the freshness verification procedure depending on the chosen threshold T. As we change the required percentage of successfully gazed stimuli to classify a gaze sample as "fresh" from 0% to 50%, the ratio of successfully detected replay attempts rises from 0 to close to 1. At the same time, the ratio of successfully classified fresh attempts starts declining as the required threshold increases over 60%, showing almost perfect results for the thresholds between 40% and 60%.

separate the fresh and the replayed gaze-responses. Such a broad range of thresholds that achieve strong classification is a desirable property for any classification system as it gives strong confidence in reported results.

Since we use $T = 50\%$ to evaluate impersonation attacks, we report specific numeric details for this threshold. The results of simulating more than 10^6 challenge-response pairs as replay attempts show that we achieve close to perfect true reject rates (TRR) of 99.94%. At the same time, very few legitimate attempts are incorrectly rejected: the evaluation shows a true accept rate (TAR) of 98.63%, a result of falsely rejecting only 14 out of 1021 legitimate attempts.

Overall, these experimental results show that our system robustly prevents replay attempts for a wide range of thresholds with very high success rates. Moreover, given that the system can detect repeated authentication attempts, and e.g. lock user's account after a certain number of failed attempts, we finally conclude that our system can effectively prevent replay attacks.

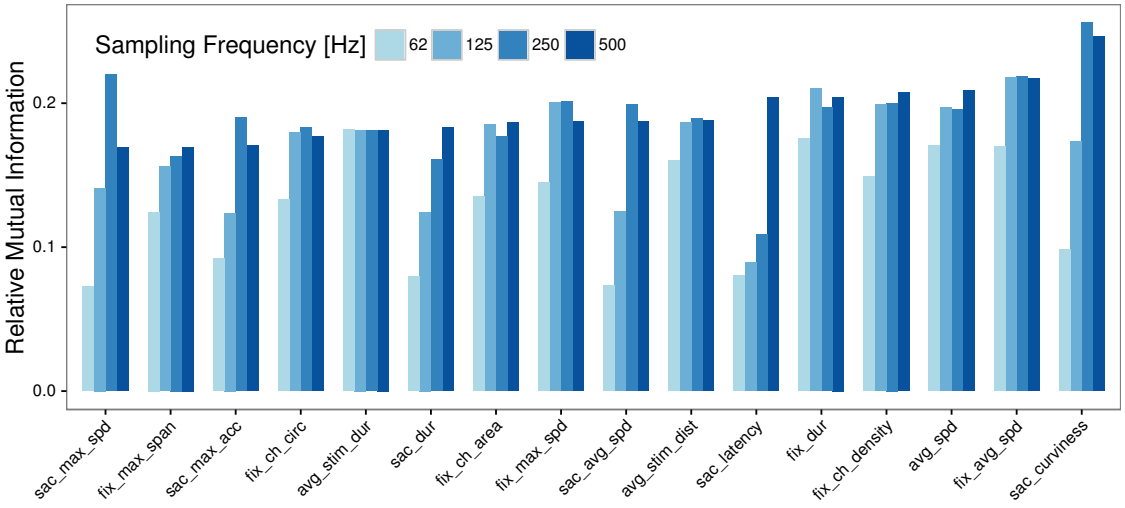


Figure 4.12: Impact of the sampling frequency on the Relative Mutual Information, $RMI_{ID}(F)$, of used features. Features are ordered according to their RMI at 500 Hz. Due to the velocity of saccadic eye movements, the differences in RMI are most visible in saccade-based features as the sampling frequency is reduced to 125 Hz or 62 Hz. On the other hand, the impact on fixation-based features is significantly smaller.

4.9 System Analysis

After evaluating the proposed system's security guarantees, we now use the dataset to analyze how would the system performance change if some of the crucial design choices had been altered. Such analysis helps strengthen future research on the same topic by providing a better understanding of the system behavior, confirming the correctness of choices that were made, and showing potential directions for future improvement.

4.9.1 Sampling Frequency

An important factor in the overall performance of a biometric system is the availability of high-quality data. Even though the sampling frequencies of widely affordable eye tracking devices keep increasing with the proliferation of cheaper high-speed cameras, most consumer-grade eye trackers still predominately capture gaze data between 60 and 240 times per second.

In order to evaluate the feasibility of using reflexive eye movements and the proposed set of features for biometric authentication with a wider range of eye

tracking devices, we now simulate a scenario in which data was acquired at lower frequencies.

Setup. We simulate data acquisition at lower frequencies by first applying a low-pass IIR filter (with a suitable limit frequency) before subsampling the data between 2 and 10 times, i.e., by discarding all but every M-th gaze measurement. This results in 9 new datasets, with sampling frequencies ranging from 250 Hz (M=2) to 50 Hz (M=10), and 9 new sets of features computed using the exact same procedure as with original data. Finally, all 10 sets of features are used to repeatedly train and test a classifier for each user, following the procedure outlined in Section 4.8.2.

Results. The results of the analysis of how the sampling rate influences the overall system error rates are shown in Figure 4.10. As the sampling rate reduces from 500 Hz to 50 Hz, the EER increases by about 11%. However, the measured difference between the error rates at 500 Hz and 125 Hz, which are supported by a range of affordable eye tracking devices, is around 2%, remaining well below 10%.

Consequently, we compute and show in Figure 4.12 the RMI_{ID} values for each feature as they are subsampled with 250 Hz, 125 Hz, and 62 Hz. The figure shows that the RMI_{ID} of several features, mostly those related to the velocity of eye movements during fixations or saccades, actually does increase as the sampling rate goes from 500 Hz to 250 Hz, while the informativeness of the majority of the other features remains relatively unchanged. As a result, the classification performance remains very similar despite the halving of the sampling frequency.

Overall, these results show that, while using a high-speed eye tracking device does indeed improve classification performance and the quality of the extracted features, even when low- and mid-range eye tracking devices are used, the overall authentication success rates are expected to remain high.

4.9.2 Size of the Negative Class During Enrollment

One of the crucial factors that impact the performance of classifiers is the variability of data seen during training. This is especially true for binary classification

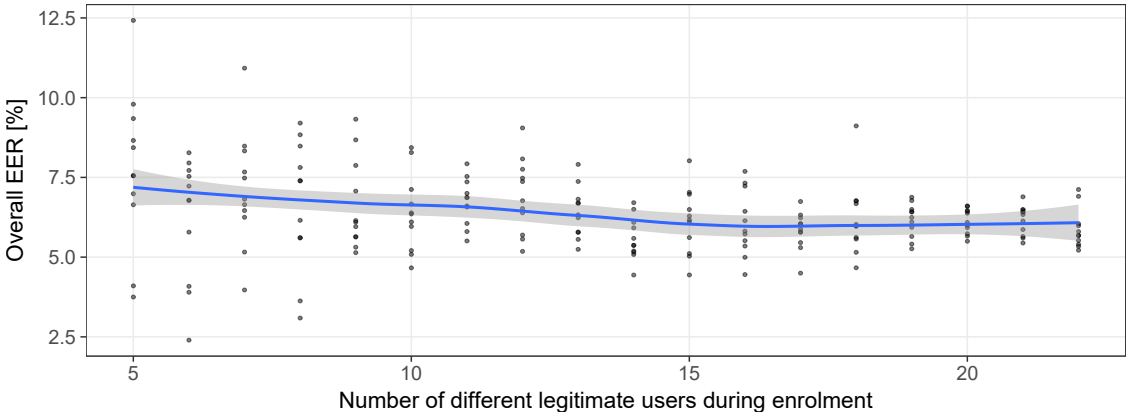


Figure 4.13: Impact of the size of the negative class on the overall EER, computed by varying the number of different users' samples that the classifier is exposed to during enrollment. For each negative class size, we repeat the measurement 10 times to show the variability in performance. Increasing the negative class size decreases the variance in system performance, while at the same time slightly reducing the overall error rates. However, once the size of the negative class reaches about 10, the overall performance stabilizes at 5-7% EER.

performed during biometric authentication, in which a specific classifier is trained for each enrolled user, with the purpose of deciding whether a new biometric sample does indeed belong to the claimed identity or not. During training, if the classifier is supplied with negative samples of limited variability (i.e. other legitimately enrolled users' gaze samples), it is likely to overfit, failing to generalize when required to classify biometric samples of new, never seen individuals. In such scenario, the classifier learns to reject specific characteristics of the seen negative samples, rather than recognizing the characteristics of positive samples and rejecting the rest.

Authentication vs Identification. When discussing classification performance in relation to the number of different classes, it is important to distinguish biometric authentication (1-1 classification) from biometric identification, as the latter requires 1-N classification to determine the identity of the biometric sample. In the identification scenario, introducing each additional class (i.e., increasing the number of users) results in making the problem distinctly harder, the probability of successfully identifying the correct class out of (N+1) is smaller than out of N classes.

However, given that in the authentication scenario the classifier always makes a binary decision, increasing the number of different users seen during training

or testing introduces significant variability only up to some level, after which the performance usually stabilizes, assuming that the samples seen during training and the samples on which the system is tested are representative of the true distribution. As a result, while it is not straightforward to generalize the identification performance of a classifier beyond the number of tested classes, this is less true for authentication performance, assuming that the classifier is exposed to sufficient variability of samples during training and that the testing samples correspond to the actual real distribution.

In order to estimate the required variability of the negative class required to achieve stable biometric authentication performance, we now evaluate the error rates of our system, depending on the number of different users that are available during enrollment of each classifier.

Setup. We simulate the scenario in which only a random subset of legitimate users' gaze tracking measurements are available during enrollment as the negative class. For each size of the negative class, we randomly choose 10 different subsets of other legitimate users, and we repeat the training process three times for each user, computing and reporting the equal error rates using the procedure outlined in Section 4.8.2.

Results. The results of the evaluation are shown in Figure 4.13. As the number of users U increases from the initial scenario of $U = 5$ towards $U = 15$, a decrease in the overall equal error rates, shown as the blue line, is visible, resulting in about 1% stronger performance. However, when $U > 15$, the measurements show no significant difference between the average classification error rates as the number of users increases.

Given that each classifier makes a binary decision "legitimate user or attacker", the amount of additional variability does not increase significantly after the number of users seen during enrollment reaches 15, resulting in comparable classification performance. Additionally, as the size of the negative class seen during training

increases, so does the variability of the measured equal error rates decrease, stabilizing at 5-7% EER for U over 19.

4.9.3 Dwell-time Threshold D

One of the main characteristics of the proposed authentication system based on reflexive eye movements is the interactivity of the stimulus in response to user's gaze. As discussed in Section 4.5, interactively moving the position of the stimulus dot allows the system to extract the maximal number of saccades in a given time, while at the same time reducing voluntary saccades that happen while the user is waiting for the stimulus to change. However, due to head and body movements as well as imperfections in the gaze tracking devices, the location of users' gaze is not always captured perfectly, which results in users sometimes not being able to successfully gaze at the stimulus position for a period of time.

The prototype that we use for experimental data acquisition was built such that the visual stimulus remains at the same position until it is either successfully gazed, or for the duration of the maximum dwell-time threshold ($D = 1000\ ms$). This ensures that the authentication process continues even if the user is unable to gaze at a particular stimulus location, or, e.g., does not pay attention to the screen - in which case the authentication will expectedly fail.

Given that the majority of reflexive eye movements are considered to have latencies below 250 ms [166], and taking into account the distribution of saccade latencies from our experiments (shown in Figure 4.4), it is likely that the presented system could use a lower dwell-time threshold (D) than the one chosen during the experimental data acquisition ($D = 1000\ ms$) without significant loss in authentication performance. Since reducing D is expected to result in a decrease of the authentication times for the users, we now analyze the impact of different values of dwell-time threshold on the system performance.

Setup. Considering that our data was recorded with $D = 1000\ ms$, we can evaluate any $D < 1000\ ms$ by simply discarding those parts of gaze tracking measurements that happen more than D milliseconds after the stimulus last changed

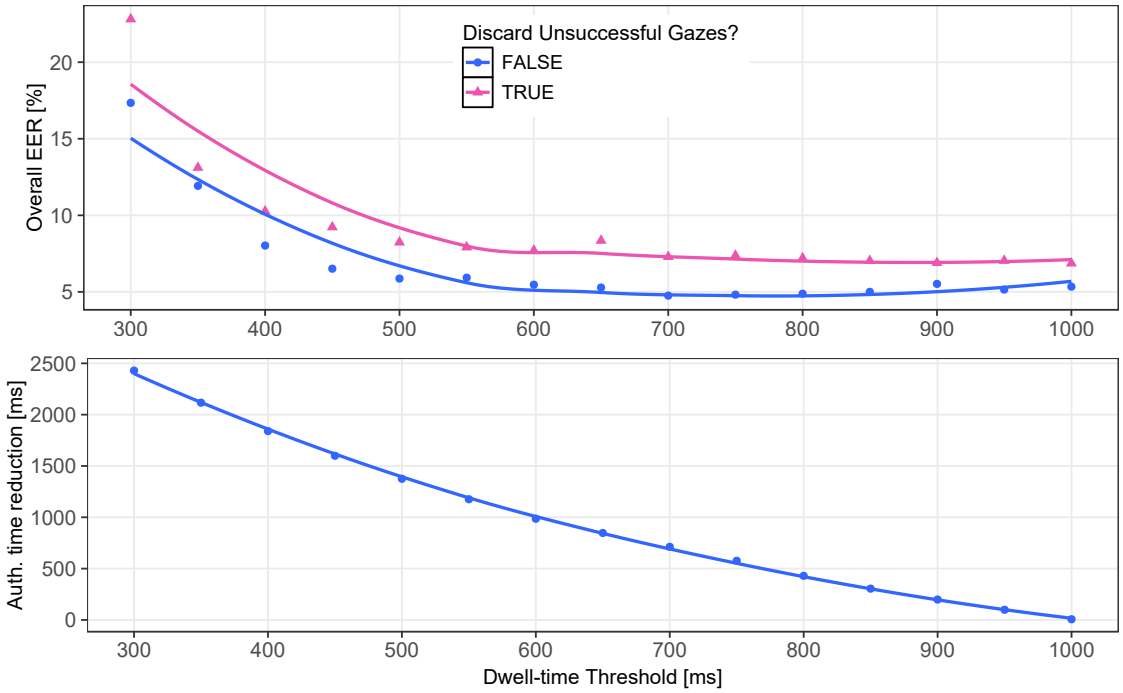


Figure 4.14: Impact of the different dwell-time thresholds (D) on (a) the mean equal error rates and (b) the reduction in median authentication time. In (a), the blue line approximates the performance of completely discarding those parts of user's eye movements which do not correspond in a successful gaze at a certain stimulus location, while the purple line corresponds to the case in which only the parts of the gaze after the threshold are discarded. Considering the physical limitations of eye movements, low dwell-time thresholds expectedly result in high error rates. However, as the threshold reaches 500 ms, error rates stabilize while at the same time reducing median authentication times by about 1500 ms.

position. For each analyzed value D, we create a separate dataset, train a classifier according to the procedure described in Section 4.6.3 and evaluate it according to the description in Section 4.8.2.

Additionally, by *"Discard Unsuccessful Gazes"* we consider the option of completely discarding all gaze samples that were measured for the whole duration while the stimulus dot was shown at a certain location if the user was not ultimately successful at gazing it in the period of D ms. We run this variant of the analysis expecting that measurements from unsuccessful gazing might be more noisy than the measurements which result in successful gazes at the stimulus dot, and that they could consequently impair classification performance, rather than improve it.

Results. The average error rates and the reductions in median authentication times for values of D ranging from 300 ms to 1000 ms are shown in Figure 4.14. In

the upper graph, the purple color indicates the scenario in which unsuccessful gazes are completely discarded when computing the features for classification, while the blue color shows the results of discarding only the samples after the threshold D.

As expected, as D decreases from 1000 ms towards 300 ms, the overall equal error rates increase as well, since some of the useful distinctive information will be discarded. Contrary to our expectation, fully discarding all measurements that did not result in a successful gaze actually increases the error rates, indicating that such gaze data still carries valuable information.

While the error rates for thresholds below 450 ms quickly grow above 15%, it is important to note that the error rates for thresholds above 500 ms are almost identical. This confirms the hypothesis that in most cases, users reflexively gaze a specific stimulus position in less than 500 ms, and that the subsequent behavior carries significantly less useful information.

As an important consequence of this analysis, we see that reducing the dwell-time threshold to 500 ms results in the reduction of median authentication times by as much as 1500 ms, which in turn increases the usability of the proposed system, while not impacting its overall classification performance.

4.9.4 Choice of the Classifier

We continue the analysis of system parameters by evaluating different classification methods and models that we could use for identity verification. Besides the radial-basis SVM (`radialSVM`), which showed the best overall performance on all comparison tests, we also test a set of five other commonly used classifiers that are implemented in the Caret library: Random Forest (`rf`), k-Nearest Neighbours (`kNN`), Generalized Linear Model (`glm`), C5.0 (`C5.0`), and Linear Discriminant Analysis (`lda`). In this subsection, the classifiers are compared according to their achieved Equal Error Rates for a range of dwell-time thresholds D and sampling frequencies.

Setup. We compare the performance of different classifiers by running a battery of tests, in which we vary the sampling rate (125 Hz, 250 Hz, 500 Hz), as well as the maximum dwell-time threshold D (500 ms, 750 ms, 1000 ms). Besides specifying

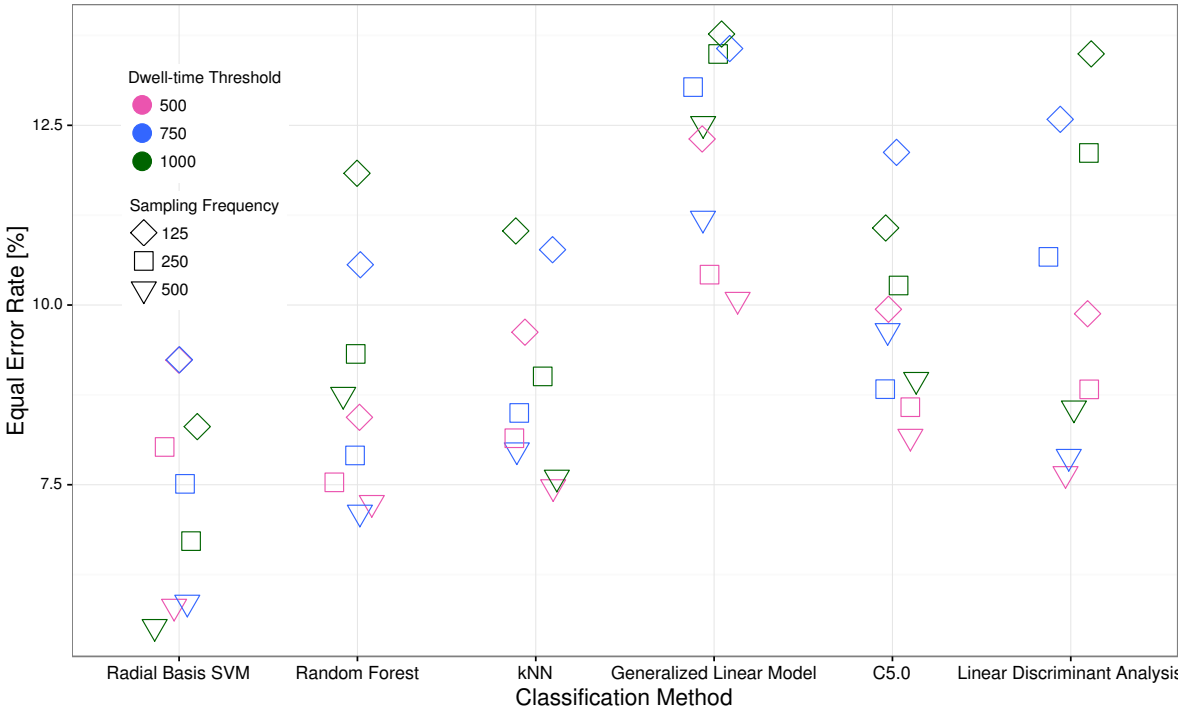


Figure 4.15: Comparison of the equal error rates for 9 different configurations of several classification methods: Radial basis SVMs, Random Forests, k-Nearest Neighbours, Generalized Linear Model, C5.0, and Linear Discriminant Analysis. The used sampling frequencies and dwell-time thresholds are depicted by the shape and color, respectively. Radial Basis Support Vector Machine achieves the lowest error rate in each specific configuration.

the exact classification method to the Caret library, all other parameters are kept the same across different classification methods, consistent with the other computed EER rates: we run a 5-fold, 3-times repeated cross-validation, and repeatedly compute and average the results for each user three times.

Results. The results of each of the 9 evaluations that were run for each of the 6 classifiers are shown in Figure 4.15, with equal error rate being the measure of classification performance. The color of each of the point indicates the cut-off threshold D, while the shape of each point indicates the sampling rate. The radial-basis SVM, which we use in all other analysis, clearly achieves the lower error rates for all combinations, with the Random Forrest classifier trailing a few percentage points behind. It is interesting to note that the results shown in previous subsections, which were all computed using the SVM classifier, indicated that the higher sampling rate and dwell-time threshold D should result in lower EER-s.

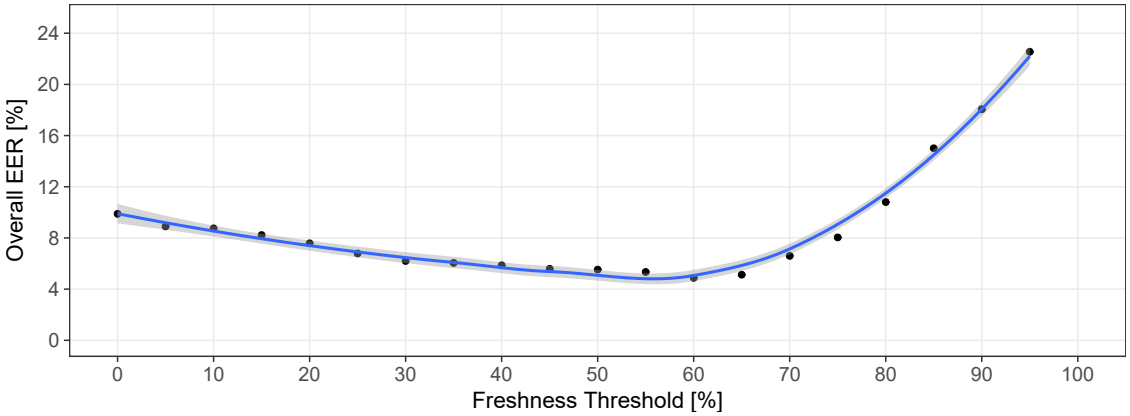


Figure 4.16: Comparison of the equal error rates for different freshness verification thresholds T. The minimal EER is achieved when T is close to 50%. T equal to 0 provides the error rates computed only based on eye movement data classification. While the error rates do increase by about 4%, they still remain at 10.47%.

However, the results in Figure 4.15 show that this is not always the case with other classifiers, especially in the case of low sampling rate, where the situation is reversed, with lower dwell-time thresholds resulting in lower overall equal error rates.

4.9.5 Impact of Freshness Verification Threshold

As described in Section 4.5.3, if the received gaze measurements do not match at least T percent of the randomly chosen stimuli positions, the respective sample is rejected as a potential replay attack. Such rejections, however, do not happen only as a result of replaying a sample that is not fresh. Firstly, they can be a result of changes in the eye tracking geometry as legitimate users carry out their daily work, counting as a false reject that negatively impacts the overall EER of the system. Secondly, since the system always uses the calibration profile of the victim whose identity is being impersonated, such rejection of a fresh measurement can also happen during an impersonation attempt. Such true rejections happen if the calibration profile does not fit the attacker sufficiently to precisely gaze at the necessary percentage and decrease the overall EER of the system.

Even though the attacker is unable to control the threshold T or disable the freshness verification component of the system, analyzing the impact of freshness verification on the overall error rates provides an intuition on the stability of the

combined system and the guarantees that could be achieved in eye tracking systems that do not require user calibration (e.g. eye-tracking glasses).

Setup. We vary the required T from 0% to 100% and compute the overall EER values for the whole dataset. Setting the threshold to 0% effectively computes the overall EER of the system in the case where freshness verification is not taken into account during authentication. On the other hand, setting the threshold to 100% is equivalent to requiring that users precisely gaze at the stimuli for all positions on the screen, resulting in high error rates due to the majority of legitimate attempts also being rejected.

Results. The results are shown in Figure 4.16. The overall EER is lowest when T is between 45% and 65%, which is in accordance with results of replay attacks evaluation (Section 4.5.3)[1].

As T increases, this causes more legitimate authentication attempts to be rejected on the grounds of potentially being a replay attack, and causes a sharp increase in the overall EER against impersonation attacks.

On the other hand, as T decreases, the effect of freshness verification module diminishes, ultimately providing the estimate of the system performance in the case where freshness verification would be completely ignored ($T = 0$). While the error rates increase by about 4% as T reduces from 50% to 0%, they ultimately stop increasing at 10.47%, showing that the performance of the system against impersonation attacks would not significantly decrease if all or most attack attempts passed freshness verification.

4.10 Discussion

Advanced attacks. A more sophisticated attacker could build a model of a legitimate user's eye movements to successfully respond to a given challenge.

[1]We note that replay attacks are evaluated using only legitimate authentication attempts since we assume that an adversary would not try replaying an impersonation attempt.

However, we argue that performing such attacks is not straightforward and requires a higher level of complexity than simply replaying a biometric sample.

Firstly, the adversary is likely to be solving a harder problem than the authentication system; while the system needs to build a discriminative model that allows making a binary decision about user's identity, the adversary needs to actually generate eye movements which correspond to the legitimate user. An indication of the difficulty of artificially creating eye-movements can be found in work by Komogortsev et al. [189], which evaluated the complexity of a significantly simpler problem: artificially generating 1-dimensional eye movements. The paper showed that those movements could be distinguished from natural recordings with high accuracy; creating realistic 2D eye-movements that correspond to a specific user is likely to be significantly harder.

Secondly, by using a challenge-response type of protocol, we ensure that the potential generative model of legitimate user's eye movements must be able to output results interactively and in real-time since the stimulus is not known in advance. This requires an additional level of sophistication that is not needed for replay attacks since the adversary needs to not only control the gaze tracking channel, but to also observe and analyze the visual channel.

Is this sufficiently fast and secure? The most commonly deployed user authentication is password-based as passwords are fairly secure, relatively fast to input, and simple to implement and understand. Considering their prevalence and simplicity, we use passwords as an informal benchmark in terms of authentication times and input error rates to of biometric authentication based on reflexive eye movements.

Over the last few years, a range of studies on password authentication have been published, several of those focusing on evaluation of entry times for different password generation strategies, as well as input and recall error rates. From a usability standpoint, a recent paper by Shay et al. [113] provides an estimate of password usage in a realistic setting and with a large number of users. The authors evaluate multiple password-composition policies by running an online experiment

with 8,143 participants, who are asked to create, remember and recall different passwords. Depending on the required password complexity, the median input times varied from 11.6 to 16.2 seconds, while input error rates ranged between 4% and 7%. The authors also note that more than 20% of participants had problems recalling their password and more than 35% of participants stated that remembering a password was hard.

Considering these findings, we believe that our results, namely a median authentication time of 5 seconds, are comparable to input times on passwords: on average, a successful authentication attempt with our proposed system does not take longer than typing a password, with an added benefit that users need neither learn nor recall any information or procedure. In terms of error rates, while the false rejection rate of 6.3% is comparable to the error rates of password entry (4-7%), it is important to note that a well chosen password provides higher security against zero-effort guessing attacks than most biometrics, eye tracking included. However, an important benefit of the proposed system is its resistance to replay, which password-based authentication lacks: if an adversary ever learns one's password, the system can be compromised at any later time.

Comparison with other biometrics. In order to further make sense of the achieved error rates, we now compare biometrics based on reflexive eye movements to some of the most widely deployed biometric modalities, such as those based on one's face, voice, and fingerprint measurements.

Face verification research has closely followed the recent advances in deep learning, resulting in sharp reduction of error rates achieved by the current state of the art systems. One of those is Google's FaceNet [200], which achieves the accuracy of 99.63% on the Labeled Faces in the Wild dataset [201], a de-facto standard for evaluation of face verification approaches. Recent work continues to further reduce these error rates [202], achieving a true accept rate of 95.6% while keeping the false accept rate at $10^{-}4$, with an accuracy of 99.82% on the Labeled Faces in the Wild dataset.

However, when considering practical deployment of face recognition systems, it is important to note the availability of one's biometric data (*selfies*) online. Following prominent examples of face recognition systems being spoofed by fake photos or videos, many recently deployed systems have started relying on dedicated depth sensing cameras to perform identity verification of one's face geometry instead of only a 2D photo or video. One such example is Apple's Face ID technology [203], for which the company claims that the likelihood of a random other user successfully authenticating is approximately 1 in 1,000,000, without giving any other details.

Speaker verification research is usually split into text dependent and text independent speaker verification. For text independent, RawNet [204] is a recently published deep neural network architecture that achieves state of the art equal error rates of 4.0%, as measured on the VoxCeleb1 dataset [205]. While direct comparisons between text dependent speaker verification systems are not straightforward, as an example we not that Apple's verification performance for the activation phrase 'Hey Siri' achieves an equal error rate of 4.3% [206].

Both in case of text independent and text dependent speaker verification, it is important to take into account that replay of voice recordings needs to be independently prevented.

Fingerprint verification research usually achieves among the lowest error rates for biometric systems, given that fingerprints are mostly static, fast to capture, and highly individual. As an example, current state of the art solutions achieve equal error rates as low as 0,543% on the long-running Fingerprint Verification Competition [207]. In terms of widely known practically deployed systems, Apple states that the chances of an unauthorized individual successfully authenticating to TouchID is approximately 1 in 50,000 [203].

However, similarly to face and voice verification systems, significant additional complexity needs to be layered on top of the reported error rates for verification in order to ensure that spoofing and replay attacks against these biometric systems are prevented.

Applications, limitations, and future work. We now discuss several challenges that remain before the proposed concepts can be applied in a wider range of practical applications: namely the practicality of achieved error rates, the temporal stability of the eye movement biometrics, and the use of static eye tracking devices.

Firstly, the equal error rates between 6% and 10% are not yet sufficient to be independently deployed in real-world systems. However, it is important to note that a real-world authentication system could combine the evaluated dynamic features with some of the static features often available as part of the standard eye-tracking procedures, such as pupil sizes, distances between user's eyes, or even iris images. Even though such features could be controlled by carefully designing the light sources or wearing masks, they increase the required effort from the adversary, while at the same time significantly reducing the error rates in previous research: from EER of 20-30% to EER of 10-20% [195]. Taking this research direction further, a potential future application of ideas proposed in this chapter is in increasing the security of various face recognition systems. Many such systems already implement measures to prevent sophisticated spoofing attacks [208], e.g., by requiring users to smile, move their head, or to gaze at the direction of the camera [209]. Consequently, combining the stability and low error rates of face recognition with the dynamic characteristics and the freshness verification of reflexive eye movements could retain the usability of face recognition while ensuring that an adversary cannot spoof such a system using the currently available methods against face recognition.

An important requirement for potential long-term biometric use of this biometric is to evaluate and improve the temporal stability of the proposed eye movement features over extended periods of time. After having shown in this work that the proposed visual stimuli does indeed quickly extract features that allow discrimination between users while at the same time preventing replay attacks, we plan to next focus on designing an extensive larger set of potential features, capturing larger datasets with different eye tracking devices, and evaluating their long-term stability over multiple sessions. Furthermore, the performance of various deep learning models is surpassing traditional machine learning approaches in many classification

scenarios for which sufficiently large datasets are available. Capturing such an extensive dataset would allow that the support vector machine classifier is replaced by a trained deep learning model, possibly resulting in further decrease of overall error rates. We leave this as an avenue for future work.

Before widescale deployment, we note that the experiments were done in a controlled environment with subjects who had no apparent eye problems. It is likely that the proposed authentication system would not achieve the same level of performance for individuals with specific eye conditions. Furthermore, the experiments did not take into account various potential factors that could influence one's eye behavior, such as tiredness or influence of medications or alcohol.

Finally, as a proof-of-concept evaluation of using reflexive eye movements for authentication, we measured the gaze samples with a static, high-end eye tracker. Given that MR and VR headsets provide a display on which the proposed reflexive stimulus could be shown to the user, and they are expected to have eye tracking capabilities in their future iterations, we look forward to applying the proposed system to such headsets in our future work.

4.11 Summary

Building upon the core idea of using reflexive human behavior for authentication, in this chapter we designed the interactive visual stimulus for rapidly eliciting standardized reflexive eye movements, and showed how this stimulus can be used to construct a fast challenge-response biometric system. Based on a series of user experiments, we showed that our stimulus indeed elicits predominately reflexive saccades, which are automatic responses that only pose low cognitive load on the user. As a result of using reflexive behavior that is fast and stable, we show that our authentication system achieves fast authentication times (median of 5 seconds) and low error rates (6.3% EER for impersonation attacks).

Most importantly, however, our proposed authentication method shows resilience against replay attacks, a property difficult to achieve with most biometrics. Our evaluation shows that the system is able to detect the replay of recorded eye traces

with a very high probability of 99.94%, thus preventing one of the most applicable attacks on biometric systems.

Considering the recent proliferation of reliable and affordable eye tracking devices, and their increasing integration into MR headsets, adding biometric user authentication based on one's eye movements to those systems could become possible as an inexpensive software update. By proposing a visual stimuli that elicits reflexive and predictable eye movement response, this chapter therefore makes an important contribution towards adding fast and replay-resilient biometric authentication on mixed reality headsets.

Since you cannot do good to all, you are to pay special attention to those who, by the accidents of time, or place, or circumstances, are brought into closer connection with you.

— Augustine of Hippo

5

Device Pairing for Mixed Reality Headsets

Contents

Figure 5.1: Real-world view of a shared mixed reality experience of two of our study participants. In this chapter, we focus on the challenge of securely and usably pairing mixed reality headsets in order to support a multitude of collaborative MR use cases and applications.

As we discussed in Chapter 2, one of the core applications of the mixed reality technology is to create shared mixed reality experiences for co-located participants, in which they experience and interact with identical virtual objects in their physical surrounding.

After proposing a method to securely authenticate users to their devices in the previous chapter, we now proceed by proposing a method to authenticate those devices among themselves. More precisely, we now tackle the challenge of secure pairing of two mixed reality headsets. From each user's perspective, this challenge boils down to ensuring that the device with which their headset has just exchanged the cryptographic keys is indeed the device on the head of the user with whom they wish to share an MR experience.

5.1 Introduction

A core requirement for many multi-user mixed reality experiences, many of which are security- and privacy-critical applications, is the ability to directly connect multiple headsets by establishing an encrypted communication channel. This channel is subsequently used to exchange the world anchors, establish a shared coordinate system, and synchronize users' interactions with identical, precisely located holograms between their devices. However, when two previously unassociated devices establish a connection for the first time, it is crucial to prevent potential man-in-the-middle (MITM) attacks. If there exists a third party that both users trust and can securely connect to, such as a central cloud server, this third party can be used to establish the communication by being an intermediary through which users exchange their keys. However, reliance on a trusted third party can not always be assumed. This is often the case in many scenarios in which Internet connectivity is limited, or in which users want to establish a decentralized, P2P network of connections.

Since the adversary is assumed to fully control all network communication, in an MITM attack he can intercept all direct communication between the legitimate users' devices and establish a separate encrypted channel with each legitimate user. The adversary can now subsequently eavesdrop on communication by decrypting and re-encrypting each message with the appropriate key exchanged with each legitimate participant. Such attacks are usually prevented by the use of another, out-of-band channel that the adversary does not control. After the initial key exchange, the legitimate protocol participants must verify that the keys that their devices exchanged and agreed upon are indeed identical and not modified by the adversary [210].

The Challenge. Despite the extensive previous research on secure device pairing, which we overview in Section 5.9, existing protocols are not directly applicable to MR headsets since they fail to address the challenges specific to this technology. Firstly, instead of assuming a single user who controls two devices, MR pairing necessarily involves two users, each with their own headset. Since MR headsets

are designed to be worn continuously, users must not be required to take them off and can, consequently, only observe the output from their own device. Secondly, considering the proliferation of MR devices, each of which include multiple front-facing cameras, it is necessary to assume that the adversary can eavesdrop (but not completely control!) all out-of-band communication. This is not a typical adversary model considered in device pairing, where the adversary usually only controls the wireless network.

Furthermore, given that mixed reality headsets have only recently become available on the consumer markets, there exists almost no previous work that is applicable to device pairing of MR headsets. The only publication that tackles this challenge relies heavily on the assumption that MR headsets will have precise wireless localization capabilities, which are currently not available in any consumer-facing MR device [110]. Consequently, despite the large number of published and available MR applications on different app stores, there currently exists no implementation or practical research proposal to securely perform direct pairing of two or more MR headsets without relying on a third party service.

Contributions. We address this challenge by relying on the unobservability of MR displays and by exploiting HoloLens' state-of-the-art inside-out positioning. This allows us to design protocols that are based on precisely positioned shared holograms to augment users' communication channels, while at the same time retaining the usability of the system.

In this chapter, we design *HoloPair*, a practical and usable system that achieves secure pairing of two mixed reality headsets. In order to evaluate its security guarantees, usability, and performance, we implement a working prototype of the system using two Microsoft HoloLens devices, and run a comprehensive user study with 22 participants. As the measurements show, the majority of participants are able to successfully detect simulated attacks or confirm that pairing was successful in as little as 8 seconds. Despite participants' lack of earlier experience with MR headsets, the results of the post-experiment questionnaire show high subjective opinion of the usability of the *HoloPair* system.

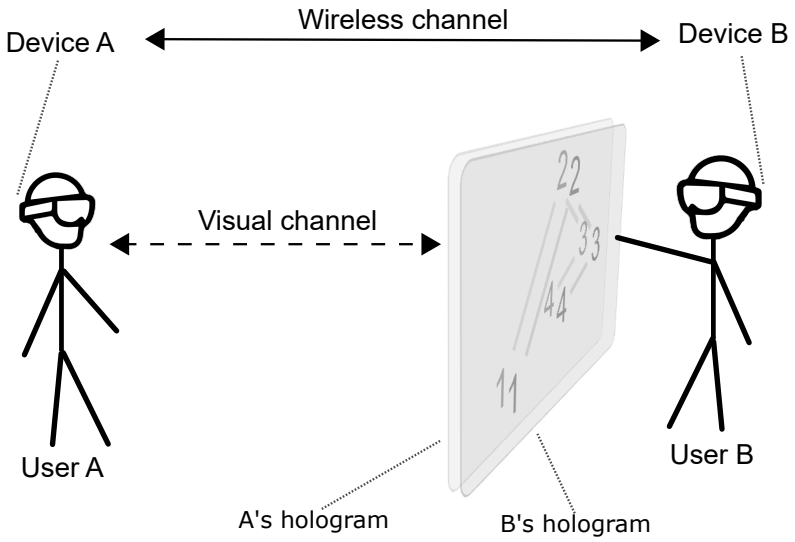


Figure 5.2: System model. Two individuals, each equipped with an MR headset, wish to establish a secure channel for subsequent communication. Devices communicate over the high-bandwidth wireless channel, which is controlled by the adversary. Users communicate over low-bandwidth visual channel, which can be eavesdropped, but not intruded by the adversary, since that would be detected by legitimate protocol participants. Each headset can independently and unobservably overlay arbitrary content over their user's view of the physical world.

5.2 Assumptions and Goals

We assume a scenario in which two users, $\mathbf{U}_A$ and $\mathbf{U}_B$, that do not have a pre-shared secret and are both wearing a mixed reality headset (such as Microsoft HoloLens), meet at in person and want to securely connect their devices to share an MR experience.

5.2.1 System Model

As shown in Figure 5.2, $\mathbf{U}_A$ and $\mathbf{U}_B$ are each equipped with a trusted device, $\mathbf{D}_A$ and $\mathbf{D}_B$, which can augment their view of the physical world by independently drawing precisely positioned holograms.

We assume that the MR headsets can communicate over a high-bandwidth **wireless channel** and have no other direct channel of communication. Users, however, can communicate over an out-of-band audio or **visual channel** on which their headsets can independently and unobservably overlay content. Consequently, a way to view the system model from the perspective of the headsets is to imagine

that they are using human participants as output channels for their communication over the out-of-band channels.

In our system model, we aim to not rely on the audio channel, as it is inherently undirected, polluted by each additional participant, and easily injected into. On the other hand, the visual channel does not depend on the environment noise, can be used in many scenarios where silence is expected (e.g. during lectures or meetings, in public spaces), and is significantly harder to undetectably inject into [211].

We assume that users do not have access to any other trusted third-party service that they could use to establish the connection, such as PKI infrastructure. Finally, since human participants are part of the protocol, we assume that each user will be involved in only one protocol run at any given moment.

5.2.2 Adversary Model

We assume a Dolev-Yao style network adversary [212], whose goal is to use his device(s) to prevent $\mathbf{U}_A$ and $\mathbf{U}_B$ from establishing a secure connection by positioning himself as the man-in-the-middle by assuming an active role in protocol execution.

In contrast to most previous work on device pairing, we consider the adversary to be co-located and able to position himself arbitrarily close to one or both protocol participants. While this gives him the ability to fully eavesdrop on the visual channel, he must, however, remain passive on the visual channel, since any intrusion would be detected as suspicious behavior by the legitimate protocol participants. For instance, if the protocol requires $\mathbf{U}_B$ to make a gesture, the adversary is unable to prevent this from happening ("jamming" it) since that would be detected by $\mathbf{U}_A$, and would cause the protocol to abort. Similarly, if adversary tried to position himself between the legitimate participants and performing a gesture, this would be detected as suspicious. However, the adversary can not observe the independently generated holograms that each legitimate user's device overlays over their view of the physical world.

We consider denial-of-service attacks to be out of the scope of this work.

5.2.3 Design Goals

Based on the discussion in the previous sections, we state the design goals for a successful system for pairing of mixed reality devices as:

- **Security.** The attacker must have a low chance of a successful man-in-the-middle attack. Users should detect attempts of a man-in-the-middle attack in the majority of cases.
- **Usability.** The system must not require users to take their headsets off. The interaction should be relatively short and users should be willing to perform it whenever they wish to share an MR experience with a new user/device. Most pairing attempts should result in a successful key confirmation.
- **Hardware Requirements.** The system should not require capabilities that are not available on current devices (specifically MS HoloLens). In order to allow seamless execution whenever two MR users decide to share their mixed realities, the proposed pairing system should neither have high computational, memory, nor energy requirements.

5.3 The *HoloPair* System

We now present *HoloPair*, a system that achieves usable and secure authenticated key exchange of two mixed reality headsets without relying on any third party.

5.3.1 System Overview

Our system builds upon the general idea of establishing a secure communication channel using Short Authenticated Strings [213], which we simplify and adapt to the specific usability considerations of MR headsets: having two users that should not be required to not take their devices off their heads, an adversary who can observe all their actions.

The devices first exchange their public keys over the high-bandwidth, but insecure channel, then commit and agree on a specific instance of a weak-hash.

Finally, the human participants confirm the authenticity of the exchanged keys using the low-bandwidth visual channel whose integrity is guaranteed.

Role of human protocol participants. In order to prevent man-in-the-middle attacks, the system relies on device owners to establish an out-of-band communication channel and authenticate the exchanged keys. During this manual process, they are aided by their MR headsets, thus allowing them to seamlessly confirm relatively high entropy in comparison to previous approaches.

5.3.2 Pairing Protocol

At the end of a successful protocol run, two previously unfamiliar devices ($\mathbf{D}_A$ and $\mathbf{D}_B$) should have exchanged and authenticated a pair of public keys, which can subsequently be used to bootstrap secure communication, for instance by deriving a shared symmetric key. The protocol, shown in Figure 5.3, consists of the initial device discovery and parameter agreement, followed by the three main steps:

0. Prerequisites. $\mathbf{U}_A$ initiates the protocol by broadcasting the willingness to pair and distributing the public parameters of the underlying cryptographic functions, together with the location of the *WorldAnchor*. Any other device that receives the broadcast will assume the role of $\mathbf{D}_B$ as soon as their user confirms the willingness to pair.

1. Public key exchange. $\mathbf{U}_A$ sends their public key $P_{\mathtt{KA}}$ and $\mathbf{D}_B$ responds by sending the public key $P_{\mathtt{KB}}$ over the wireless channel.

2. Weak-Hash Commitment. Since the out-of-band visual channel has low bandwidth, using it to directly compare and authenticate the exchanged keys would require unusably long time. Therefore, human participants are usually required to compare values (strings, images, colors, shapes) generated from different *weak-hash* functions, which have significantly smaller output entropy, and thus a larger probability of hash collision.

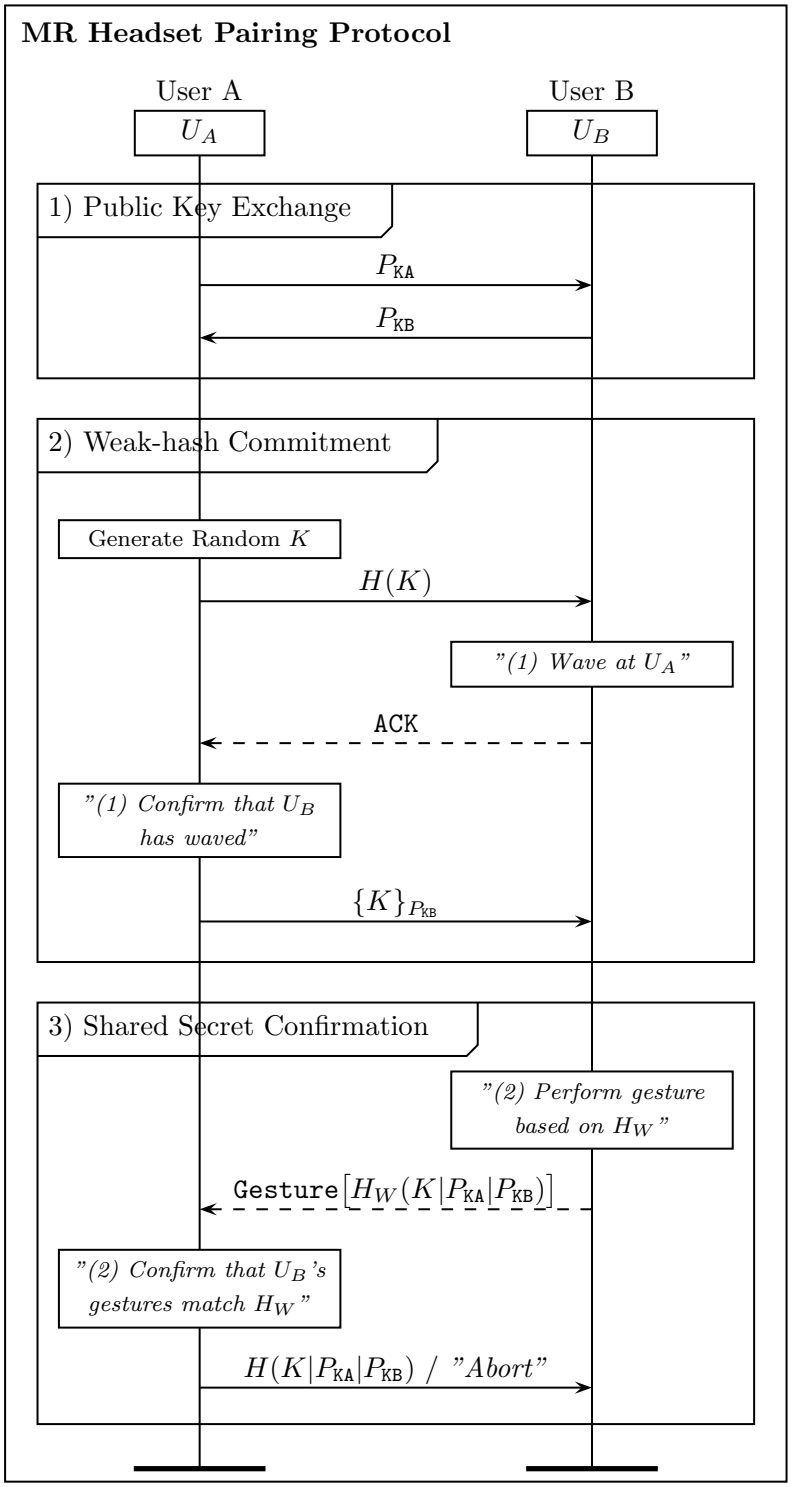


Figure 5.3: The *HoloPair* key confirmation protocol consists of three main steps. (1) Using the insecure channel, the devices exchange their public keys. (2) Next, device A commits to a specific instance of the weak-hash H_W and privately opens it after U_B acknowledges receipt. (3) Finally, U_B uses the low-bandwidth visual channel (dashed lines) to communicate the weak-hash H_{W_B}, which U_A verifies and confirms/aborts the execution in the last message.

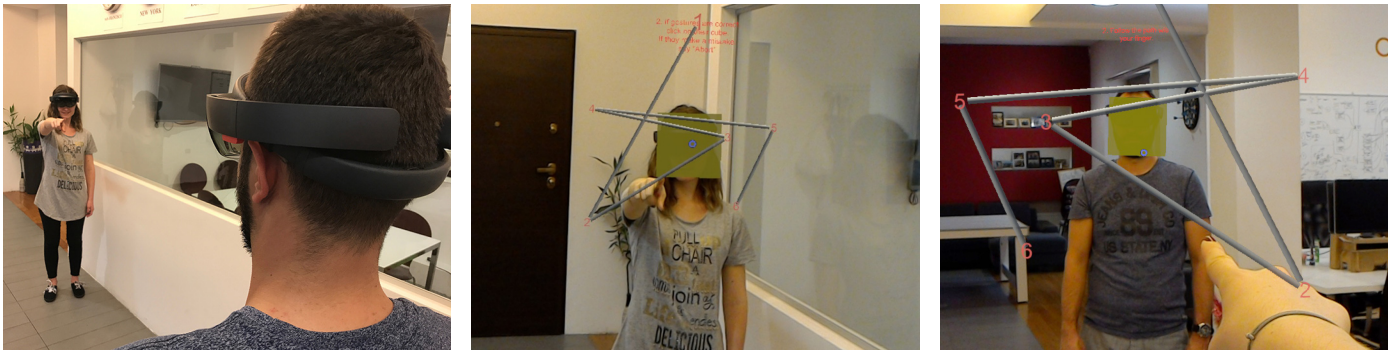
In order to prevent an active man-in-the middle attacker from performing an off-line collision attack on the weak-hash by finding a suitable pair ($P_{\texttt{KA}}$', $P_{\texttt{KB}}$'), $\mathbf{U}_A$ commits to a specific instance of the weak-hash, defined by randomly chosen value K and sends its hash to $\mathbf{U}_B$. After $\mathbf{D}_B$ receives the commitment H(K) in the third message, $\mathbf{U}_B$ is instructed to acknowledge the receipt over the visual channel (for instance by waving to $\mathbf{U}_A$), after which $\mathbf{U}_A$ opens his commitment by sending the encrypted value of K to $\mathbf{U}_B$. Briefly stated, this forces the attacker to actively commit to a set of replacement keys before the actual instance of the weak-hash function is known. We further discuss the need for weak-hash commitment and visual acknowledgment in the security analysis in Section 5.4.

The encrypted message also includes the *WorldAnchor*, which specifies the origins of the shared coordinate system and the transformations between two devices.

3. Shared Secret Confirmation. After exchanging the value K, both devices can now independently compute the weak-hashes from the received public keys, $H_W(K|P_{\texttt{KA}}|P_{\texttt{KB}})$, which will be identical only if the exchanged keys are indeed authentic.

The reason for user participation in the protocol is to confirm that the exchanged public keys are authentic, and consequently, that the weak-hashes independently computed from their values indeed match. By relying on the unique capabilities of MR headsets, we increase the usability of the comparison by guiding users with holographic objects shown in space between them, which are generated to uniquely encode the value of the weak-hash values computed on their headsets.

Depending on the characteristics of the generated hologram, $\mathbf{U}_B$ is required to perform a specific gesture, while $\mathbf{U}_A$ confirms that the observed gesture matches his expectation. Finally, if $\mathbf{U}_A$ concludes that the shared key confirmation was successful, $\mathbf{D}_A$ sends the full hash $H(K|P_{\texttt{KA}}|P_{\texttt{KB}})$, which $\mathbf{D}_B$ uses to confirm that the pairing was indeed successful and displays the message to $\mathbf{U}_B$.



(a) Users as seen outside of MR. **(b)** View of User A. **(c)** View of User B.

Figure 5.4: Views of both users as they are using the *HoloPair* system to pair their devices. $\mathbf{U}_B$ needs to follow the generated object with their hand, while $\mathbf{U}_A$ observes the hand movements and confirms that they indeed correspond to the hologram generated on his $\mathbf{D}_A$ from the value $H_W(K|P_{\mathtt{KA}}|P_{\mathtt{KB}})$.

5.3.3 Gesture for Shared Secret Confirmation

We emphasize here that our protocol does not depend on the specific gesture (or some other procedure) that users use to confirm that they indeed share the same secret value $K|P_{\mathtt{KA}}|P_{\mathtt{KB}}$. Consequently, while designing and developing the *HoloPair* system, we implemented and tested several different versions of the shared secret confirmation, which we fully describe in Section 5.7.

We base the remainder of this chapter on the shared secret confirmation scheme that was shown to be the best performing, both in terms of theoretical security guarantees and subjective usability in our pilot user study (Section 5.6.1). As depicted in Figure 5.2 and visible in Figure 5.4, each independently computed weak-hash is used to construct a holographic shape that consists of N positions on a plane in the physical space between users. Given the exchanged *WorldAnchor* and the precise positioning, both MR headsets show the holographic shapes at exactly the same location. In order to verify that the constructed shapes match, $\mathbf{U}_A$ observes (on the visual channel) as $\mathbf{U}_B$ moves their finger along the generated shape and thus confirms that the weak-hashes generated on both devices indeed match, which is only possible if the exchanged public keys and the value K are authentic.

5.4 Security Analysis

We now evaluate the design of the *HoloPair* system according to the security goals from Section 5.2.3. First, we analyze the protocol's guarantees in a short security sketch in Section 5.4.1 and then discuss the likelihood of a successful random hash collision attack in Section 5.4.2.

5.4.1 Security Sketch

The attacker succeeds if he is able to eavesdrop on the communication between A and B after the protocol execution. Given that we base out protocol on existing proposals for establishing secure communication that are based on comparison of short secrets, we refer to earlier work for an extensive security proof [213] and provide an overview of the arguments in several claims:

Claim 1. In order to successfully eavesdrop on subsequent communication, the adversary must force legitimate users $\mathbf{U}_A$ and $\mathbf{U}_B$ to agree on a different set of public keys than they originally envisioned, $P_{\mathtt{KA}}$', $P_{\mathtt{KB}}$' without aborting the protocol execution. Otherwise, protocol participants would assume that the exchanged keys are not safe to use and likely repeat the protocol.

Claim 2. Since the attacker can not remain undetected if he tries intruding the out-of-band holographic channel in the last step, he must ensure that the out-of-band shared secret confirmation finishes successfully from both participants' point of view. Assuming attentive users (we discuss this assumption later), this can happen only if the weak hashes computed by both $\mathbf{D}_A$ and $\mathbf{D}_B$ are equal, namely that $H_{W_A}(K|P_{\mathtt{KA}}|P'_{\mathtt{KB}}) = H_{W_B}(K'|P'_{\mathtt{KA}}|P_{\mathtt{KB}})$.

Claim 3. Even though the entropy of H_W is not large, the attacker who tries impersonating $\mathbf{U}_A$ to $\mathbf{U}_B$ is required to commit to some value K' in Step 2, before discovering the value K (actually chosen by $\mathbf{U}_A$) in Step 3. This prevents the attacker from being able to perform an extensive off-line search for some suitable K' in the smaller output space of potential values of H_W that would result in a

weak-hash collision. As a result, the likelihood of choosing such K' that results in exactly the right combination of $P_{\texttt{KA}}$', $P_{\texttt{KB}}$', and K' is inversely proportional to the output space of H_W. We emphasize that the MITM attacker could only learn the plaintext value of K sent by $\mathbf{U}_A$ after already committing to some values $P_{\texttt{KA}}$' and K' towards $\mathbf{U}_B$. Consequently, the attacker has no means of finding alternative values $P_{\texttt{KA}}$' and $P_{\texttt{KB}}$' whose resulting holograms appear more similar to human participants than any randomly chosen pair of weak-hashes.

Claim 4. Despite the initial exchange of the *WorldAnchor* over the insecure channel, its modification does not help the attacker since it can only cause a linear translation of $\mathbf{U}_B$'s holograms in the physical space. This would result in a mismatch between the positioning of holograms shown to $\mathbf{U}_A$ and $\mathbf{U}_B$, and cause one of the participants to abort the protocol.

In **conclusion**, the attacker whose goal is to position himself as a passive man-in-the-middle between $\mathbf{D}_A$ and $\mathbf{D}_B$ during their initial pairing has no better choice than trying to randomly guess a pair of replacement keys $P_{\texttt{KA}}$', $P_{\texttt{KB}}$' that would yield the same hashes H_W on both devices. Consequently, his chance of success is inversely proportional to the entropy of the output space of the weak-hash H_W, which directly depends on the chosen variant of the holographic shared secret confirmation. We analyze this probability in the next section.

5.4.2 Probability of a Weak-hash Collision

We now analyze the likelihood that a different pair of keys still results in a weak-hash collision $H_{W_A}(K|P_{\texttt{KA}}|P'_{\texttt{KB}}) = H_{W_B}(K'|P'_{\texttt{KA}}|P_{\texttt{KB}})$, for the shared secret version of the confirmation step described in Section 5.3.3. We analyze the other two variants of the shared secret confirmation in Section 5.7.

The shape of each possible instance of the shared hologram is uniquely determined by its N coordinate pairs (X_i, Y_i). In our implementation, we use a total of 10 different values for both X_i and Y_i, which results in a probability that another pair

of keys results in exactly the same shared hologram to be:

$$P(N) = \frac{1}{(10 \times 10)^N} = \frac{1}{100^N}$$

We note here that due to headsets' holographic guidance, the theoretical entropy of the shared secret confirmation step is significantly larger than, for instance if users would be required to read a sequence of strings or digits of a given number, as has been proposed in previous work. This additionally confirms the usability benefits that mixed reality devices can offer to many existing systems and security schemes.

Finally, by adapting the length of the sequence (defined by N) each of the variants of the confirmation step can be adapted based on the security needs and expectations of a specific scenario.

5.4.3 User Inattentiveness

Given the high output entropy of the used gestures for shared secret confirmation, it is likely that the most probable reason of attack success is user inattentiveness, which results in users not verifying the sequence carefully, or even immediately clicking *"Accept Gesture"* before any gesture was made by $\mathbf{U}_B$.

The problem of user attentiveness is a challenging one, both in terms of performance evaluation, and in terms of designing interfaces that would encourage one to pay attention that has received wide interest from the research community [214].

In order to give an estimate of the ability of *HoloPair* users to detect potential attacks, in the next section describe a working prototype of the *HoloPair* system. We use the prototype to run a user study with 22 participants in which we simulate a man-in-the-middle attack in 20% of pairing attempts to experimentally evaluate the security guarantees of the *HoloPair* system.

5.5 System Prototype

In order to experimentally evaluate the feasibility, security guarantees, and performance of the proposed *HoloPair* system, we build a working prototype using two

Microsoft's HoloLens devices and we make the source code and the implementation available to the public.

5.5.1 Source Code and Development

The prototype is written in the C# programming language, using the Unity framework [58]. When building the functionality specific to HoloLens, we rely the components from Microsoft's official HoloToolkit-Unity repository, which provides functionality such as spatial mapping, world anchors and gesture recognition.

HoloToolkit-Unity is a public repository on GitHub, with many contributions (merged pull-requests) coming from the wider developer community. We thus created a fork of the official repository, and packaged our prototype as one of the provided examples according to Microsoft's instructions. Excluding external references, our prototype consists of 3241 lines of C# code, which are located in the `Assets/Examples/HoloPair` folder.

The source code is available online. Since the motivation behind our work was not only to suggest a suitable pairing protocol, but also to improve the current security practices of the Windows Mixed Reality platform, we have started the process to have our code included into the official HoloToolkit-Unity repository. Furthermore, the source code of the prototype implementation is publicly available at:

`https://tinyurl.com/holopair`

Building and Contributing. In order to build and run *HoloPair*, one should clone the repository, load the main HoloToolkit project in Unity, and open the `HoloPair` scene. After creating a Visual Studio solution from Unity, the solution should be deployed on two HoloLens devices connected to the same wireless network. The first device that loads the application will assume the role of $\mathbf{D}_A$, while the other will assume the role of $\mathbf{D}_B$.

We have made our best-effort to make the code readable and easily extensible for further development. Since we plan to continue actively developing the *HoloPair*

prototype, we will gladly accept any comments, suggestions, or pull requests. For more details, see the related technical report [215].

5.5.2 Main Implementation Components

We now briefly discuss the implementation of the main components.

Networking and device discovery. We use Unity's High-Level Networking API to discover other devices that are running the *HoloPair* prototype by broadcasting/listening to specific messages on port 8888. While our current prototype assumes that devices share the same wireless network, there is no limitation to extend the prototype and support direct ad-hoc wireless connections in the future.

Cryptographic functions. We use the standard Microsoft's implementations of the 2048-bit RSA PKCS1 for asymmetric cryptography and 256-bit SHA2 for hashing.

Constructing the shared hologram from the weak-hash. The shapes that represent *weak-hashes* are generated by extracting bits from the base-64 string representation of the full hash in order to generate N coordinates that define them. In our implementation, for each of N points that comprise the shape, we extract sufficient number of binary bits to generate one of 10 different X coordinates and one of 10 different Y coordinates.

Hologram sharing - establishing the shared *world anchor*. In order for multiple devices to show the identically located content to their users, they must first agree on a shared coordinate system that will be used as a frame of reference regardless of users' subsequent movements. In our prototype, we use HoloToolkit's implementation of *"World Anchoring"*, which in most cases achieves positioning errors smaller than a few centimeters.

Positioning of the shared holograms. In our current implementation, the shared hologram is shown on a line between users, initially 1.65 m from $\mathbf{U}_B$, and then moves towards the $\mathbf{U}_B$ during a period of 3 seconds, to finish at a distance

of 0.85 m. We've made the design decision to implement such movement in order to ameliorate the slightly limited field of view of the current version of HoloLens ($30° \times 17.5°$). This allows $\mathbf{U}_B$ to first get the full view of the shape, and then to be close enough so that they can reach it with their hands.

Confirming protocol success & aborting. In the current prototype, $\mathbf{U}_A$ confirms that the observed gesture was correct by *gazing* at $\mathbf{U}_B$'s head and performing a *click* gesture, for which we use HoloLens' gesture recognition module. We deliberately use a gesture instead of voice commands to prevent the attacker from making an attack successful by simply generating, potentially even inconspicuously [211], a confirmation voice command.

However, we believed that it would be more convenient to use voice recognition for the case when users suspect to have detected an attack attempt. In such cases, we asked users to say *"Abort"*, expecting to increase usability over the *gaze and click* gesture. We discuss this (false) intuition in further detail in Section 5.8.

5.5.3 User Experience

As shown in Figure 5.4, despite the seemingly large number of protocol steps, *HoloPair* users are required to perform only two manual steps in order to securely pair their MR headsets. The instructions are as follows:

Instructions for the user in the role $\mathbf{U}_A$:

1. Once you see $\mathbf{U}_B$ waving, gaze and click on them.
2. Watch as $\mathbf{U}_B$ moves their finger along the shown shape. If their hand movement follows the shape, gaze and click on them. Otherwise say *"Abort"*.

Instructions for the user in the role $\mathbf{U}_B$:

1. Wave towards $\mathbf{U}_A$.
2. Move your finger along the shape shown in front of you, starting from number 1.

As the usability evaluation in the next section shows, the manual steps are based on natural gesture that users learn quickly. While a few initially needed to practice the *gaze and click* gesture, this is a standard primitive that MR headset users are likely to already have mastered before starting to use *HoloPair*. However, the manual behavior that *HoloPair* introduces, namely following the shape with their finger, was naturally and easily performed by all protocol participants.

5.6 Experimental Evaluation

We now experimentally evaluate the *HoloPair* system with respect to average pairing time, users' ability to detect man-in-the-middle attempts, and subjective usability, measured with a questionnaire. Additionally, we provide data on computational performance of the developed *HoloPair* prototype.

5.6.1 Pilot User Study

While designing and prototyping the *HoloPair* system, we implemented three different versions of the shared secret confirmation step (shown in Figure 5.9), and we chose the most suitable one after running a small-scale pilot user study.

In the pilot user study, 6 participants were instructed to repeatedly pair their headsets using the supported schemes in both roles, and were afterwards asked to state which scheme they found the most usable. Somewhat surprisingly, all participants indicated that they preferred the version of the shared secret confirmation step in which $\mathbf{U}_B$ is required to follow the shape generated in space with their hand (version **(c)** in Figure 5.9). Besides being subjectively the most usable, this scheme conveniently supports the highest theoretical entropy of the weak-hash ($(10 \times 10)^N$), and was also the easiest to explain to pilot-study participants.

Consequently, we decided to fully focus the main user study, which consisted of 22 new participants, on evaluating the performance, security, and usability of this variant of the shared secret confirmation.

5.6.2 Main User Study

In our main user study, we invited a total of 22 participants to experimentally use the system.

Demographics. We recruited a total of 22 participants (14 female, 18 male, aged from 21 to 33) using mailing lists and social media posts. None of the study participants owned a HoloLens device nor had extensive prior experience using MR headsets.

Setup. The experimental setup consisted of two Microsoft HoloLens devices that were connected on the same wireless network. The experimental evaluation took place in a shared office space with the area of approximately 50 m^2.

Participants were invited to the study in pairs, and were not grouped by any specific criteria except times at which they were available to participate in the experiment. Upon arriving, participants were provided a written explanation of the goals of the study and given the chance to ask questions. They were then asked to sign the consent form which stated that the data anonymously captured during the experiment will be used for publication. Given that no personally identifiable information was captured in the experiment and that all data was stored anonymously, the institution that hosted the experiments did not require an institutional review board approval for these kinds of studies.

The experiment introduction explained the reasoning behind the need to securely pair mixed reality headsets, the envisioned usage scenario and the potential attacks that can happen during the process. Participants were told that their goal will be to repeat the pairing attempt several times, and that an attack might be simulate during some of the pairing attempts.

Procedure. Not having any previous experience with AR, study participants were first given the opportunity to get accustomed with using a mixed reality headset: specifically using gesture recognition (*click* to confirm a successful pairing), and voice recognition (say *"Abort"*when an attack is detected).

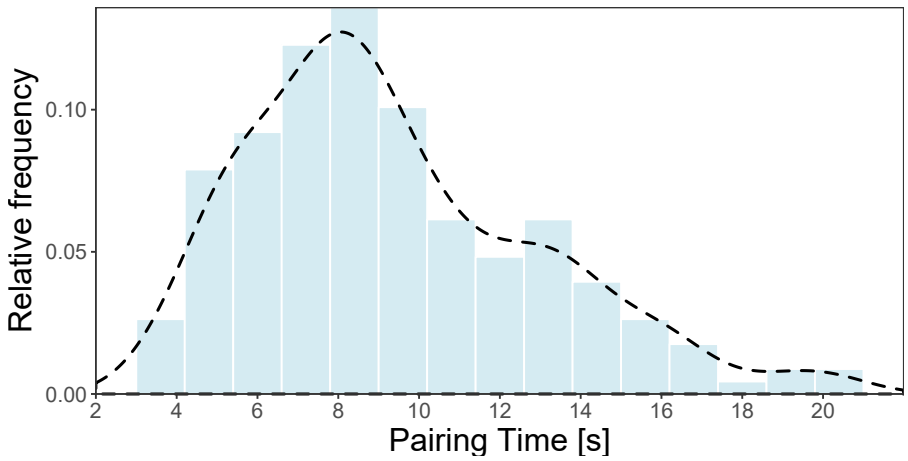


Figure 5.5: Relative frequencies of all pairing times in the main user study when no attack was simulated. Pairing times are computed from the moment from when $\mathbf{U}_A$ confirms that user $\mathbf{U}_B$ has waved until $\mathbf{U}_A$ has either confirmed a successful pairing or one of them said *"Abort"*.

Each pair of participants was asked to perform a minimum of 10 pairing attempts, after which the protocol roles ($\mathbf{U}_A$, $\mathbf{U}_B$) were switched and participants performed at least 10 additional pairing attempts.

The experiments measured the impact of two independent variables:

(1) Attack Simulation. The main need for user involvement in the *HoloPair* system is to detect potential man-in-the-middle attacks, which are evident by a mismatch of independently generated weak-hashes on two headsets. In order to evaluate users' ability to detect such attacks, we simulated differing shapes being shown to participants in randomly chosen 20% of the pairing attempts.

(2) N, Shape Complexity. In order to evaluate the impact of the complexity of the shared secret confirmation step on the dependent variables (total time and success rates), we varied the value of N, the number of shape segments. In each pairing attempt, N is randomly chosen from the set $\{4, 6, 8\}$.

Measured Data. After having a total of 22 participants take part in the main study, we gathered data on the execution of a total of 230 pairing attempts, out of which a man-in-the-middle attack was simulated in 44 cases.

For each pairing attempt, we measured two sets of dependent variables: **(1)** timestamps at which users entered each step of the pairing protocol, and **(2)** whether

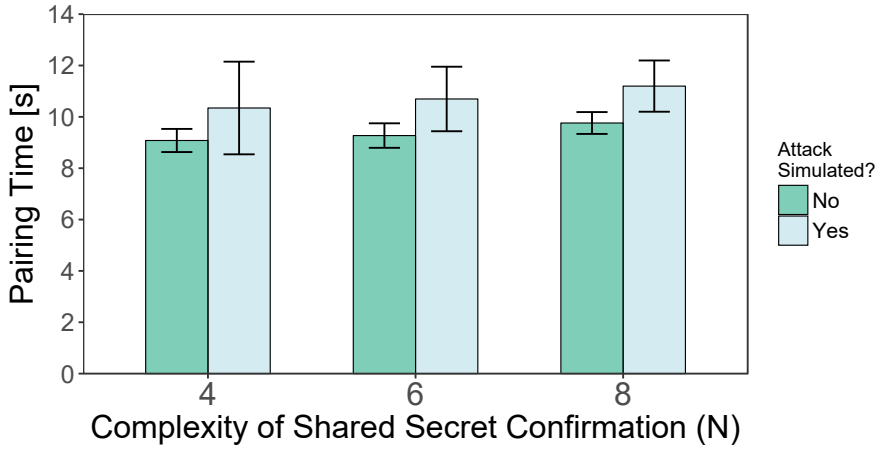


Figure 5.6: Average pairing times as the function of the complexity of the shared secret confirmation step, N. As N changes from 4 to 8, the average pairing time increases, but only slightly. This indicates that the total time spent performing the gesture does not depend on its complexity as much as it does on other user behavior, such as waiting, making the decision, or *clicking*.

the pairing attempt was successful, either by detecting a potential attack (when it was simulated), or correctly exchanging the shared secret (when no attack simulation took place).

Results: Pairing Time. Figure 5.5 shows the relative distribution of the total times for all pairing attempts in our main user study, measured from the moment when $\mathbf{U}_A$ reveals its commitment on a specific instance of the weak-hash. The median pairing time for users is **between 8 and 9 seconds**, while 80% of successful pairing attempts finish in less than 13 seconds. These times are comparable or lower than previously reported confirmation times for similar device pairing schemes [216, 153], in which device pairing for two users took between 5.7 seconds to only perform a hand gesture and 20 seconds to perform the whole protocol. Furthermore, considering that this process needs to be repeated only once for each new pair or devices, as we show in the remainder of this section, the majority of study participants found the System both usable and sufficiently fast.

It is interesting to note that we observed longer average pairing times in the case when users decided to abort the pairing execution, as shown in Figure 5.6. This is likely due to two reasons. Firstly, after observing that the gesture did not match

the expectation, participants often repeat it before deciding to abort. Secondly, we observed that, despite the simplicity of the voice command (*"Abort"*), their instruction was sometimes not recognized by the device on the first attempt, which increased the time before the device recorded that a decision was made.

However, we emphasize that in the case of attacks, the time until users make an ultimate decision is of significantly smaller influence. It is much more important that participants in our study detected potential attacks with high success rates, which we discuss next.

Results: Success Rates of Pairing and Attack Detection. Previous research has shown that users are often inattentive, do not understand the risks, or simply proceed without even trying to verify the exchanged shared secrets [214]. Since we simulate attack attempts in a percentage of pairing runs, we are able to estimate the likelihood of a successful man-in-the-middle attack even when the two weak-hashes do not collide, but users fail to recognize this. We consider a pairing attempt successful if $\mathbf{U}_A$ confirmed that shapes match when there is no attack simulation, or if the same participant called *"Abort"*in the case where an attack was indeed simulated.

The success rates that our system achieved in our user study are highly encouraging: our results show that **91% of simulated attacks (43/47) were detected** by the study participants. While there numbers are high, it is important to note the possibility that study participants in general are generally more vigilant by the virtue of being measured and performing a novel interaction. However, even though participants might become less attentive as they get accustomed to using the *HoloPair* system, we note here that the measured success rate is comparable or better than the results achieved in studies which similarly tested user's ability to compare different short strings, hashes, or pictures [210, 214].

An even higher success rate was achieved in the case where there was no attack, where **98% of pairing attempts were successful (181/186)**, with only 4 false aborts when both weak-hashes did indeed match. These results are comparable with the previously reported results for mobile device pairing [216, 153], in which users were able to successfully pair in more than 90% of the cases when no attack was

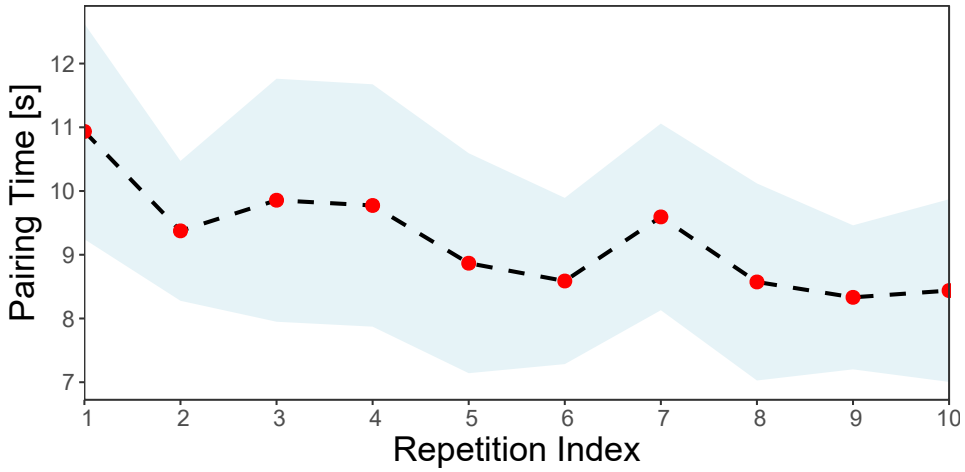


Figure 5.7: Impact of learning on the mean pairing times. As users repeat the same procedure multiple times, their pairing times decrease by about 20%, indicating that users become more accustomed to the gestures over time.

simulated. This further confirms that relying on precisely located holograms that are independently shown to both participants and using gestures to validate that their shapes indeed match allows for confirmation of fairly high-entropy information.

Results: Impact of the Shared Hologram Complexity (N). We now look into how the average mixed reality headset pairing times depend on N, the complexity of the secret shared hologram.

As shown in Figure 5.6, increasing N does expectedly increase the average required time for two users to pair their headsets, but only to an extent that is within 1 second. The small difference is visible both in the case of attack simulations (light green) and in the case when no attack was simulated (green).

The small difference in pairing times indicates that users spent the majority of pairing in other behavior, such as waiting, deciding, or inputting the decision into their own device (confirming via a *click* gesture, or aborting by saying *"Abort"*).

Results: Learning Effects.

Finally, we analyze the extent to which users learn to use the *HoloPair* system more efficiently with repetition of the pairing procedure. Figure 5.7 shows the mean and standard deviation of all successful pairing attempts in our study, grouped by

their session index. All pairing attempts in which one participant takes the of $\mathbf{U}_A$ are considered a single session, and we thus have two sessions per participant pair.

As expected, the mean pairing time reduces as users repeat the procedure multiple times, from about 11 seconds in their first measured attempt, to less then 9 seconds on their 10th attempts (we do not take into account the "practice" attempts here). These results suggest that despite their initial inexperience with using MR headsets, participants indeed quickly become accustomed with the *HoloPair* system, as they also indicated in the usability questionnaire (Q7), which we discuss next.

5.6.3 Usability Questionnaire

After participants performed multiple pairing attempts in both roles, we assessed the usability and user perception towards the *HoloPair* system and the implemented prototype by asking them to complete the System Usability Scale [151] (SUS). SUS is a simple and general questionnaire designed to quickly evaluate the usability of a broad range of systems [217]. We chose to use SUS given the lack of previous work in device pairing for mixed reality headsets and the resulting lack of previously used evaluation methods that the developed prototype could be compared with. We therefore hope that the evaluation results presented here would be used as a basis for future comparison with systems who tackle the same challenge.

The questionnaire consists of 10 statements such as *"Q6: I thought there was too much inconsistency in this system"* or *"Q9: I felt confident using the system"* that users of the system grade on a Likert scale (1 - Strongly disagree, 5 - Strongly agree). The full list of questions is provided in the Appendix A.2.

Results: Usability Questionnaire. We show the overall results of the SUS questionnaire in Figure 5.8. None of the study participants thought that they had to learn a lot before they could get going with the system (Q10), and while some felt they would need a help of a technical person (Q4, 5%), most users generally found the system easy to use (Q3, 95%) and believed others would learn to use the system very quickly (Q7, 91%).

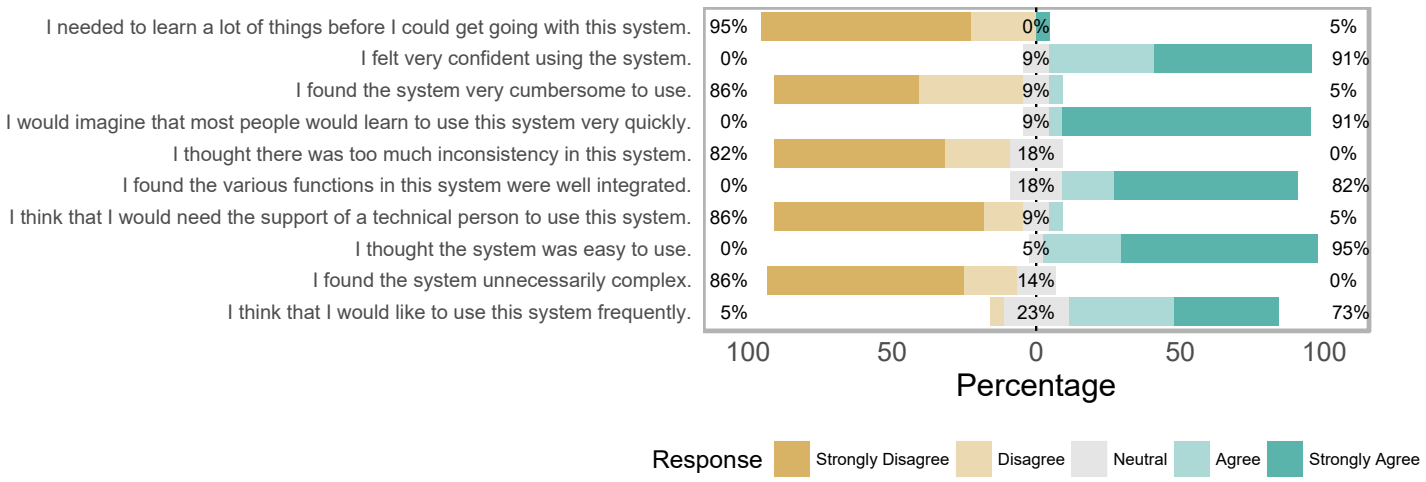
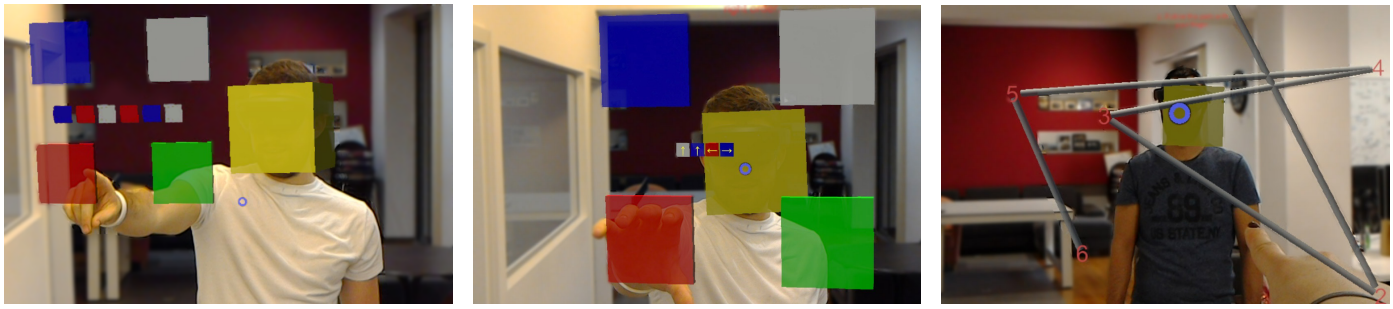


Figure 5.8: Participants' responses to the SUS questionnaire show a high average SUS score of 86.4. None of the participants thought that they had to learn a lot before they could get going with the system (Q10, 5%). While some felt they would need a help of a technical person (Q4, 5%), most generally found the system easy to use (Q3, 95%) and believed others would learn to use the system very quickly (Q7, 91%).

While some users believed that they would need support of a technical person to be able to use the system (Q4, 5%), this is likely due to the fact that learning how to use the mixed reality gestures did indeed require initial help from the experimenters, but was quickly grasped by the participants. This learning effect is as also visible in Figure 5.7, showing that the average duration for a single pairing attempt reduced with each new try. Furthermore, once users get accustomed to such gestures by using other Microsoft Holographic applications, they are likely to not need any help to start using the *HoloPair* system. We leave this hypothesis for future work, which should be evaluated once more members of the general public become accustomed with mixed reality systems.

Based on the total of 22 participants who completed the questionnaire, none of which have had extensive previous experience using HoloLens, the *HoloPair* system achieves an **average overall SUS score of 86.9**. Previous research on interpreting individual scores concluded that the mean SUS score of 85 translates into users' adjective rating of "Excellent" [217]. Consequently, we conclude that the majority of the study participants found the usability of the *HoloPair* system to be well above the average.



(a) *"Touch the larger cubes according to the color sequence of length N."*

(b) *"Touch the cube and then swipe in one of 4 directions as indicated by the arrows."*

(c) *Follow the shape with your hand."*

Figure 5.9: Three different versions of the shared secret confirmation step that we use in this chapter. In (a), U_A confirms that U_B is touching the correct sequence of colored cubes. In (b), after touching any of the cubes, U_B is required to also swipe their hand in one of the 4 directions (depicted by arrows). Finally, in (c), U_B follows the 3D shape in front of him with his hand, while U_A confirms that the shape shown on his D_A matches the hand movement (and correspondingly the shape on D_B).

5.6.4 Prototype Performance

Table 5.1 provides measurements for the total, average and maximum/minimum values of the following measurements: battery load, RAM, CPU and GPU load, and network bandwidth.

The maximal increase in GPU, RAM, and CPU load is bellow 15% of their maximal amounts, while the total network bandwidth for the full execution of the pairing protocol is about 700kB. Such small performance footprint is expected, considering the small number of messages that our system actually exchanges over the wireless network, and contrasting this with the need to exchange precise spatial data in the process of finding a *world anchor* for hologram sharing.

While performance was not one of our considerations during the development of the *HoloPair* prototype, and thus the measured values could likely be further improved, we conclude that the current prototype already achieves a low overall impact on the existing HoloLens device.

5.7 Alternative Shared Secret Confirmation Steps

HoloPair does not rely on any specific form of the shared secret confirmation step. Consequently, in our research we designed, developed, and evaluated three different versions of this confirmation step, shown in Figure 5.9. We now describe the initial two schemes that were tested, before deciding to focus on the final scheme (based on *virtual pipes*), that was described throughout the chapter.

(a) - "Cubes": Our initial design choice was based on the idea of having a shared keyboard that one user would touch, while the other observes the gestures and verifies that they are correct. As shown in Figure 5.9a, in order to ensure a simple user experience, $\mathbf{U}_B$ is required to look at the sequence of colors shown at the bottom of their screen and accordingly touch one of the four larger cubes to communicate the value of the weak-hash generated on his headset. At the same time, $\mathbf{U}_A$'s device independently generates the sequence of N and shows the four larger cubes at the same location as $\mathbf{U}_B$'s device. $\mathbf{U}_A$ observes if $\mathbf{U}_B$'s sequence of colors indeed match, and thus verifies high probability that the exchanged public keys are indeed authentic.

This results in the total number of different configurations that can be presented with N colors to be 4^N, and correspondingly, the expected probability of successfully guessing an alternative pair of public keys for an attacker to be:

$$P_a(N) = \frac{1}{4^N}$$

(b) - "Cubes with Arrows": In an extension of the first variant, shown in Figure 5.9b, the weak-hash that is independently generated on both devices also specifies one of 4 directions for each of the colors in the sequence. Correspondingly, $\mathbf{U}_B$ is required not only to touch the larger cube of the same color, but also to make a hand movement in the direction indicated on his sequence. The reasoning behind this version is that additional hand movement does not significantly impact the usability

of the secret confirmation step, while it, at the same time, squares the number of possible weak hashes that users can communicate using a sequence of length N.

As $\mathbf{U}_B$ is additionally required to move their hand in one of the four directions, this further increases the number entropy of the weak-hash to $(4 \times 4)^N$ and decreases the probability of a successful attack to:

$$P_b(N) = \frac{1}{(4 \times 4)^N} = \frac{1}{16^N}$$

5.8 Discussion

Automating the Confirmation Step. In this chapter, we rely on $\mathbf{U}_A$ to verify that the observed gesture made by $\mathbf{U}_B$ indeed matches the expectation based on the H_{W_A}. This step could in future be automated by incorporating a gesture recognition system that would be able to track the precise location of $\mathbf{U}_B$'s hands. However, it is important to note that the current system also relies on the inherent human ability to detect anomalies and e.g. follow only the legitimate user, while detecting or ignoring any adversary's attempt to inject into the visual out-of-band channel. We thus leave this possibility for future work, together with the challenge of designing user gestures that would be particularly suitable for automated verification.

Designing for MR. While implementing the *HoloPair* prototype, we evaluated several design choices that had a large impact on the usability of the system. Besides the (incorrect) intuition that using a voice command to abort the protocol run would be more convenient, the largest usability improvement came as a the result of using only mixed-reality holograms to display information. Despite the initial expectation that important messages and objects would be best visible if always visible in a form of a Heads-Up-Display (HUD), reading information shown as real, mixed-reality objects with a fixed location in the environment proved to be significantly more natural. This is likely due to the fact that such holograms can be approached when needed, quickly and naturally glanced, or otherwise ignored.

Table 5.1: Performance impact of the *HoloPair* prototype

Max δ GPU load	7% (from 22% idle)
Max δ CPU load	15% (from 36% idle)
Max RAM load	225 MB
Max δ Energy load	10% (from 50% idle)
Total network bandwidth	700 kB
Total application size	130 MB

Given that is one of the core new capabilities of the HoloLens device in comparison to previous MR devices that do not support precise *world anchoring*, we here emphasize the importance of re-evaluating existing design practices for various security primitives as they are being implemented in mixed reality.

5.9 Related Work

General Device Pairing. Comprehensive overviews of a many different device pairing methods and their usability evaluations can be found in [210, 218]. For a recent work that extensively surveyed multiple shared secret confirmation steps for mobile devices and evaluated user's ability to detect potential attacks, see [214]. However, given that neither the two users is able to observe the output of both headsets, we are unable to directly apply previous work to the MR headset scenario. Secondly, in contrast to previous work in which users are required to e.g. copy some value from one device to the other, we make the assumption that the adversary can fully eavesdrop even the out-of-band channel. Considering the proliferation of mixed reality headsets and the fact that each of them has multiple front-facing cameras, we believe this to be a necessary assumption.

Finally, despite previous proposals to e.g. shake mobile phones simultaneously and distribute the shared secrets this way [219], there is no obvious way to expose two MR headsets to the same outside conditions that a co-located adversary could not easily copy. Given that even a simple comparison of visual outputs from two

headsets is non-trivial, the problem of pairing MR headsets is likely to become an active topic of future security and usability research.

Secure pairing of MR headsets. Given the novelty of the MR headsets and the very recent availability of mixed reality headsets, the topic of MR pairing has not yet been extensively explored, with only a single related paper with the same focus [110]. However, the authors take a significantly different approach, by building their own hardware prototype which assumes that future MR headsets will have the support (multiple antennas) to perform precise wireless localization. To the best of our knowledge, this chapter presents the first security-focused research that uses the novel capabilities of Microsoft's HoloLens device to achieve usable and secure pairing of two MR headsets. Moreover, this work is the first proposes for a practically achievable MR headset pairing protocol that assumes only the existing capabilities of the HoloLens device.

5.10 Summary

In this chapter we proposed *HoloPair*, protocol and a system for secure and usable pairing of mixed reality headsets. We built a working prototype implementation of *HoloPair* using two Microsoft HoloLens headsets. By running a user study with a total of N=22 participants, we evaluated the feasibility of the proposed protocol in terms of security guarantees, usability, and prototype performance.

The experimental evaluation of the *HoloPair* prototype shows that participants with little or no prior experience using MR headsets are able to achieve high rates of detecting attack simulations or successfully pairing when no attack is simulated. Furthermore, the system is highly usable, as evident by short pairing times and high average scores achieved on the usability questionnaire, while having low computational requirements.

As direct device pairing is at the core of many current and future mixed reality applications, the ability for developers to create secure connections between their users' devices is of crucial importance. As the number of devices that support the

WMR platform grows, the developer community will likely continue publishing new applications that rely on shared mixed reality experiences with increasing speed. In order to ensure that security concerns are considered from early stages of the growth of the developer community, we've made the full implementation available to the public and started the process to include the source code in Microsoft's official HoloLens-Unity repository.

Given the lack of practical proposals or implementations of secure device pairing despite the rapid growth of MR platforms, we believe that the research presented in this chapter is an important step towards making future MR interactions secure and private from the start.

Trust, but verify.

— Proverb

6

Continuous Input Authentication by Visual Supervision of Clients

Contents

Finally, in this chapter we show how the capabilities of mixed reality technologies can also be used to secure interactions with legacy systems, even when assuming the stronger adversary model that we use in this thesis. The research presented here is the first to propose using the front-facing cameras of mixed reality systems for visual supervision of other devices' screens during normal user input. Due to having a higher resolution camera and stronger processing capabilities than currently available mixed reality headsets, in this chapter we investigate this general concept by implementing a prototype on a recent smartphone. However, the core concepts that we propose and evaluate in this chapter can be readily translated to mixed reality headsets.

6.1 Motivation

When connecting to a remote server using their local clients, such as their laptops, users typically authenticate at the beginning of each session. If the authentication is successful, the remote server subsequently considers all of the client's requests in this session as authenticated and authorized by the user. However, as Figure 6.1 shows, if the client is under the adversary's control, the link between what the user intends to input ("AB"), what they see on the screen after input ("AB"), and what the server actually receives ("XY") is broken.

As we argued in Chapter 2, such compromise of the local client is a realistic assumption that must be considered during system design, especially for sensitive applications such as online banking [73], configuring industrial [76] and medical devices [77], or accessing private customer data [74].

As an example, online banking systems assume the possibility of compromise and thus usually mandate that transaction data is additionally authenticated by the user. A common approach is to require that users key in several digits of the sensitive fields (such as the beneficiary's account number) into a dedicated hardware token

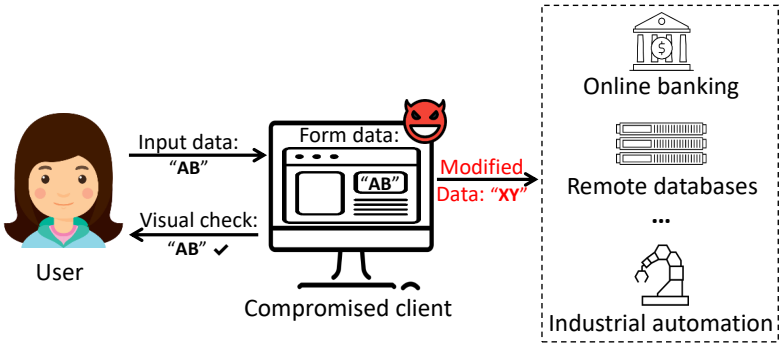


Figure 6.1: Motivating scenario. The user communicates with various remote services, such as online banking, remote databases, or industrial device configuration by inputting data via the compromised local client. Despite correctly displaying user's input on the screen ("AB"), the adversary arbitrarily modifies its value to "XY" as it is sent to the remote service.

or a smartphone application. This device serves as a second factor to authenticate sensitive data and generates a Transaction Authorization Number (TAN), which the user inputs in their client. Finally, the server verifies that each request coming from the client includes a matching TAN matching the transaction details. Therefore, even in the case of a full client compromise, such verification ensures that the adversary is unable to generate or modify fields of the user input that are protected by TAN.

In this chapter, we are inspired by the general idea of using TAN-like methods; our goal is to extend this concept to support authentication of all data that the user inputs into a specific form. However, if all data is to be authenticated, approaches that require explicit user input on the second-factor device do not scale well, since they lead to duplication of effort and disturb the normal user workflow.

Extracting user intent. We build upon the core idea that, during interaction with a large class of networked services, the user's intended input is visually displayed on the client's screen (e.g., in online banking or remote database access). Despite what a compromised client might attempt to do in the background, users communicate their intention by inputting and modifying the values shown on the screen until they are satisfied with what they see, or abort if they are prevented from doing so. Consequently, by extracting the contents of the client's screen during normal user input, a device that serves as a second factor can infer the values that the legitimate user intends to submit to the server without requiring user's explicit

input. This further allows it to generate a TAN-like signature, which serves as a *proof-of-user's-intent* (POI) [220]. If the remote service requires to independently receive such proof-of-intent along each request, the adversary is prevented from either generating arbitrary user input or from modifying the input provided by the user despite controlling the client.

As we discuss in detail in Section 6.10, the general idea of supervising user input on an untrusted client has been researched before. However, previous work relied on the assumption that the client is only partially compromised by assuming the existence of either a trusted virtual machine [220], an operating system [221], or an *attester* application [222].

Visual supervision of user input. Capabilities of recent smartphones are becoming sufficient to allow continuous, real-time analysis of high-resolution camera feeds. This motivates us to research the novel approach of *visual supervision of compromised clients*. In this approach, a camera-equipped device (a smartphone, or a mixed reality headset) acts as the second factor that analyses the client's screen to seamlessly extract and authenticate users' input during interaction with remote services and force the client to behave honestly.

While seemingly simple, successful implementation of visual supervision of user input requires that multiple technical and research challenges are carefully addressed. These range from evaluating the technological readiness of existing smartphones for continuous Optical Character Resolution (OCR) of another computer screen, to preventing concurrent on-screen data manipulation during user input, and detecting and preventing attacks that manipulate users into inputting data under different semantics.

As a first step towards evaluating the present-day feasibility of visual supervision of electronic devices, in this chapter we describe, build, and experimentally evaluate *IntegriScreen*, a prototype system that implements the introduced concepts. *IntegriScreen* visually analyzes the clients' screen and relaying the extracted information as a visual proof-of-intent to the remote server, thus preventing any dishonest behavior from the client, while leaving user workflow largely unchanged.

Main contributions. In summary, this chapter makes the following main contributions:

1. **System design.** We propose and describe *IntegriScreen*, a system that protects the integrity of the user's input to a remote server by using a smartphone to visually supervise the user's interaction with an untrusted client.

2. **Advanced UI attacks.** We discuss and prevent various attacks that the adversary can attempt: from simply modifying the data on the screen after user input, to manipulating the user into submitting data under a different context.

3. **Prototype & experimental evaluation.** In order to evaluate the feasibility of the approach on today's smartphones, we build a fully functional prototype of the *IntegriScreen* system and test it with three different devices against a range of automated attacks.

4. **User study.** Finally, we run a user study with 15 participants, in which we measure their ability to detect and prevent attacks with *IntegriScreen*, as well as evaluate the usability of the developed prototype.

6.2 Assumptions and Goals

We start by stating the assumptions about the system and the adversary, as well as the design goals that the proposed system should achieve.

6.2.1 System Model

Figure 6.2 shows an overview of *IntegriScreen*. The user uses a local client to provide input to a remote server (service). We assume that the remote server exposes a form-based configuration page that is accessible via a web browser running on the client system. As such, all the data whose integrity needs to be protected is presented on the client's screen. The input can come from a keyboard or a mouse.

The user's interaction with the local client is visually supervised by an application on her smartphone, which is statically placed such that it can continuously capture

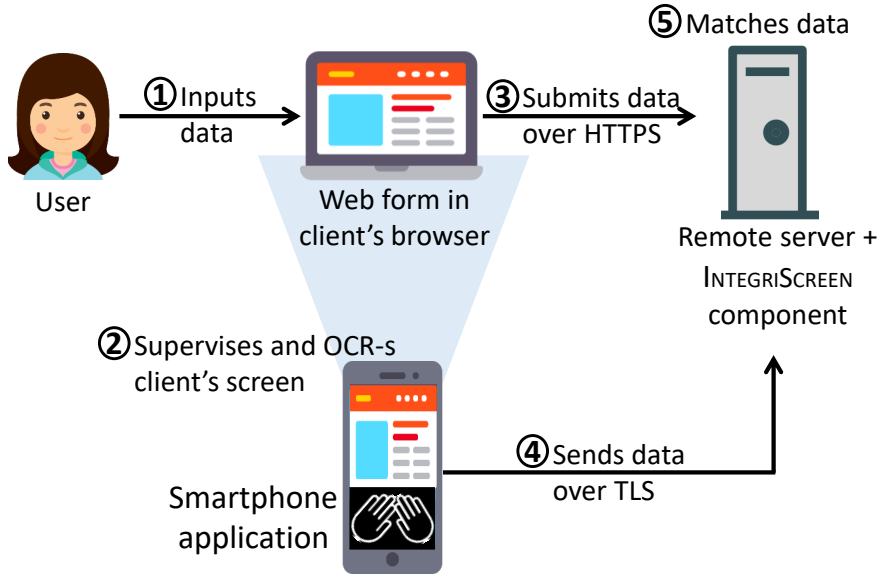


Figure 6.2: System overview. As the user is inputting data into a web form on an untrusted client to communicate with a remote server, the smartphone's camera captures the interaction. Upon submitting the data, there are thus two independent channels between the user and the remote web server: a standard `HTTPS` channel through the client's browser, and a `TLS` channel through the smartphone application. The server uses the second channel to verify the authenticity of the values that were received through the first channel.

the contents of the client screen during input. Due to limitations of smartphone input, the user should not be required to input any data through the smartphone.

6.2.2 Adversary Model

We assume that the client system is fully compromised. The adversary has unrestricted physical access to the client system and can freely modify it. The adversary also actively controls the network; he can drop, delay, replay, or generate arbitrary network level messages.

However, we assume that the smartphone is not compromised; only the legitimate user can unlock the smartphone and run applications on it. The remote server is assumed to be honest and trusted by the user.

The adversary's goal is to achieve that the remote server accepts a request that does not correspond with the legitimate user's intended input. We consider protecting the privacy of user data, stealing user's credentials (e.g. via keylogging)

or data exfiltration to be out of the scope of this work. We also consider denial of service attacks to be out of scope of this work.

6.2.3 Design goals

Based on the discussion from the previous section, we now state the design goals that the required solution must achieve. The solution must:

- Authenticate remote requests that users make through compromised clients, i.e., ensure that the adversary can neither generate nor modify existing remote requests successfully.
- Ensure that users are not being manipulated into inputting and submitting data that they would not submit in the absence of the adversary.
- Require minimal added interaction in the absence of attacks: do not require that users input or explicitly verify any data except the data they are inputting on the client device.

6.3 Visual Supervision for User Input Authentication

We start by providing the general idea of visual supervision of user input and discuss challenges that need to be successfully addressed in order to ensure that interactions with the remote server match the user's intentions.

6.3.1 Approach Overview

The overall approach of the *IntegriScreen* system is shown in Figure 6.2. The system consists of 3 main components: 1) the web form and its code, running on an untrusted local client, 2) the trusted remote server with a *IntegriScreen* web server component, and 3) the mobile application, running on the smartphone under the user's control.

The general flow of the user's interaction with *IntegriScreen* is shown in Figure 6.2 and described below:

① The user inputs the data through the form running in the untrusted client's browser.

② The smartphone application extracts the data that is input by the user by executing optical character recognition (OCR) of the client system's screen in order to generate a visual proof-of-intent.

③ The browser transfers the user's input over a `HTTPS` channel to the remote server.

④ The smartphone application then transfers the generated proof-of-intent to the remote server over a dedicated `TLS` channel between the smartphone and the remote server.

⑤ Upon receiving data from both the browser and the mobile device, the *IntegriScreen* server component matches the data from two inputs, as shown in Figure 6.3. If the two inputs match exactly, the web server accepts the input; otherwise, it rejects it.

Security of the approach. While relatively simple, the described design of *IntegriScreen* already protects against the majority of attacks discussed in Section 6.1. Even after compromising the client and obtaining the victim's authentication credentials, the adversary can neither generate new requests, nor covertly modify the data before submitting. This is prevented by the server, which would reject the requests due to not having received a matching proof-of-intent.

6.3.2 Remaining Challenges

There remain, however, several other challenges that need to be addressed in order to ensure that all received requests truly correspond with the user's intended input. We first summarize all the research challenges:

Preventing UI manipulation attacks. Even if the adversary is prevented from modifying the data in transit between the user and the server, fully controlling the client device still allows him to manipulate the user to achieve his malicious goals as

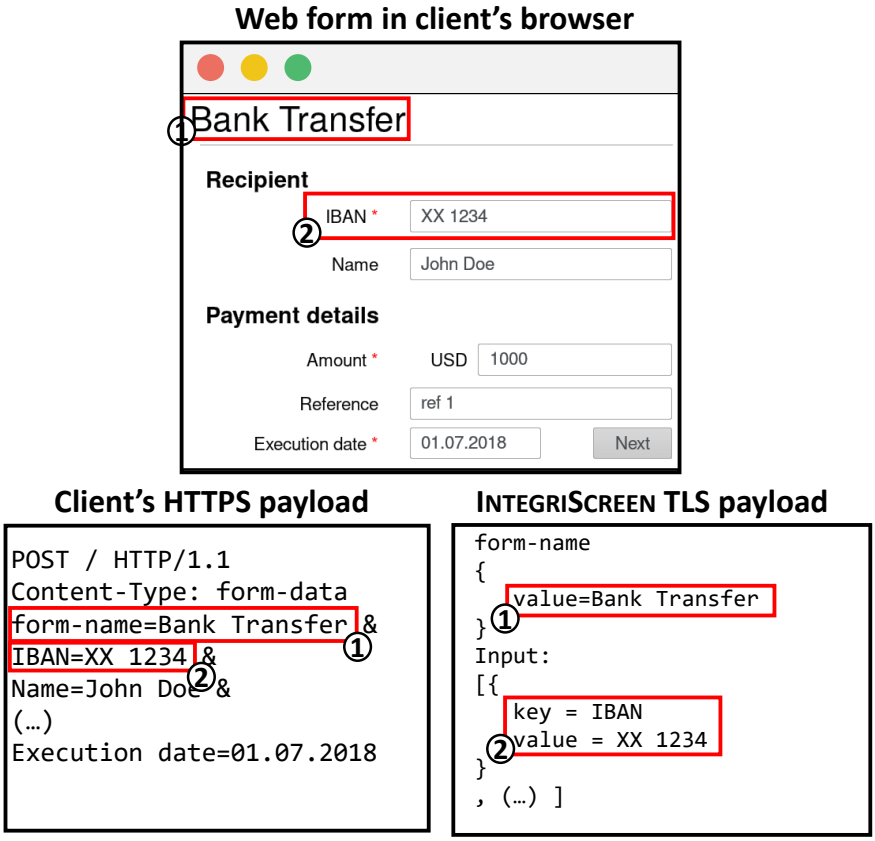


Figure 6.3: ***IntegriScreen*** **input data matching. The figure shows the input data matching mechanism in the** ***IntegriScreen*** **server using the bank transfer example illustrated in Figure 6.5. The** ***IntegriScreen*** **server receives data from two channels: 1) the `HTTPS` channel from the client, and ii) the TLS channel from the** ***IntegriScreen*** **mobile application. ① and ② annotate a subset of the data that are part of the payload, namely the name of the specific web form and the IBAN value.**

he controls the initial step of the process: loading and presenting the user interface shown in the browser. Such an adversary can arbitrarily modify all elements of the presented interface and thus achieve that, while the victim indeed inputs all the data, and the data reaches the remote service unchanged, its actual semantics differ between the user and the remote server. For example, in case of remote configuration of medical devices, changing a single text label in the user interface (or even only its relative position!) can result in an anesthesiologist inputting a value using a wrong measurement unit, e.g. ml instead of dl, and seriously harming the patient due to a tenfold increase in the drug dosage.

As we discuss in Section 6.4, *IntegriScreen* prevents such attacks by requiring that the remote server provides a specification of the user interface (UI) and

by ensuring that the UI loaded on the client's screen matches its specification during all stages of user input.

Preventing on-screen data modification attacks. So far, there was no explicit discussion of the moment at which mobile application captures the data shown of the screen. Even if the data shown on the client's screen is truthfully sent to the remote server, the integrity of user input is not necessarily guaranteed if data extraction happens only before the form is submitted from the client. Despite the user inputting the intended data, the adversary can subsequently modify the content of the screen in such a way that the user does not notice the change, but the smartphone application later accepts it, and the server thus receives the same maliciously modified data from both channels.

For example, for the web form shown in Figure 6.3, while the user is focused on inputting the payment amount or execution date, the adversary modifies the previously input IBAN (International Bank Account Number), without the user noticing. Furthermore, the adversary could aim to modify the values shown on the screen while the user is absent, not focused on the screen, or a malicious window temporarily overlays the browser form to shift the victim's attention.

As we discuss in the next section, a crucial extension of the proposed architecture is to include real-time, continuous supervision of the screen content, impose specific expectations about the design and behavior of the user interface, and only allow the mobile application's proof-of-intent to be submitted if the data has been generated in accordance to these rules.

Challenges of visually supervising another device. While humans – arguably only after years of experience – successfully understand and interact with modern computer interfaces – despite their ever-changing, platform-specific intricacies – computers still face significant challenges for similar tasks. For example, even a seemingly straightforward task of detecting computer screens in images is still an open research question [27]. It is therefore an interesting challenge to propose a

system that requires *minimal changes to the user interface*, while at the same time achieving security guarantees against the adversarial client system.

Furthermore, despite recent significant improvements in Optical character recognition (OCR) based on deep learning [223], achieving *consistent continuous detection of textual content* on another device's screen requires several deliberate design choices. Using OCR libraries naively, e.g. attempting to detect all text shown on a large part of the client's screen, results both in low performance (< 0.5 fps) and significant parts of text not being detected due to different font sizes and types.

Finally, continuous visual supervision of another device's screen requires that the mobile device is statically positioned so that its camera captures the whole area of interest. This, however, means that the supervised screen is captured from different positions and different angles and requires *precise estimation and removal of the pose* between the two devices.

6.4 *IntegriScreen*: System Architecture

We now describe in detail how *IntegriScreen* addresses the remaining challenges discussed in the previous section to ensure that all remote requests truly correspond to a legitimate user's intended input.

6.4.1 Verifying the Integrity of the User Interface

A small change in a single label's relative position can allow the adversary to alter the semantics of user input, e.g. by moving the label with the instruction *"input the value in mm"* closer to another input element, and thus manipulate the victim into submitting data in accordance with his malicious goals. In order to protect the user from such UI manipulation attacks, it is therefore crucial to assert that the loaded interface closely matches its specification throughout for the whole duration of the user interaction: not just that the textual values, but also of all the relative positions of UI elements have not been modified.

We now discuss a minimal set of design guidelines that standardize the web form's behavior in order to harden it against such attacks. By following these guidelines, a

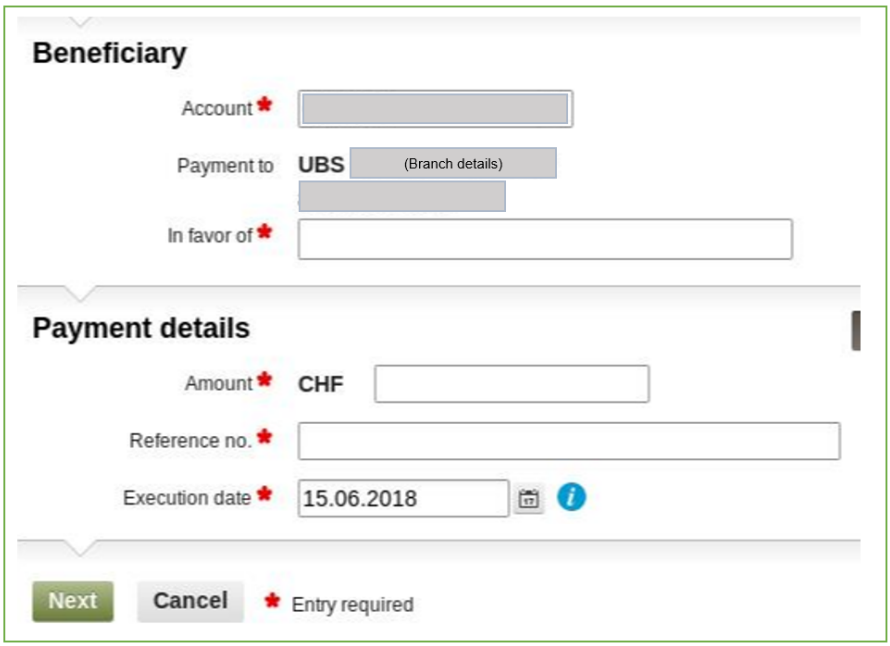


Figure 6.4: Real-world motivation for the running example. The design of the running form used throughout the chapter and in the user study is based on the web banking interface of one of the large financial organizations.

web form can be precisely described in a simple *JSON form specification* file, such as the one shown in Listing 6.1, which the smartphone application requests from the server, parses, and uses as a guide during supervision of user input.

Figure 6.5 shows the result of hardening the web form that was used as the running example in Figure 6.3:

① **Form Border.** *IntegriScreen* relies on the web form being surrounded by a visible boundary, which serves two roles. First, it helps the smartphone application to estimate and remove the pose between the screen and the mobile device for different spatial arrangements. Second, the border also limits the area of visual analysis of the screen by excluding other OS/browser UI elements and thus increasing the user's privacy. For simplicity, the current prototype uses a solid green color to indicate the form border. However, this can be fully customized as long as the four corners that indicate the protected area can be detected and tracked by the mobile app [224].

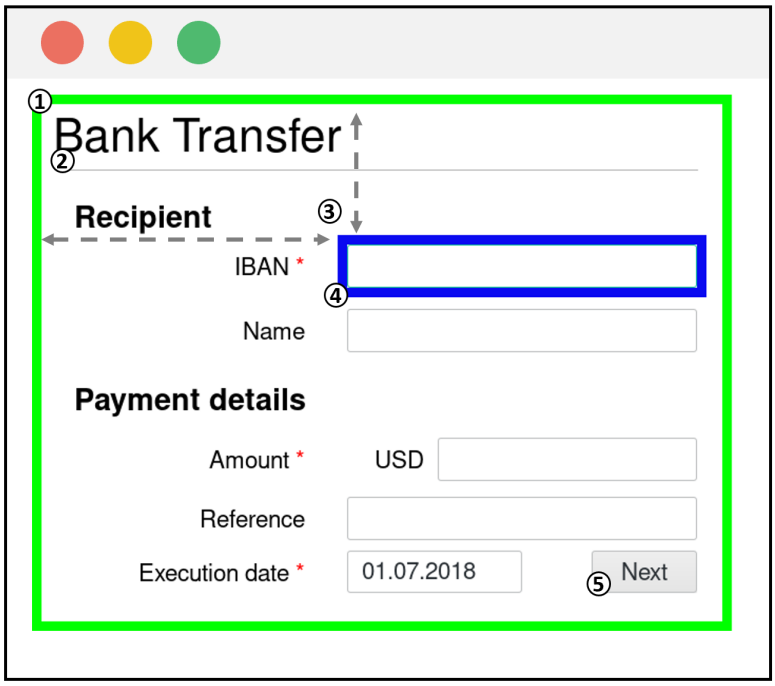


Figure 6.5: *IntegriScreen* hardening of web forms. The green rectangle ① supports the form localization and perspective realignment. The form is defined by its title ② and its precise specification of each UI element's relative position ③. The currently focused element is clearly emphasized by a blue rectangle ④, and the form is submitted upon a button press ⑤. The interface is based on the a real-world web form used for online banking by a major organization (shown in Figure 6.4).

② **Form Title.** *IntegriScreen* requires that each form has an unique title in order to support downloading of its corresponding specification file. While in this chapter we assume that the unique form identifier is displayed in the upper left corner, we emphasize its URL could also serve this purpose. As we discuss in Section 6.5, the adversary can only cause a denial-of-service by manipulating the form title.

③ **Form Specification.** To enable UI verification, *IntegriScreen* requires that the expected relative positions of all UI elements are known, and a specification of such positions is available to the mobile application after the client loads the web form. An example of such specification is given in Listing 6.1: the borders of each UI element are precisely defined relative to the frame border, as are their type (label or input element), and initial values. Such specification allows the application to ensure that none of the expected UI elements are missing,

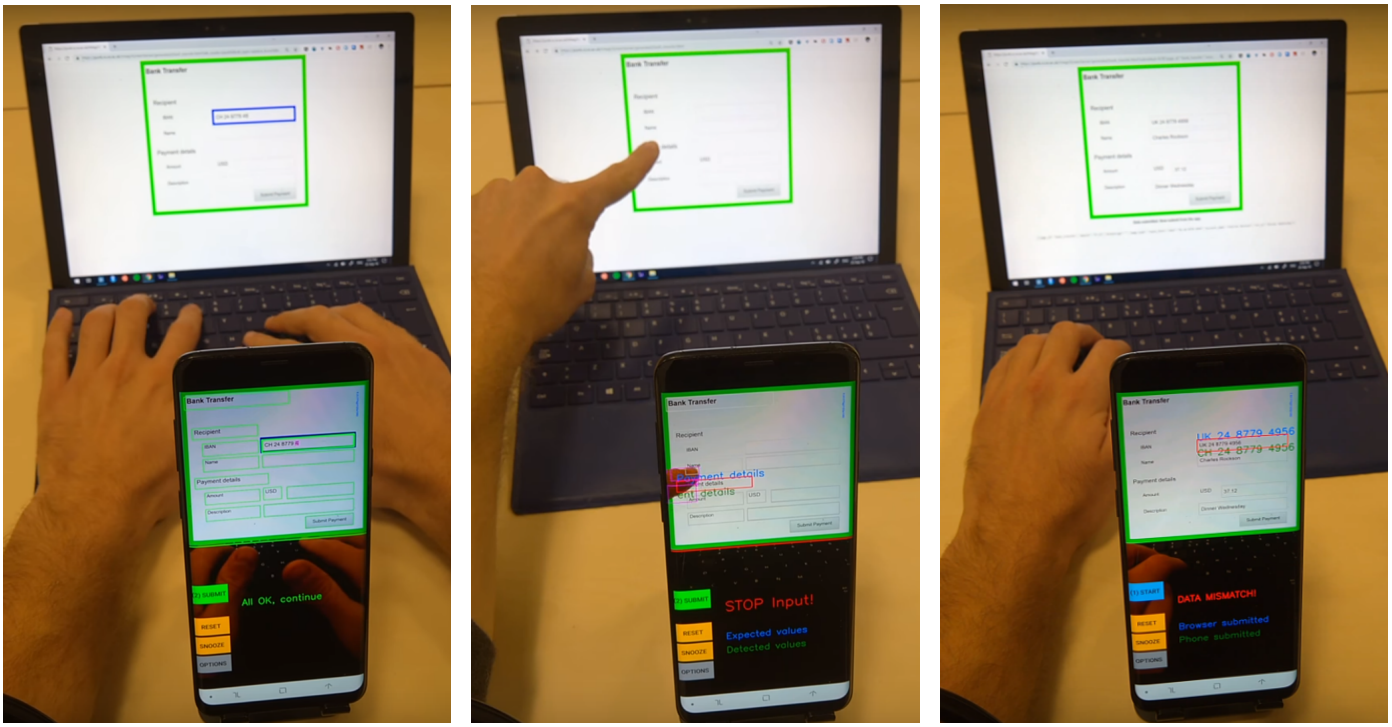
Listing 6.1: Form specification example.

```
{
  "ratio": "1280:960",
  "page_id": "Bank Transfer",
  "elements": [
  {
    "id": "IBAN_label",
    "type": "label",
    "initialvalue": "IBAN *",
    "x_position": "10"y_position": "25"width": "30"height": "8},
  {
    "id": "IBAN_value",
    "type": "input",
    "initialvalue": "",
    "x_position": "42"y_position": "25"width": "50"height": "8},
  (...)
  ]
}
```

modified, or added, and thus prevent UI-based manipulation.

④ **Focused Element Border.** The input element that is currently being edited (in focus), must be highlighted in order to indicate the screen area where data changes are allowed to happen. While such a visual guide is already implemented on most modern browsers (and we use a blue rectangle for simplicity), *IntegriScreen* allows for full customization of its design. In order to simplify detection of attacks, we mandate that while the focused element is changing, the remainder of the form (outside of the focus) must remain static. Finally, the UI is mandated to also set a restriction on how quickly can the focus move from an element whose value has changed. As we discuss in Section 6.4.3, this does not limit normal user behavior, while at the same time ensuring stable OCR performance and detection of on-screen modification attacks.

⑤ **Visible Request Data.** Visual supervision requires that the results of inputting the sensitive data that comprises the remote request are clearly shown on the client's screen. This, for example, allows mouse interaction to change the state of checkboxes or a calendar widget to choose a date, as long as the chosen value is visibly shown afterwards. We assume that the form includes a "Submit" button that generates the remote request towards the server.



(a) UI Verification. After loading the web form specification, the application verifies that all UI elements shown on the client's screen match their expectation and overlays the green element borders on realigned camera feed, indicating that users can start with input.

(b) Supervising user input. As the value of one of the UI elements changes unexpectedly (due to occlusion or attack), the difference in expected and detected values is highlighted. Any subsequent input is prevented until the suspicious modification is reverted.

(c) Results of the server comparison mismatch, shown on the mobile device. The client-submitted data is shown in blue, while the mobile device-submitted data is shown in green.

Figure 6.6: User experience of the smartphone application (a) and (b) show the mobile application during normal user input. The smartphone automatically aligns the camera feed to focus on the green rectangle and realign to a flat perspective. The expected locations and values of UI elements are precisely overlaid on the camera feed and shown on the smartphone screen. (c) is an application screenshot, captured during automated evaluation. The image shows a mismatch between the client-submitted and smartphone-submitted data.

The above requirements are highly extensible, as they do not enforce a specific layout and allow services that implement *IntegriScreen* to use their own branding and style. We further discuss relaxing or modifying some of the requirements in Section 6.9.

6.4.2 Server Component

Protecting an existing remote service with *IntegriScreen* requires that a simple *IntegriScreen* server component is added as a proxy between an existing service

and potentially compromised client. The server component ensures that only a request with a matching proof-of-intent is actually forwarded to the remote service, which can thus remain unmodified.

In summary, the server component provides the following functionalities:

1. **Serves** the web form specification upon a request from the *IntegriScreen* mobile application.

2. **Receives** requests from the client.

3. **Receives** proofs-of-intent from the mobile application.

4. **Matches** received requests with their corresponding POI-s. If there is no mismatch, the server component forwards the client's request to the remainder of the remote service, otherwise, the request is dropped.

5. **Notifies** both the client and the mobile device about the matching outcome.

Generating specifications for web forms. The mobile application requires that the layout of the web form is described in a specification file, which can either be provided by developers of the remote service, or automatically generated by the *IntegriScreen* server component. We note that the other direction is also possible: the server component generates the valid HTML web form based on its specification. We implement this approach for the experimental evaluation in Section 6.7, in which we evaluate the performance of the system on hundreds of tests on randomly generated web forms.

6.4.3 Smartphone Application

The primary goal of the mobile device application is to provide an independent, out-of-band confirmation of the user's intended input despite the client compromise. This is achieved by verifying that the user interface matches its specification through the user interaction, supervising against any on-screen modification attacks, and

capturing the data shown on the client's screen to generate and send a respective proof-of-intent.

Namely, after being loaded by the user, the *IntegriScreen* mobile application performs the following steps:

1. **Locates** the border of the web form shown on the client's display, realigns the captured video feed to a flat perspective (as shown in Figure 6.8), and extracts the form's unique title.

2. **Loads** the corresponding UI specification from the *IntegriScreen* server.

3. [**Continuously**] **verifies that the UI** of the web form presented on the client device matches its specification, i.e., that all UI elements are present and that none have been modified or added.

4. [**Continuously**] **supervises user input**, allowing only the element in focus to change, and only when the user is present and active.

5. **Submits** the generated proof-of-intent to the same server endpoint from which the web form specification was loaded.

6. **Notifies** the user about the result of server's data comparison: either success (if client and smartphone-submitted data match) or failure (in case of data mismatch). In the latter case, the user could be allowed to choose which of the two versions of submitted data he wants to use (or a combination thereof), thus effectively doing a 3-way merge of the data submitted through two channels.

Steps **(3)** and **(4)** form the core of the *IntegriScreen* mobile application, as they are continuously executed for each frame that the mobile device captures, thus protecting the user against UI manipulation and on-screen modification attacks. We now describe them in more detail.

(3) Continuous UI Verification. For each captured frame, the application verifies that all UI elements match their expectation by running text extraction on each of the positions specified in the form specification. Values of labels never allowed

to change, and their text values are compared to the received form specification. Since input elements can and do change, the application keeps track of their latest known value and verifies if it matches the expectation. Finally, to detect any added UI elements (which could misguide the user), the text detection module is running on the frame from which all positions of UI elements have been whitened out.

In the case of any missing, modified or added UI element, the application visually warns the user that input is not allowed and further notifies the user by ringing or vibrating if any input element still changes. The application clearly augments its preview of the screen to show which of the elements are problematic (showing them in red), and prevents any data input until the mismatch is ameliorated. An example of such warning is shown in Figure 6.6c.

(4) Continuous Input Supervision. Besides verifying that all UI elements have their expected values, the application also verifies that several other conditions hold before it updates the internal value model for any of the input elements:

1. **Only the focused element changes.** The only element that is allowed to change is the input element in focus. All other elements must remain unchanged.

2. **Activity detection.** If the value of the active element has changed since the last frame, the app must have also detected that users hands have been present and that they moved.

3. **Focus changes slowly.** If some active element's value changes, the focus should not change to another element in less than x ms after the last edit and in less than y ms since this element first came into focus. The server can optionally set the values of x and y in the specification file, otherwise default values are used (in our prototype we set $x = 300$ ms and $y = 2000$ ms).

Requirement **(1)** ensures that the user needs to only focus on the value of the currently active element (which they are editing), while all other elements are *protected*; **(2)** ensures that no element changes while the user is not present and editing the form; finally, **(3)** serves two goals: on the one hand, given the limitations

on the number of frames that the mobile devices can process each second, it ensures that any change in the value of the active element is correctly detected. On the other hand, it ensures that the adversary can not move the focus to another element and change its value without legitimate user noticing, as this would either last for a minimum of several seconds, or be detected as an attack attempt by the application. We experimentally evaluate these assumptions in the user study in Section 6.7 and show an example of input supervision detecting malicious modification in Figure 6.6c.

Furthermore, we note that, in order to prevent the adversary from prematurely submitting input data, the proof-of-intent is submitted to the server only as a result of the user explicitly pressing the *Submit* button on the mobile device.

Occlusion and multi-page forms. We finally note that the proposed system design natively supports user interactions in which the form is temporarily occluded, such as changing the browser tab or minimizing the browser window. In such cases, the mobile application will prevent any input due to UI verification failing, but as soon as the form is again displayed on the client's screen, it will repeat UI verification and, if the values of all elements remained unchanged, continue allowing user input.

Furthermore, such design allows for simple support of multi-page interfaces, as the application will simply store the values of all input elements on each page as the user edits them in arbitrary order. The only required modification would be to implement the mobile application logic to generate a proof-of-intent that includes values from the whole multi-page form.

6.5 Security Analysis

We now informally analyze how *IntegriScreen* provides the required security guarantees necessary to authenticate the user input under the specified strong adversary model. The analysis starts with general reasoning, and then continues describing the protection offered by *IntegriScreen* against more specific attacks.

Generating arbitrary client requests. For most authenticated remote services, as soon as the legitimate user authenticates, the compromised client necessarily gains

access to their credentials and can subsequently generate (authenticated!) requests to the remote server without any user interaction. However, this is not the case for *IntegriScreen*, since the remote server requires an authenticated proof-of-intent from the smartphone application in order to act upon any request coming from the client.

Covertly modifying the client request. *IntegriScreen* protects against the adversary that modifies the data exchanged between the client and the server without any visible changes on the client's screen since this would cause the received proof-of-intent from the smartphone application to the matching by the server.

As a result, the adversary is forced to achieve that the mobile device also submits the proof-of-intent that exactly matches his malicious data. Such data manipulation can happen either by tricking the user into submitting data that matches adversary's intentions (e.g. by manipulating UI instructions), or by carefully changing the data on the screen so that the modifications are detected by the mobile application, but not by the user.

6.5.1 UI manipulation attacks

If the client is compromised, the adversary fully controls the web form shown to the user on the screen and could change the context to manipulate the user to actually input the malicious data himself.

IntegriScreen prevents such attacks by ensuring that the web form shown to the user directly corresponds to the specification: all labels must show the correct text, all default values of input elements must be present (as their modification could also misguide the user), and no additional text is allowed to be present on the presented web form.

If any of those requirements are not met, the smartphone application clearly shows the offending UI element to the user and does not accept any new input. We experimentally measure the performance of this UI verification in Section 6.7.1 and discuss the potential extension to non-textual elements in Section 6.9.

Modifying the form header. Since the smartphone application relies on optical recognition of the form title to detect which form specification to load, the adversary can easily cause the smartphone application to load a form specification that he fully controls by changing the title name. We, however, emphasize that this only results in a DoS attack since the Application uses the same endpoint to load the specification and to submit the proof-of-intent at the end of user input. As a result, the original server endpoint never receives a matching proof-of-intent and the attack is not successful.

6.5.2 On-screen data modification

Besides attempting to manipulate the user into inputting malicious data, the adversary can attempt to directly add, modify, or delete the data shown on the screen. This can happen either before, during, or after the user's input. The adversary succeeds if such changes are not detected by the user, but the mobile application registers them as legitimate. *IntegriScreen* includes several mechanisms to prevent such attacks.

Not conforming to UI behavior specifications. We start by noting that the application enforces that the form must follow the design guidelines specified in Section 6.4. For example, in order to change any element, the adversary must clearly indicate the location of attempted change to the mobile app by showing a blue rectangle around it. Similarly, if the web form does not show the blue rectangle to indicate the focused element, no legitimate user input is allowed, resulting only in a DoS. If, for instance, multiple rectangles are shown at the same time, or the user is allowed to change focus faster than the form specification allows, the application warns the user and prevents any changes in the values of its proof-of-intent. The adversary is, thus, not allowed to change the form behavior.

Modification during user absence. One potential attack would be if the adversary waited for the user to load the form, start the application, and leaves their desk while letting the application run. During this time, the adversary

has an opportunity to change any values on the screen, and thus trick the user into submitting modified data. *IntegriScreen*, however, prevents such attacks by mandating for user's presence and hand activity during any detected screen changes and raises an alarm if they are absent. The adversary must thus attempt an attack concurrently with user's interaction with the client. We discuss other ways to implement this step in Section 6.9.

Concurrent Data Modification. Given that the application raises an alarm and prevents any changes of its data model if two elements are shown to be in focus or if an unfocused element ever changes, such an attack would be detected. Therefore, if the user is present and is inputting data into some element X, the adversary is unable to concurrently change any other element Y due to the fact that Y would have to also be in focus. We experimentally evaluate the performance against such attacks in Section 6.7.2 and in the user study (Section 6.8).

In case of concurrently modifying the active input element by both the adversary and the user, we assume that the user detects such changes (which are similar to autocorrect not behaving according to user's expectation) and will not move the focus to the next input element until they are satisfied with its content. We evaluate this assumption in Section 6.8.1 and discuss further measures to reduce this assumption about user behavior in Section 6.9.

Rapid change of focus. Finally, a potential attack is to change the focused element from X to Y and back to X so that the mobile application detects the change in focus and in value of Y, but the user remains unaware (or considers the change to simply be a glitch). However, if such a change happened too fast, it would be raise an alarm due to the limits imposed in the form specification and enforced by the application. Namely, if the value of a focused element Y is changed, then the form should delay changing the focus to the next element (X) for at least 300 ms after Y is changed, and also ensure that the total time that element Y was focused is at least 2 seconds.

The users are thus likely to detect such sudden changes in focus during data input (for at least 2 seconds), and we evaluate this assumption as part of our user study in Section 6.8.

6.6 Prototype Implementation

In order to evaluate the real-world feasibility of visual supervision of user input to a compromised client, we implemented a prototype of the proposed *IntegriScreen* system using an Android smartphone, Apache server and multiple web forms that have been adapted according to current *IntegriScreen* specifications. Overall, the system prototype and evaluation framework were implemented using a total of 5165 lines of Java, C++, Python and JavaScript code.

6.6.1 Mobile application

We implement the prototype of the smartphone application on a Samsung Galaxy S9+ mobile device. We also evaluate the performance of the developed prototype on two other smartphones: Samsung Galaxy S6 and Google Pixel 2XL. The developed Android application consists of 3340 lines of Java and native C++ code. We use Google's Android Text Recognition API [225] for optical character detection and recognition. The rest of core image processing functionality is implemented both in Java and C++, using OpenCV [226].

User experience. As shown in Figure 6.6, the user interface of the mobile application is simple to use. The typical usage consists of simply pressing a single button to start the supervision, and proceeding with normal input on the client device as soon as all the camera feed of all UI elements is augmented by green rectangles, which indicates that UI verification succeeded (Figure 6.6a). Once the user finishes with input and submits the data from the client, they are only required to press the button "Submit" to submit the generated proof-of-intent to the server. In case of any mismatch, the application clearly shows the offending element in red, together with the two different values that did not match: either the *expected*

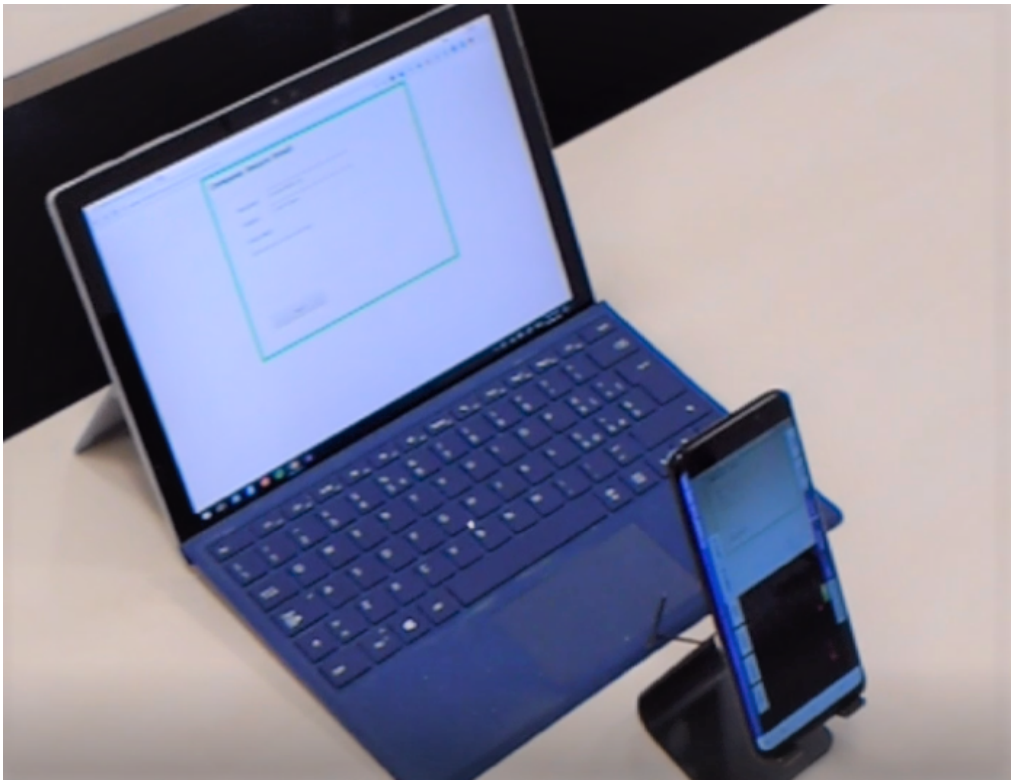


Figure 6.7: Experimental setup. The mobile device is positioned on a simple stand between the user and the client's screen.

value and the text *detected* on the screen in case of UI verification mismatch, or the values submitted by the *browser* and the *phone* (Figure 6.6b).

Perspective realignment. Considering the wide range of possible angles at which the smartphone can be positioned relative to the client's screen and different sizes and ratios of forms that the application needs to support, the crucial first step in the image processing pipeline is to detect the corners of the green rectangle (currently using their hue values) and then use linear perspective realignment [227] to crop and reorient the input frame as if it was captured with no angle.

OCR matching. Given the inherent limitations of visually detecting whitespace and imperfections in the used OCR libraries, in this prototype we consider two strings equal if they only differ in whitespace and character capitalization. We note that the proposed system can easily support various levels of matching strictness that would be provided in the form specification.

Table 6.1: Success rates of UI Verification on a 100 randomly generated forms. The second value shows the overall percentage of correctly detected UI elements across all forms. In the first three measurements, devices were positioned in front of the screen (Figure 6.7), with 5 seconds per form load.

Mobile Device	Forms	UI Elements
Samsung Galaxy S9+	98%	99.75%
Google Pixel 2XL	93%	97.86%
Samsung Galaxy S6	82%	95.15%
Samsung G. S9+ [Fig 6.8]	93%	99.12%
Samsung G. S9+ [3 seconds]	93%	98.97%

User activity detection. We use the lower part of the camera feed to detect hand activity by filtering out the background of the keyboard, and detecting significant changes between two consecutive frames.

6.6.2 Client and Server

We simulate the compromised client on two laptops: Microsoft Surface Pro 4 and Apple MacBook Pro. As shown in Figure 6.7, the smartphone is placed on a phone stand in front of the client's keyboard.

In order to simplify the process of generating a large number of valid HTML web forms and their corresponding specifications for automated system evaluation, we developed a tool that automatically converts a given specification file to a valid web form.

The prototype server is implemented using the Apache Tomcat 9.0 framework [228] with 870 lines of code. It serves the web forms requested by the client, accepts the requests and proof-of-intent from the client and the mobile application and notifies them of the comparison results.

6.7 Experimental Evaluation

We now evaluate the guarantees provided by the *IntegriScreen* system, by running a series of experimental tests against attacks that an adversary might attempt.

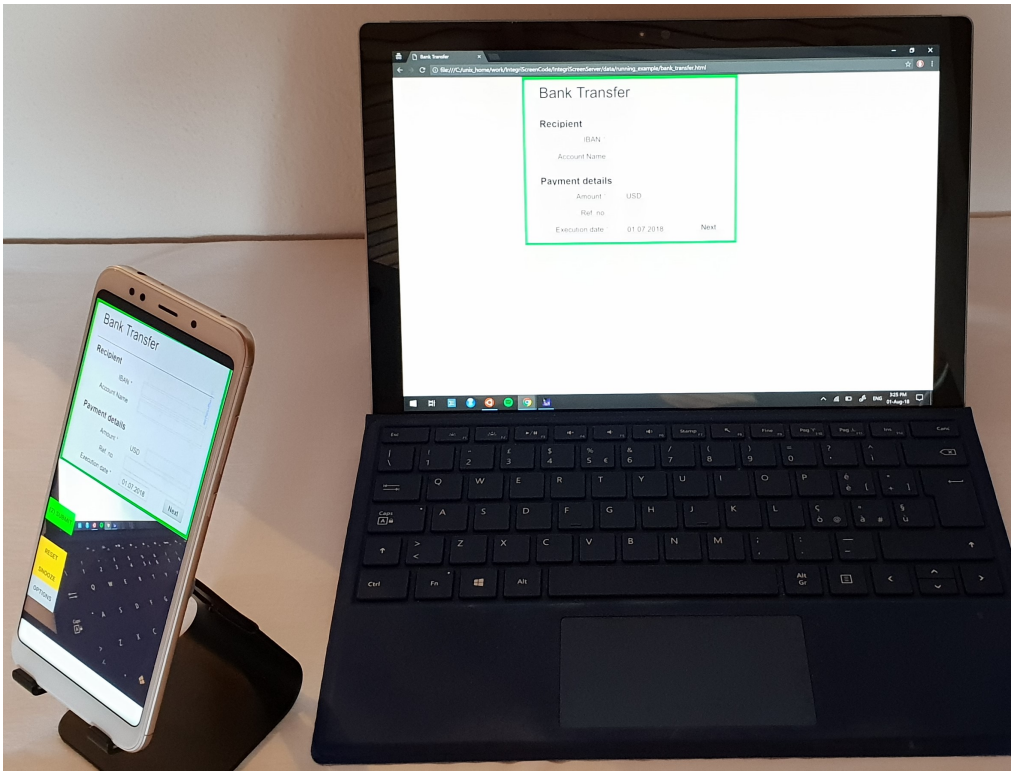


Figure 6.8: Alternative experimental setup, in which the angle towards the client's screen is significant. The application corrects the angled camera feed to a flat perspective, which allows successful text detection and precise alignment of form elements.

6.7.1 Preventing UI Manipulation

We start by evaluating the performance of UI verification for a randomly generated set of web forms of varying complexity in the absence of attacks. This provides a measure of application's success in correctly positioning and detecting the visual elements on the screen. Given the differing angles and positions between the screen and mobile device, as well as the potential OCR mismatches, this is a challenging task. The performance is computed for several configurations, differing in the used mobile device, relative positioning between the two devices and the total time allowed for the form to be loaded and verified.

We finally analyze in more detail the verification performance of the configuration that we use in the remainder of experimental evaluation.

Setup. We use a set of 100 randomly generated web forms (such as the one in Figure 6.9), which have a varying complexity of visual elements (between 4 and

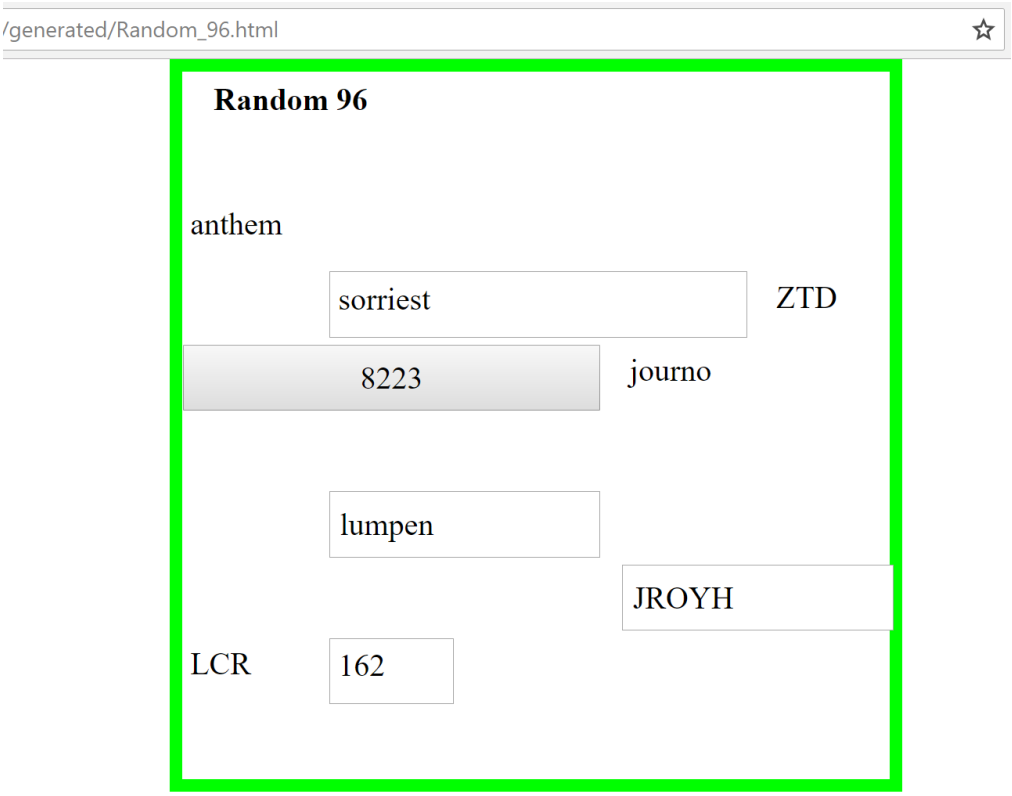


Figure 6.9: Example of a randomly generated form used to test the UI verification performance. Each element consists of an English word, number, or a random alphabetical string.

9) and types of data as labels and default input values (English words, numerical, and random alphabetical strings).

The testing script on the client consecutively loads all the forms in the dataset, showing each for 5 seconds before continuing to the next one. During each of these periods, the smartphone application automatically detects the form based on its title, loads its specification from the server, and performs UI verification. If the verification is successful, the form is marked as perfectly verified; otherwise, the offending mismatches are stored for later analysis.

We evaluate the UI verification performance using three Android devices: Samsung Galaxy S9+, Google Pixel 2XL, and Samsung Galaxy S6. Additionally, we evaluate the difference in performance when the device is positioned directly in front of the screen and to the right of the keyboard, observing the client's screen from an angle (Figure 6.8).

Results. Table 6.1 shows the results of verifying the UI for all randomly generated forms, which are highly encouraging. The highest recognition rate was achieved when the application is deployed on Samsung Galaxy S9+, which was successful in detecting, loading the specification, and **verifying the text values of 98%** of the forms in less than 5 seconds. Consequently, this translates to 99.75% of the UI elements being correctly detected and recognized on the screen. The other two tested devices also achieved high detection performance: Samsung Galaxy S6 correctly verified 82% of forms and more than 95% of UI elements, while these percentages are 93% and 97.86% for Google's Pixel 2XL, showing the feasibility of the proposed approach across a range of current mobile devices.

All three devices achieved stable performance, with average processing rates of 2.6 (Samsung S6), 3.3 (Google 2XL) and 4.7 (Samsung S9+) frames per second.

Positioning and verification time. Table 6.1 also shows the performance of the UI verification procedure for a spatial configuration in which the mobile device is positioned on the right side of the client's keyboard, resulting in a significant angle towards its screen (Figure 6.8). Despite the challenges of precisely realigning and detecting UI elements at a large angle, the evaluation shows that the prototype is successful at **correctly verifying 93%** of the forms from the dataset, resulting in an overall per-element detection rate of 99.12%.

Finally, we measure the verification performance when the total time allowed for the application to verify a single form is reduced to 3 seconds, thus increasing the required load time before users can start input. The results show that the application maintains a high detection rate even for such short intervals. The form verification rate for this short duration remains at a high 93%, with a per-element detection rate of 98.97%.

Text recognition mismatches. It is important to note that the verification rates depend both on real-time detection and realignment of the captured video, and on the general performance of the used OCR engine. We provide a sampling of OCR mismatches in Table 6.2. Despite some of the randomly generated forms

Table 6.2: Text recognition mismatches.

Actual Text	Detected Text
FBN	FB
mummify	munmmi
RXSTFHJJ	RXSTFHIJ
nebulae	bulae
4714004	471400
280	[not detected]
064	054
bairn	baim

being mismatched, the analysis shows that the majority of failures are a result of a consistent mismatch of a single UI element, which was often misdetected by a single character.

6.7.2 Preventing On-Screen Data Modification

The period of user input is continuously supervised by the smartphone to prevent the adversary from maliciously modifying the data shown on the screen without the user noticing. We now evaluate the probability that *IntegriScreen* application detects modification of an element that is not a result of user activity, i.e., one that happens either during page load, or at the time of user input, outside of the focused element.

Setup. To achieve testing consistency and allow running a large number of controlled experiments, we simulate user input with Selenium WebDriver [229], a commonly used UI testing framework. We evaluate potential UI manipulation and on-screen data modification attacks by consecutively loading the randomly generated forms, simulating user input, and running a script that randomly replaces three subsequent characters either before or during simulated user interaction (Figure 6.6c). In particular, the simulated user inputs random strings with realistic speed of an average *touch* typist (120-200 ms per character) [230] in each of the input elements, and finally submits the filled form to the server.

This adversarial change can be one of two different types of attacks that *IntegriScreen* application must automatically detect and prevent:

B_1 Concurrent modification. Changing an input element that is not in focus concurrently with the simulated legitimate user's input.

B_2 Modification before input. Changing the value of an element while the form is being loaded.

We consider the attacks prevented if the application stops user input before the form is submitted.

A short screen recording of simulating $\mathbf{B_1}$ attacks can be seen at the following link:

`https://tinyurl.com/integriscreen-video`

Results. The results of simulating both attacks are highly promising. The application **successfully detected 98.9% (91/92) of the simulated attacks of type B_1**, preventing the user from submitting data that was maliciously modified during user input.

When simulating attacks of type $\mathbf{B_2}$ and modifying a single element during load time on each of 100 randomly generated forms, the *IntegriScreen* application was **successful in detecting 100% of the simulated attacks**. More precisely, none of the forms were marked as successfully verified, thus preventing the user from becoming victim of an UI manipulation attack and potentially submitting data under different intent. Overall, out of 800 UI elements that were loaded (out of which 100 have been modified), the application detected 690 of them to be according to the specification.

In conclusion, our automated evaluation shows that the developed prototype is capable of detecting almost all simulated attacks that are detectable from visual supervision. In the next section, we evaluate the probability that users detect potential attacks which are hard to distinguish by simply observing the client's screen, and measure if users would successfully understand the potential attacks detected by the implemented system prototype.

6.8 Prototype User Study

Finally, in order to evaluate the usability of the developed prototype and the security guarantees provided by *IntegriScreen*, we invited participants for a preliminary user study.

Demographics and instructions. We recruited a total of 15 participants aged between 21 and 31. The only requirement was a minimum age of 18 years; the test population consists of 10 males and 5 females.

The participants were first given a short, written description about the experiment and the *IntegriScreen* system (included in the Appendix A.1) and instructed that they can chose to withdraw their participation in the experiment at any time.

Participants signed a consent form before starting the experiment and were debriefed after the experiment about the simulated attacks. Given that no personally identifiable information was captured in the experiment and that all data was stored anonymously, the institution that hosted the experiments did not require an institutional review board approval for these kinds of studies.

They were then instructed to fill out 10 forms that represent different online banking transactions by inputting the provided test data. The data for input was given either printed out on paper, or on another screen, as per the participant's choice. As shown in Figure 6.5, each transaction required inputting the recipient's IBAN (16-24 alphanumeric bank account code), first name and last name, as well as the transaction amount and reference description.

6.8.1 Experimental Attack Evaluation

While participants were filling out a randomly chosen subset of forms, the client device simulated execution of a total of four different types of attacks on the *IntegriScreen* system. The goal of these simulations was to observe user behavior: the likelihood of detecting attacks themselves or correctly understanding the attack detections made by the *IntegriScreen* prototype.

Setup. The experimental setup consisted of a smartphone Samsung S9+ running the developer *IntegriScreen* application, a Surface Pro 4 laptop that represented the local client used to input data, and another desktop computer that was used to setup the experiment and to display the test data that participants were inputting. In order to observe natural responses and not prime participants to expect potential attacks or specific attack type, they were given no instruction that potential attacks might be simulated during the experiment.

The attack types simulated in the user study are:

U_1 Modify the value of the currently focused element during user input.

U_2 Move the focus to another element, modify its value, and return the focus to the original element. Change the original element in accordance with user's keypresses that were delayed while focus was shifted.

U_3 Without moving the focus, modify the value of one of the inactive UI elements while the user is changing another UI element.

U_4 Modify the value of some element after 3 seconds of user inactivity without moving the focus.

U_1 and **U_2** were chosen to measure the assumptions about participant's detection of potential attacks which can not be prevented solely by smartphone's visual supervision as the adversary's behavior is not distinguishable from legitimate user's. The other two attacks (**U_3** and **U_4**) measured the success rates of the mobile application detecting and the participants successfully correcting the potential attacks following the application's warnings.

The order of attacks was randomized. Each attack modified 3 consecutive characters of the same type as the data already on the screen, e.g. digits were replaced with digits, and so on). Attacks were triggered after users made 6 keypresses on a randomly chosen input element (trigger element) for **U_1**, **U_2**, and **U_3**; in case of **U_4**, the attack was initiated as soon as trigger element modification was followed by 3 seconds of no keypresses. The trigger element and the element that was modified

(target element) were randomly chosen. The only exception was $\mathbf{U_1}$, where both the trigger and target element were always the IBAN field in order to simulate the worst case scenario, which is both the hardest to detect for participants (due to no obvious syntax in IBANs) and potentially most damaging if successful.

Results. During the experimental user study, a total of 56 attacks attempts were simulated by the client system (since $\mathbf{U_4}$ was not always activated). We consider an attack successful if the participant did not correct the modified data before submitting it to the remote server. The detection rate of attacks in our study is highly encouraging: users were successful in detecting 91.1% (51/56) of simulated attempts, even without receiving any indication or training about potential attack vectors that they should guard against.

The only type of attack that was successful against more than one participant was $\mathbf{U_1}$, in which the adversary concurrently modifies the value of three characters of the IBAN at the same time as participants are copying it from the data sheet. As a result, their focus inherently switches between the source data and the UI element that they are modifying. The participants, however, still detected and corrected 80% (12/15) of such attempts. These results are closely in line with previous research, which concluded that participants spent between 70 and 80 percent of time looking at the screen during textual input [231]. Outside of a deception-based user study, however, the application would instruct the users about the need to verify the focused element before proceeding to the next one, thus likely further decreasing the chances of a successful attack.

The remaining attack types were all detected in more than 90% of attempts: Attack $\mathbf{U_2}$, in which users are required to detect that the focus changed to another element without their interaction was detected in 93.3% of attempts. All users were able to successfully correct $\mathbf{U_3}$, given that the application clearly indicated the element that was modified while not being in focus. Similarly, only a single user did not correct an instance of attack $\mathbf{U_4}$, since the value on the screen changed a brief moment before submitting data from the browser and then from the phone.

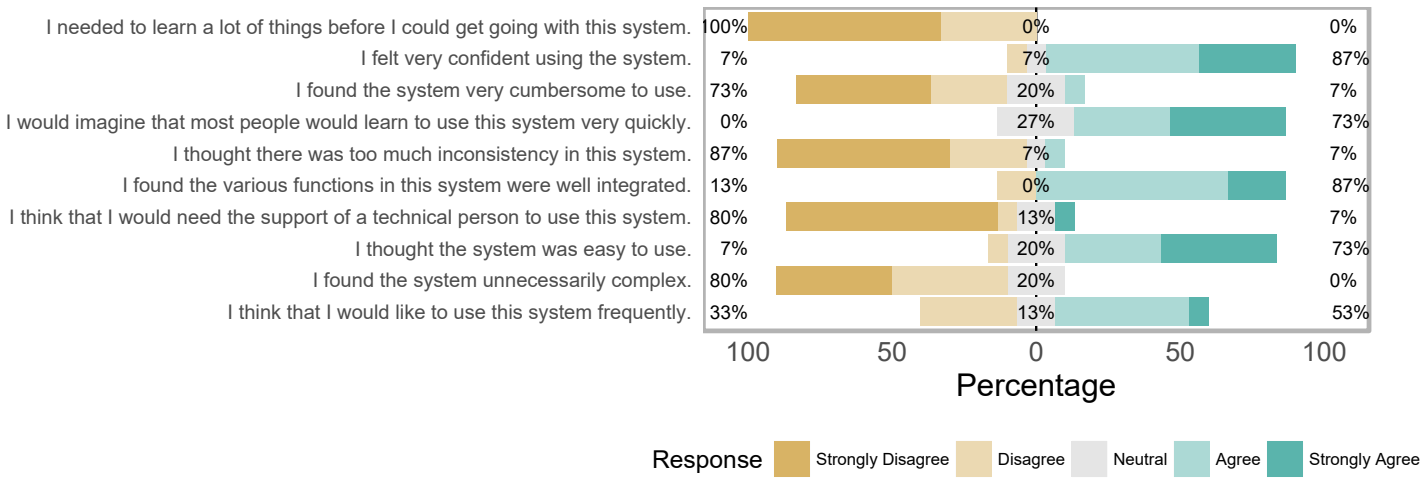


Figure 6.10: Participants' responses to the SUS questionnaire show a high average SUS score of 78.3. While some participants felt they would need a help of a technical person (Q4, 7%), most generally found the system easy to use (Q3, 73%), felt confident using it (Q9, 87%), and believed others would learn to use the system very quickly (Q7 73%).

In cases where no attack was simulated, 3 participants had to reload one of the web forms due to a misrecognition of the OCR engine; no false mismatch was ever reported by the server. Overall, the participants successfully submitted data for 97.6% of transactions that they were instructed to perform.

6.8.2 Usability Questionnaire

At the end of the experiment, the participants were asked to fill out a System Usability Scale (SUS) questionnaire [151], a general and broadly used tool for evaluating the usability of systems and products [217].

The questionnaire consists of 10 Likert scale statements (1 - Strongly disagree to 5 - Strongly agree), and the overall usability score is computed by adding the scores on odd-numbered questions (e.g. *"Q9: I felt confident using the system"*), and subtracting the scores on even-numbered questions (e.g. *"Q6: I thought there was too much inconsistency in this system"*), before normalizing the value to lie between 0 and 100.

Given the lack of similarly deployed systems that aim to protect the integrity of each user's request to the remote server, we chose SUS as a general tool that

could be easily compared with other approaches that aim at solving the same challenge in the future.

Results. The scores of user's perception of the prototype system are shown in Figure 6.10. None of the participants thought that they had to learn a lot before they could get going with the system (Q10) or agreed with the statement that the system was unnecessarily complex (Q2). While some felt they would need the help of a technical person to start using the system (Q4, 7%), most generally found the system easy to use (Q3 73%), felt confident using it (Q9 87%), and believed others would learn to use the system very quickly (Q7 73%).

We note that participants in our study evaluated the usability of *IntegriScreen* only after being required to consecutively input 10 transactions, along with unexpected behavior as a result of attack simulations that some participants perceived as technical glitches of the system rather than malicious behavior.

However, the overall SUS score given by the participants was a high **78.3**. Previous research on interpreting individual SUS scores with a single adjective would thus subjectively describe the usability of the developed prototype as "Good" (mean SUS of 71.4) [217]. The high rates of attack detection and the achieved usability scores of the developed prototype thus allow us to conclude that the experimental evaluation confirms the potential of deploying visual supervision of user input in the future.

6.9 Discussion

We now discuss the open questions, limitations, and possible extensions of the proposed system.

Detecting user attention and non-repudiation. In this research, the system uses hand movement to detect user's presence and activity. However, when the mobile device is placed between client and the user, its front-facing camera is well positioned to capture the user's face. Given the face tracking capabilities that are available in recent iOS and Android mobile phones, as well as recent advances in

mobile camera-based eye tracking [232], the system could be extended to precisely track user's attention on the screen and require that gaze is present for certain data modification. Furthermore, if the mobile device used face recognition to continuously authenticate the user, the system could also provide non-repudiation guarantees.

Non-textual UI elements. As the first step towards implementing visual supervision to authenticate user's interactions with remote services, in this work, we focused on textual input. However, the presented approach can be extended to support an array of non-textual UI elements – as long as their final state is visually shown on the screen – such as checkboxes, sliders, or calendar widgets. Furthermore, while UI verification currently focuses on text extraction, we note that this step could also be implemented by a more literal comparison of the client's screen with an screenshot of the web form, as rendered by the server, given that its ratios and relative positions of all elements are clearly specified.

Multiple device compromise. In this work, we assume that only the client device (laptop) is compromised; the adversary has no control over the smartphone, which is running the *IntegriScreen* mobile application, and which can only be unlocked by the legitimate user. We note that it should not be automatically assumed that these two devices are completely independent, since many users often connect their smartphones to their laptop devices for synchronization, which could allow the adversary to also compromise the smartphone. While such behavior indeed increases the likelihood of compromise of both devices, we note that compromising both devices still requires additional effort for the adversary. Finally, we note that in an opposite scenario to the one considered in this chapter (when only the smartphone is compromised), the adversary is still prevented from modifying or generating any request since the data sent from the compromised smartphone would not match the data sent from the client that is not under his control.

Privacy and security of visual supervision. While continuously recording one's interaction with another electronic device seems intrusive, we note that all processing in *IntegriScreen* happens on the mobile device. Therefore, the server

only receives a duplicate of the data from the client. If sending duplicate of data is not suitable for any reason, it is straightforward to modify the system to compute and only send a digest similar to TAN to the server.

However, a visual channel gives users clear control over which data the mobile device observes, i.e., only what is shown on the screen at a given moment. This is in contrast to most OS-level applications, which typically get unrestricted access to the whole filesystem and can perform significant privacy violations in the background while keeping the user oblivious.

Finally, using only a visual channel between the client and the mobile device has the benefit of significantly reducing the likelihood of a smartphone compromise since the smartphone does not need to have a bidirectional communication with the compromised client.

6.10 Related Work

Depending on the assumed system and adversary models, previous work either relies on a trusted hypervisor that supervises a compromised virtual machine or relies on the use of another trusted device that serves as a second factor.

Trusted hypervisors. Our general approach is most similar to Gyrus [220], a system that enforces integrity of user-generated network traffic of protected applications by comparing it with the text values displayed on the screen by the untrusted VM. However, while we use a similar approach, Gyrus requires application-specific logic and does not prevent potential UI manipulation attacks.

Not-A-Bot (NAB) [222] aims to ensure that the data received from the client has indeed been generated by the user and rather than malware. This is achieved by having the server require a proof of user's keyboard or mouse activity shortly before each request, which is generated by a trusted *attester* application. Similarly, BINDER [221] focuses on detecting malware break-ins and preventing data exfiltration by implementing a set of rules that correlate user input with outbound connections.

While these approaches are similar to *IntegriScreen* in ensuring that outgoing requests match user's activity on the client device, our solution differs in that it allows for a fully compromised client.

Trusted devices. Assuming a fully compromised client mandates that an additional trusted device is used to secure the interaction with the remote server.

Weigold et al. discuss several approaches of using a dedicated hardware device to confirm sensitive transaction data and propose ZTIC [233]. Their system relies on a device with simple user input and display capabilities, on which users confirm summary details for a banking transaction. Another approach is taken by Kiljan et al. [234], who propose a simple *Trusted Entry Pad (TEP)*. The TEP computes signatures of user-input sensitive values and sends them independently to server for verification. However, such approaches either require users to input data, which breaks the normal workflow and duplicates efforts, or require them to confirm transaction details, which leads to habituation and decreased attentiveness. The approach of continuous visual supervision improves on previous work by neither requiring additional input, nor relying on user attentiveness during transaction confirmation.

6.11 Summary

This work is based on the realization that video capture and processing capabilities of mobile devices are becoming sufficient for continuous visual analysis of other electronic devices. Motivated by the idea of having an MR headset serve as a security advisor during sensitive interactions with other devices, we propose *IntegriScreen*. *IntegriScreen* is a system that forces a compromised client to behave honestly by analyzing its screen during user input and sending the extracted data to the remote server.

We show the feasibility of this approach by developing a fully functional prototype, evaluating it with a series of experimental tests, and running a user study in which we measure participants' responses to simulated attacks. The results are

highly promising. The system prototype automatically prevents simulated attacks in more than 98% of attempts, while participants in the study detect the majority of the remaining attacks. The SUS score of 78.3 provides an adjective evaluation of the usability of *IntegriScreen* of *"Good"*, which further validates the proposed system and the developed prototype.

Considering the rapid increase in processing power and camera quality of mobile devices, as well as the announcements of multiple smart headsets with front-facing cameras by some of the largest technology companies, we believe this research to be an important first step towards using visual supervision to secure users' interactions with untrusted devices.

The scientific man does not aim at an immediate result. He does not expect that his advanced ideas will be readily taken up. His work is like that of the planter - for the future. His duty is to lay the foundation for those who are to come, and point the way.

— Nikola Tesla

7 Conclusion

In this final chapter, we summarize the main contributions and discuss the potential avenues for future work. We conclude by providing closing remarks and lessons learned during the doctoral work presented in this thesis.

Motivated by the novel computer-human interfaces of the first commercially available mixed reality headsets, in this thesis we aimed to take a systematic approach to improving the security of mixed reality systems. We thus focused on challenges of confirming user's identity by analyzing their eye movements, establishing a secure channel between two mixed reality headsets, and using such devices to ensure the integrity and authenticity of the data that a remote server receives from a local client. Toward these goals, we have surveyed previous work and proposed, built, and evaluated three cyber-physical systems that fill the identified research gaps, while all relying on similar concepts.

When addressing the topic of usable and secure user authentication for mixed reality systems, we emphasize the challenge of visual observation and of replay attacks that stem from adversary's ability to reuse a previously captured biometric measurements. By building upon existing neurophysiological research, we therefore propose combining the biometric analysis of one's eye movements with an interactive visual stimulus which can always be made unique. The novel contribution of this research is in designing a stimulus that triggers predictable and reflexive responses,

proving the system with rapidly collectible and stable biometric measurements. Furthermore, by incorporating a challenge-response protocol with biometric authentication, our system is able to verify the freshness of the biometric measurements. Given that most MR headsets include or will include eye tracking capabilities, we believe this to be an important step towards incorporating fast replay-resilient user authentication in future mixed reality systems.

Inspired by the visual stimulus that consists of randomly chosen positions, in this thesis we next focused on securely establishing the shared mixed reality experiences between two MR headsets. Despite numerous earlier proposals for general device pairing, we note the lack of practical proposals for direct pairing of MR headsets, given the specific challenges that these systems pose: the devices can not output to anyone besides their user, who should not be required to take their device off their head. Furthermore, it is necessary to assume that malicious adversaries have the capability to observe all interactions between the legitimate participants. We thus propose *HoloPair*, a system that relies on the ability of MR headsets to precisely display the same holograms at independently computed positions in space to both users who wish to establish a secure channel. This visual guidance then allows them to use hand gestures in order to confirm the authenticity of the exchanged keys and thus establish a secure channel without relying on any PKI infrastructure or other trusted third parties. Due to the recency of these technologies, our work is the first security research that actually built and evaluated a system prototype based on a Microsoft's HoloLens mixed reality headset. Additionally, we contribute to the wider developer community by making the source code of the prototype system publicly available. As mixed reality headsets become more common, we believe this is an important contribution to making sure that shared MR experiences between previously unpaired devices remain secure.

The device pairing system that we propose in this thesis assumes exactly two users of reality headsets that wish to establish a secure connection. A natural extension of this research would be to investigate how can a group of users establish

a secure connection with the least amount of effort. One obvious approach would be to dedicate a single participant as the group leader who establishes individual connections with the remaining $N-1$ participants and then generates and distributes a shared group key. However, it is an interesting research challenge to propose group device pairing protocols for mixed reality systems that would require the shortest total execution time.

Recent research has shown that biometric systems that use eye tracking glasses achieve error rates that are comparable to the rates that we measured in our work. However, both research projects used a stimulus shown on a static computer screen. Given that the first MR headsets that support eye tracking capabilities have become available in the late 2018, and that they support showing visual stimuli as a physical object in user's surrounding, which would be very similar to the holographic objects used to guide gestures in Chapter 5, it is an interesting research challenge to implement the proposed eye-tracking biometric solution on those MR headsets and evaluate the achieved authentication performance, as well as the user's perception of the system's usability.

Given the similarities between the visual stimulus used in Chapter 4 and the gestures used in Chapter 5, another natural avenue for future work would be to automate the verification of the gesture by only requiring one of the users to follow the other user's finger with their eyes, without actually seeing the expected gesture. Since the latest MR headsets can track one's eye movements, the device would be the one making the final decision and thus preventing users from skipping the gesture verification step by always claiming a match.

Finally, in this thesis we show that mixed reality technologies do not only pose new research challenges, but that they can also be used to secure existing systems. We thus propose the concept of continuous visual supervision of legacy client devices (such as laptops) by the user's trusted device that serves as a second factor for input data authentication. This ensures that, even in the case of client compromise, the adversary is not able to maliciously craft or modify user's communication

with a remote server. We show the feasibility of this approach by building a prototype and evaluating it in a user study and a series of experimental tests. We believe this to be an important research direction towards future application of mixed reality technologies that help non-expert users in their security-sensitive interactions with other computer systems.

We implemented the aforementioned system on a smartphone due to their higher-resolution cameras and processing performance. However, as the capabilities of mixed reality headsets further improve in the near future, a natural extension of the work from Chapter 6 would be to implement the proposed system on a mixed reality headset, and thus fully unlock the potential of visual supervision of one's interaction with legacy systems.

Furthermore, *IntegriScreen* currently prevents malicious modification of data shown on the screen by limiting the speed at which element values can change and ensuring that changes happen only while a user is present and actively typing. Considering that smartphones and mixed reality headsets are starting to include gaze tracking capabilities, another direction for future research that connects Chapters 4 and 6 is to extend the concept of visual supervision by integrating the exact information about where the user is looking. This would further increase the security of the proposed system by making it possible to require that the changes on the screen can happen either only where the user is currently looking, or have to be gazed at in order to be considered *validated* by the user.

When a new and overarching technology becomes widely available, it often provides the adversaries with novel attack capabilities, requiring that existing threat modeling assumptions are re-evaluated and modified. The solutions that we propose in this thesis have consequently been designed to be resistant to continuous visual observation and to achieve certain security guarantees despite the possibility of device compromise.

Furthermore, if these technologies introduce novel human-computer interfaces, they often make previous research on achieving a specific security goal on similar

systems non-applicable. In the case of mixed reality, despite the large body of previous systems security research, the lack of keyboard or touchpad input and the inability to directly show data to other protocol participants motivated us to propose and evaluate novel methods to achieve user authentication and device pairing: two slightly different, but related challenges, which resulted in solutions that have many common points as potential directions for future research.

However, while new technologies can be used by the adversaries to strengthen their attacks, they can also be used to protect existing systems. We have therefore proposed one way in which the mixed reality capabilities of future devices can be used to ensure that user's interaction with a local client remains secure despite potential client compromise, and we look forward to seeing more research in this direction.

Despite the potential of mixed reality systems to profoundly change how users interact with their environment, other electronic devices and among themselves, there has been relatively little research that focuses on the security and privacy challenges related to this emerging core technology. However, given the short duration since the first mixed reality devices became commercially available, this research gap is not surprising.

In this thesis, we have reduced the identified gap by systematizing the related research, tackling several open problems, and highlighting several avenues for future work. We therefore believe that the presented work provides a strong and timely foundation towards making the future mixed reality systems secure and usable.

The most important decision we make is whether we believe we live in a friendly or a hostile universe.

— Albert Einstein

References

[1] Nikola Tesla. "The wonder world to be created by electricity". In: *Manufacturer's Record* 9 (1915), pp. 37–38.

[2] Ivan E Sutherland. "The ultimate display". In: *Multimedia: From Wagner to virtual reality* (1965).

[3] Craig Smith. *Pokemon Go Statistics and Facts*. Accessed December 2018. 2018. URL: https://expandedramblings.com/index.php/pokemon-go-statistics/.

[4] Jacob Kastrenakes. *Battery pack sales doubled after Pokémon Go's release*. Accessed December 2018. 2018. URL: https://www.theverge.com/2016/8/5/12387896/battery-pack-sales-double-after-pokemon-go-release.

[5] Allana G LeBlanc and Jean-Philippe Chaput. "Pokémon Go: A game changer for the physical inactivity crisis?" In: *Preventive medicine* 101 (2017), pp. 235–237.

[6] DavidPhelan. *Tim Cook on the importance of coding and Augmented Reality*. Accessed August 2018. 2017. URL: https://www.independent.co.uk/life-style/gadgets-and-tech/features/apple-tim-cook-boss-brexit-uk-theresa-may-number-10-interview-ustwo-a7574086.html.

[7] Mariella Moon. "Apple has NASA minds working on its AR glasses". In: *Engadget* (2017). Accessed May 2017. URL: https://www.engadget.com/2017/04/25/apple-augmented-reality-glasses-nasa-jeff-norris.

[8] Jon Fingas. *Apple buys the creator of a 'seamless' mixed reality headset*. Accessed October 2018. 2017. URL: https://www.engadget.com/2017/11/21/apple-buys-company-making-mixed-reality-headset/.

[9] Jillian D'Onfro and Jay Yarow. "Google Is Leading A 542 Million Investment In Magic Leap". In: *Business Insider* (Oct. 21, 2014). URL: http://www.businessinsider.com/magic-leap-google-investment-2014-10.

[10] Apple Inc. *ARKit 2*. Accessed November 2018. 2018. URL: https://developer.apple.com/arkit/.

[11] Google Inc. *Tango Augmented Reality Computing Platform*. Accessed May 2017. 2017. URL: https://get.google.com/tango/.

[12] Google Inc. *ARCore*. Accessed November 2018. 2018. URL: https://developers.google.com/ar/.

[13] Facebook Inc. *Spark AR Studio*. Accessed December 2018. 2018. URL: https://sparkar.com/ar-studio.

[14] Microsoft. *Microsoft HoloLens*. Accessed August 2018. 2018. URL: https://www.microsoft.com/en-us/hololens.

[15] Magic Leap. *Magic Leap One*. Accessed November 2018. 2018. URL: https://www.magicleap.com/magic-leap-one.

[16] Meta Vision. *Meta Augmented Reality*. Accessed November 2018. 2018. URL: https://www.metavision.com/.

[17] Tom Warren. *Microsoft reveals Dell and Asus Windows Mixed Reality headsets*. Accessed May 2017. URL: https://www.theverge.com/2017/5/31/15717478/dell-asus-windows-mixed-reality-headsets-features.

[18] Elizabeth Landau. "'Mixed Reality' Technology Brings Mars to Earth". In: *NASA Jet Propulsion Laboratory* (2017). Accessed May 2017. URL: https://www.jpl.nasa.gov/news/news.php?feature=6220.

[19] David Lumb. "Microsoft HoloLens becomes an AR assistant for spinal surgery". In: *Engadget* (2017). Accessed May 2017. URL: https://www.engadget.com/2017/05/05/microsoft-hololens-becomes-an-ar-assistant-for-spinal-surgery.

[20] Gwen Ackerman and Dina Bass. "Israeli Army Prepares Augmented Reality for Battlefield Duty". In: *Bloomberg Technology* (2017). Accessed May 2017. URL: https://www.bloomberg.com/news/articles/2016-08-15/microsoft-s-hololens-technology-adopted-by-israeli-military.

[21] Joshua Brustein. *Microsoft Wins $480 Million Army Battlefield Contract*. Accessed November 2018. 2018. URL: https://www.bloomberg.com/news/articles/2018-11-28/microsoft-wins-480-million-army-battlefield-contract.

[22] Paul Sawers. *Microsoft launches SharePoint Spaces so you can view company content in mixed reality*. Accessed November 2018. 2018. URL: https://venturebeat.com/2018/05/21/microsoft-launches-sharepoint-spaces-so-you-can-view-company-content-in-mixed-reality/.

[23] Kiron Lebeck, Tadayoshi Kohno, and Franziska Roesner. "How to safely augment reality: Challenges and directions". In: *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*. ACM. 2016, pp. 45–50.

[24] Kiron Lebeck et al. "Securing Augmented Reality Output". In: *Proceedings of the 38th IEEE Symposium on Security and Privacy (Oakland 2017)*.

[25] Suman Jana et al. "Enabling Fine-Grained Permissions for Augmented Reality Applications with Recognizers". In: *22nd USENIX Security Symposium (USENIX Security 13)*. Washington, D.C.: USENIX, 2013, pp. 415–430.

[26] Eisa Zarepour et al. "A context-based privacy preserving framework for wearable visual lifeloggers". In: *Pervasive Computing and Communication Workshops (PerCom Workshops), 2016 IEEE International Conference on*. IEEE. 2016, pp. 1–4.

[27] Mohammed Korayem et al. "Enhancing Lifelogging Privacy by Detecting Screens". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: ACM, 2016, pp. 4309–4314.

[28] John Vilk et al. "SurroundWeb: Mitigating Privacy Concerns in a 3D Web Browser". In: *Proceedings of the 2015 IEEE Symposium on Security and Privacy*. SP '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 431–446.

[29] Ivo Sluganovic et al. "Using Reflexive Eye Movements for Fast Challenge-Response Authentication". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: ACM, 2016, pp. 1056–1067.

[30] Ivo Sluganovic et al. "HoloPair: Securing Shared Augmented Reality Using Microsoft HoloLens". In: *33rd Annual Computer Security Applications Conference (ACSAC 2017*. 2017, p. 13.

[31] Ivo Sluganovic et al. "Analysis of Reflexive Eye Movements for Fast Replay-Resistant Biometric Authentication". In: *ACM Transactions on Privacy and Security* 22.1 (Nov. 2018), pp. 1–30.

[32] Ivo Sluganovic et al. "IntegriScreen: Visually Supervising Clients to Continuously Authenticate User Input". In: *Submitted to IEEE European Transactions on Security and Privacy (IEEE Euro SnP)*. IEEE, 2019.

[33] Ivan Martinovic and Ivo Sluganovic. "Augmented Reality Security Assistant". United Kingdom Patent Application No. GB1802739.1. Feb. 2018.

[34] Marc Roeschlin et al. "Generating Secret Keys from Biometric Body Impedance Measurements". In: *ACM CCS Workshop on Privacy in Electronic Society (WPES)*. 2016, p. 10.

[35] Mika Juuti et al. "STASH: Securing transparent authentication schemes using prover-side proximity verification". In: *IEEE International Conference on Sensing, Communication and Networking (SECON)*. 2017, p. 10.

[36] Mario Frank et al. "Using EEG-Based BCI Devices to Subliminally Probe for Private Information". In: *Submitted to ACM CCS Workshop on Privacy in Electronic Society (WPES)*. 2017, p. 10.

[37] Mika Juuti et al. "Implementing Prover-Side Proximity Verification for Strengthening Transparent Authentication". In: *Sensing, Communication, and Networking (SECON), 2017 14th Annual IEEE International Conference on*. IEEE. 2017.

[38] Giulio Lovisotto et al. *Mobile Biometrics in Financial Services: A Five Factor Framework*. Tech. rep. CS-RR-17-03. University of Oxford, Department of Computer Science, Mar. 2017, p. 15.

[39] Ivan Martinovic, Lucas Kello, and Ivo Sluganovic. *Blockchains for Governmental Services: Design Principles, Applications, and Case Studies*. Tech. rep. Oxford Centre for Technology and Global Affairs, 2018.

[40] Google. *Google Glass*. Accessed August 2018. 2018. URL: `https://www.x.company/glass/`.

[41] Facebook. *Oculus Rift*. Accessed November 2018. 2018. URL: `https://www.oculus.com/`.

[42] HTC Corporation. *HTC Vive Pro*. Accessed November 2018. 2018. URL: `https://www.vive.com`.

[43] Peter Graham. *Epson Adds Biometrics to its MOVERIO AR Smart Glasses Platform*. Accessed October 2018. 2018. URL: https://www.vrfocus.com/2018/05/epson-adds-biometrics-to-its-moverio-ar-smart-glasses-platform/.

[44] Dimitris Chatzopoulos et al. "Mobile augmented reality survey: From where we are to where we go". In: *IEEE Access* 5 (2017), pp. 6917–6950.

[45] Marcus Tonnis et al. "Experimental evaluation of an augmented reality visualization for directing a car driver's attention". In: *Mixed and Augmented Reality, 2005. Proceedings. Fourth IEEE and ACM International Symposium on*. IEEE. 2005, pp. 56–59.

[46] In-Ho Choi and Yong-Guk Kim. "Head pose and gaze direction tracking for detecting a drowsy driver". In: *Big Data and Smart Computing (BIGCOMP), 2014 International Conference on*. Jan. 2014, pp. 241–244.

[47] Matt Zeller and Brandon Bray. *HoloLens hardware details*. Accessed July 2018. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details.

[48] Microsoft. *Universal Windows Platform*. Accessed November 2018. 2018. URL: https://docs.microsoft.com/en-us/windows/uwp/.

[49] Alex Turner, Matt Zeller, and Brandon Bray. *HoloLens Gaze*. Accessed August 2018. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/gaze.

[50] Ben Dickson. *How eye tracking will enhance the AR and VR experience*. Accessed August 2018. 2018. URL: https://bdtechtalks.com/2018/06/11/ar-vr-eye-tracking-foveated-rendering/.

[51] Fove. *The World's First Eye Tracking Virtual Reality Handset*. Accessed July 2015. 2015. URL: http://www.getfove.com/.

[52] Devindra Hardawar. *Tobii proves that eye tracking is VR's next killer feature*. Accessed August 2018. 2018. URL: https://www.engadget.com/2018/01/13/tobii-vr-eye-tracking/.

[53] Tony Darin. *7Invensun will provide eye-tracking for many worldwide XR devices*. Accessed November 2018. URL: https://skarredghost.com/2018/08/31/7invensun-will-provide-eye-tracking-for-many-worldwide-xr-devices/.

[54] Chris Wiltz. *Magic Leap One Teardown: A Leap Forward for AR/VR?* Accessed October 2018. URL: https://www.designnews.com/design-hardware-software/magic-leap-one-teardown-leap-forward-arvr/204060129459400.

[55] Christian Moller and Flavio Protasio Ribeiro. *Capacitive sensors for determining eye gaze direction*. US Patent 9,888,843. Feb. 13, 2018.

[56] Macy Bayern. *Eye-tracking tech skyrockets as businesses invest in security and VR applications*. Accessed October 2018. URL: https://www.techrepublic.com/article/eye-tracking-tech-skyrockets-as-businesses-invest-in-security-and-vr-applications/.

[57] Microsoft. *Develop for Windows Mixed Reality Platform*. Accessed October 2018. URL: https://developer.microsoft.com/en-us/windows/mixed-reality.

[58] Unity Inc. *Unity Game Development Platform*. Accessed May 2017. 2017. URL: https://unity3d.com/.

[59] Furkan Tari, Ant Ozok, and Stephen H Holden. "A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords". In: *Proceedings of the second symposium on Usable privacy and security*. ACM. 2006, pp. 56–66.

[60] Jingchao Sun et al. "VISIBLE: Video-Assisted Keystroke Inference from Tablet Backside Motion." In: 2016.

[61] Malin Eiband et al. "Understanding shoulder surfing in the wild: Stories from users and observers". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM. 2017, pp. 4254–4265.

[62] David Cardinal. *Hands Off With Ambarella's Camera-Only Self-Driving Car Tech*. Accessed August 2018. 2018. URL: https://www.extremetech.com/electronics/266694-self-driving-without-lidar-ambarella-camera-eva-road-test.

[63] Rick Barret. *Dashboard cameras sales rising fast due to safety-conscious drivers*. Accessed December 2018. 2018. URL: https://eu.usatoday.com/story/money/cars/2018/02/26/dashboard-cameras-sales-rising-fast-due-safety-conscious-drivers/373631002/.

[64] The Statistics Portal. *CCTV camera market size worldwide from 2011 to 2016 (in billion U.S. dollars), by product category*. Accessed November 2018. 2018. URL: https://www.statista.com/statistics/525951/global-cctv-camera-market-size-by-product/.

[65] Aniket Bera and Dinesh Manocha. "Reach-realtime crowd tracking using a hybrid motion model". In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 740–747.

[66] Nest. *Nest Indoor Security Camera*. Accessed October 2018. 2018. URL: https://nest.com/uk/cameras/nest-cam-indoor/overview/.

[67] Lucas Matney. *Piccolo is building a gesture-based smart home "vision assistant"*. Accessed November 2018. 2018. URL: https://techcrunch.com/2018/02/28/piccolo-is-building-a-gesture-based-smart-home-vision-assistant/.

[68] Béla Genge and Călin Enăchescu. "ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services". In: *Security and communication networks* 9.15 (2016), pp. 2696–2714.

[69] Aerix Drones. *Aerix Vividius - The World's Smallest FPV Drone*. Accessed November 2018. 2018. URL: https://aerixdrones.com/products/vidius-the-worlds-smallest-fpv-drone.

[70] Forbes. *Over A Million Coders Targeted By Chrome Extension Hack*. accessed June 2018. URL: https://www.forbes.com/sites/leemathews/2017/08/03/over-a-million-coders-targeted-by-chrome-extension-hack.

[71] Arstechnica. *Adware vendors buy Chrome Extensions to send ad- and malware-filled updates*. accessed June 2018. URL: https://arstechnica.com/information-technology/2014/01/malware-vendors-buy-chrome-extensions-to-send-adware-filled-updates.

[72] EU cyber security agency (ENISA). *High Roller online bank robberies reveal security gaps*. Accessed July 2018. 2018. URL: https://tinyurl.com/ENISA-online-bank-robberies.

[73] Hamad Binsalleeh et al. "On the analysis of the zeus botnet crimeware toolkit". In: *Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on*. IEEE. 2010, pp. 31–38.

[74] Wired. *Uber Paid Off Hackers to Hide a 57-Million User Data Breach*. accessed July 2018. 2017. URL: https://www.wired.com/story/uber-paid-off-hackers-to-hide-a-57-million-user-data-breach.

[75] Gurchetan S Grewal et al. "Du-vote: Remote electronic voting with untrusted computers". In: *Computer Security Foundations Symposium (CSF), 2015 IEEE 28th*. IEEE. 2015, pp. 155–169.

[76] Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. "Security and privacy challenges in industrial internet of things". In: *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*. IEEE. 2015, pp. 1–6.

[77] M. Rushanan et al. "SoK: Security and Privacy in Implantable Medical Devices and Body Area Networks". In: *2014 IEEE Symposium on Security and Privacy (SP)*. May 2014, pp. 524–539.

[78] Paul Milgram et al. "Augmented reality: A class of displays on the reality-virtuality continuum". In: *Telemanipulator and telepresence technologies*. Vol. 2351. International Society for Optics and Photonics. 1995, pp. 282–293.

[79] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. "Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR". In: *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society. 2008, pp. 193–202.

[80] Franziska Roesner, Tadayoshi Kohno, and David Molnar. "Security and privacy for augmented reality systems". In: *Communications of the ACM* 57.4 (2014), pp. 88–96.

[81] Jaybie A de Guzman, Kanchana Thilakarathna, and Aruna Seneviratne. "Security and Privacy Approaches in Mixed Reality: A Literature Survey". In: *arXiv preprint arXiv:1802.05797* (2018).

[82] Alessandro Acquisti, Ralph Gross, and Fred Stutzman. *Faces of facebook: Privacy in the age of augmented reality*. 2011.

[83] Tamara Denning, Zakariya Dehlawi, and Tadayoshi Kohno. "In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2014, pp. 2377–2386.

[84] Mohammed Korayem et al. "Screenavoider: Protecting computer screens from ubiquitous cameras". In: *arXiv preprint arXiv:1412.0008* (2014).

[85] Suman Jana, Arvind Narayanan, and Vitaly Shmatikov. "A scanner darkly: Protecting user privacy from perceptual applications". In: *Security and Privacy (SP), 2013 IEEE Symposium on*. IEEE. 2013, pp. 349–363.

[86] Robert Templeman et al. "PlaceAvoider: Steering First-Person Cameras away from Sensitive Spaces." In: *Annual Network & Distributed System Security Symposium (NDSS)*. 2014, pp. 23–26.

[87] Sarah M Lehman and Chiu C Tan. "PrivacyManager: An access control framework for mobile augmented reality applications". In: *Communications and Network Security (CNS), 2017 IEEE Conference on*. IEEE. 2017, pp. 1–9.

[88] Nisarg Raval et al. "Markit: Privacy markers for protecting visual secrets". In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM. 2014, pp. 1289–1295.

[89] Franziska Roesner et al. "World-driven access control for continuous sensing". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2014, pp. 1169–1181.

[90] Lucas Silva Figueiredo et al. "Prepose: Privacy, Security, and Reliability for Gesture-Based Programming". In: *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE. 2016, pp. 122–137.

[91] Parmy Olson. *Meet The Virtual Headset That Uses Your Eye As A Screen*. Accessed November 2018. 2018. URL: `https://www.forbes.com/sites/parmyolson/2013/11/20/meet-the-virtual-headset-that-uses-your-eye-as-a-screen/`.

[92] Avegant. *Avegant Light Field Technology*. Accessed November 2018. 2018. URL: `https://www.avegant.com/`.

[93] Tadayoshi Kohno et al. *Display leakage and transparent wearable displays: Investigation of risk, root causes, and defenses*. Tech. rep. Microsoft Research, Tech. Rep., February 2015.[Online]. Available https://www.microsoft.com/en-us/research/publication/display-leakage-and-transparent-wearable-displays-investigation-ofrisk-root-causes-and-defenses.

[94] Silicon Micro Display. *The world's first and only consumer head-mounted display to offer native 1080p personal viewing experience*. Accessed October 2018. URL: `http://www.siliconmicrodisplay.com/st1080.html`.

[95] Eric E Sabelman and Roger Lam. "The real-life dangers of augmented reality". In: *IEEE Spectrum* 52.7 (2015), pp. 48–53.

[96] Stefano Baldassi et al. "Challenges and New Directions in Augmented Reality, Computer Security, and Neuroscience–Part 1: Risks to Sensation and Perception". In: *arXiv preprint arXiv:1806.10557* (2018).

[97] Kiron Lebeck et al. "Securing augmented reality output". In: *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE. 2017, pp. 320–337.

[98] Surin Ahn et al. "Adaptive Fog-Based Output Security for Augmented Reality". In: *Proceedings of the 2018 Morning Workshop on Virtual Reality and Augmented Reality Network*. ACM. 2018, pp. 1–6.

[99] Moni Naor and Adi Shamir. "Visual cryptography". In: *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer. 1994, pp. 1–12.

[100] Andrea G Forte et al. “EyeDecrypt—Private interactions in plain sight”. In: *International Conference on Security and Cryptography for Networks*. Springer. 2014, pp. 255–276.

[101] Sarah J Andrabi, Michael K Reiter, and Cynthia Sturton. “Usability of Augmented Reality for Revealing Secret Messages to Users but Not Their Devices.” In: *SOUPS*. Vol. 2015. 2015, pp. 89–102.

[102] Feng Huang et al. “Piano ar: A markerless augmented reality based piano teaching system”. In: *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2011 International Conference on*. Vol. 2. IEEE. 2011, pp. 47–52.

[103] Anindya Maiti, Murtuza Jadliwala, and Chase Weber. “Preventing shoulder surfing using randomized augmented reality keyboards”. In: *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*. IEEE. 2017, pp. 630–635.

[104] Christian Winkler et al. “Glass unlock: Enhancing security of smartphone unlocking through leveraging a private near-eye display”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM. 2015, pp. 1407–1410.

[105] Simon AS Briggs et al. *Secure passcode entry using mobile device with augmented reality capability*. US Patent App. 14/887,861. May 2016.

[106] Hassan Khan, Urs Hengartner, and Daniel Vogel. “Augmented Reality-based Mimicry Attacks on Behaviour-Based Smartphone Authentication”. In: *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM. 2018, pp. 41–53.

[107] Nolen Scaife, Christian Peeters, and Patrick Traynor. “Fear the reaper: characterization and fast detection of card skimmers”. In: *27th {USENIX} Security Symposium ({USENIX} Security 18)*. 2018, pp. 1–14.

[108] Puneet Batta, Trevor Miranda, and Jacob Thomas. *Detection of an unauthorized wireless communication device*. US Patent App. 15/800,616. Mar. 2018.

[109] Kiron Lebeck et al. “Towards Security and Privacy for Multi-User Augmented Reality: Foundations with End Users”. In: *Towards Security and Privacy for Multi-User Augmented Reality: Foundations with End Users*. IEEE. 2018.

[110] Ethan Gaebel et al. “Looks Good To Me: Authentication for Augmented Reality”. In: *Proceedings of the 6th International Workshop on Trustworthy Embedded Devices*. TrustED ’16. Vienna, Austria: ACM, 2016, pp. 57–67.

[111] Confident Technologies. *Smartphone Users Choose Convenience Over Security*. 2011. URL: http://confidenttechnologies.com/news%5C_events/survey-shows-smartphone-users-choose-convenience-over-security.

[112] Karen Renaud. “Blaming noncompliance is too convenient: What really causes information breaches?” In: *IEEE Security & Privacy* 10.3 (2012), pp. 57–63.

[113] Richard Shay et al. “Can long passwords be secure and usable?” In: *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI ’14* (2014), pp. 2927–2936.

[114] Joseph Bonneau et al. "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes". In: *Proceedings - IEEE Symposium on Security and Privacy* 2012 (2012), pp. 553–567.

[115] DARPA. *Broad Agency Announcement: Active Authentication*. Tech. rep. DARPA, 2012.

[116] Rene Millman. *New version of L0phtCrack makes cracking Windows passwords easier than ever*. Accessed October 2018. 2016. URL: https://www.scmagazineuk.com/new-version-l0phtcrack-makes-cracking-windows-passwords-easier-ever/article/1476745.

[117] Samuel Gibbs. *Dropbox hack leads to leaking of 68m user passwords on the internet*. Accessed October 2018. 2016. URL: https://www.theguardian.com/technology/2016/aug/31/dropbox-hack-passwords-68m-data-breach.

[118] Adam J Aviv et al. "Towards Baselines for Shoulder Surfing on Mobile Authentication". In: *Proceedings of the 33rd Annual Computer Security Applications Conference*. ACM. 2017, pp. 486–498.

[119] Stephen Braun. *NSA: Co-worker provided digital key to Snowden*. Accessed October 2018. 2014. URL: https://eu.usatoday.com/story/news/nation/2014/02/14/nsa-snowden-co-worker/5478945.

[120] Ceenu George et al. "Seamless and secure vr: Adapting and evaluating established authentication systems for virtual reality". In: *Annual Network & Distributed System Security Symposium (NDSS)*. 2017.

[121] Ruide Zhang et al. "AugAuth: Shoulder-surfing resistant authentication for augmented reality". In: *Communications (ICC), 2017 IEEE International Conference on*. IEEE. 2017, pp. 1–6.

[122] Dhruv Kumar Yadav et al. "Design and analysis of shoulder surfing resistant PIN based authentication mechanisms on Google Glass". In: *International conference on financial cryptography and data security*. Springer. 2015, pp. 281–297.

[123] Lin Hong et al. "Identity Authentication Using Fingerprints". In: *Proceedings of the First International Conference on Audio- and Video-Based Biometric Person Authentication*. AVBPA '97. London, UK, UK: Springer-Verlag, 1997, pp. 103–110.

[124] J. Daugman. "How Iris Recognition Works". In: *IEEE Transactions on Circuits and Systems for Video Technology* 14.1 (Jan. 2004), pp. 21–30.

[125] John Daugman. "New methods in iris recognition." In: *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society* 37.5 (2007), pp. 1167–1175.

[126] Mario Frank et al. "Touchalytics : On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication". In: (2012), pp. 1–20.

[127] L Wang et al. "Silhouette analysis-based gait recognition for human identification". In: *Pattern Analysis and Machine ...* 25.12 (2003), pp. 1505–1518.

[128] Marina Blanton and William M P Hudelson. "Biometric-Based Non-transferable Anonymous Credentials". In: *ICICS*. 2009, pp. 165–180.

[129] Alex Hern. *Hacker fakes German minister's fingerprints using photos of her hands*. Accessed November 2018. 2014. URL: https://www.theguardian.com/technology/2014/dec/30/hacker-fakes-german-ministers-fingerprints-using-photos-of-her-hands.

[130] Prakash; Panjwani. "Crowdsourcing Attacks on Biometric Systems". In: *SOUPS*. 2014, pp. 257–269.

[131] A. Boehm et al. "SAFE: Secure authentication with Face and Eyes". In: *Privacy and Security in Mobile Systems (PRISMS), 2013 International Conference on*. June 2013, pp. 1–8.

[132] Fred Richardson, Douglas Reynolds, and Najim Dehak. "Deep neural network approaches to speaker and language recognition". In: *IEEE Signal Processing Letters* 22.10 (2015), pp. 1671–1675.

[133] Tomi Kinnunen and Haizhou Li. "An overview of text-independent speaker recognition: From features to supervectors". In: *Speech communication* 52.1 (2010), pp. 12–40.

[134] Federico Alegre et al. "On the vulnerability of automatic speaker recognition to spoofing attacks with artificial signals". In: *EUSIPCO 2012, 20th European Signal Processing Conference*. 2012.

[135] Tomi Kinnunen et al. "Reddots replayed: A new replay spoofing attack corpus for text-dependent speaker verification research". In: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE. 2017, pp. 5395–5399.

[136] Jagmohan Chauhan et al. "BreathPrint: Breathing acoustics-based user authentication". In: *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM. 2017, pp. 278–291.

[137] Stefan Schneegass, Youssef Oualil, and Andreas Bulling. "SkullConduct: Biometric user identification on eyewear computers using bone conduction through the skull". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM. 2016, pp. 1379–1384.

[138] Sugang Li et al. "Whose move is it anyway? Authenticating smart wearable devices using unique head movement patterns". In: *Pervasive Computing and Communications (PerCom), 2016 IEEE International Conference on*. IEEE. 2016, pp. 1–9.

[139] Tahrima Mustafa et al. "Unsure How to Authenticate on Your VR Headset?: Come on, Use Your Head!" In: *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics*. IWSPA '18. Tempe, AZ, USA: ACM, 2018, pp. 23–30.

[140] Yiran Shen et al. "GaitLock: Protect virtual and augmented reality headsets using gait". In: *IEEE Transactions on Dependable and Secure Computing* (2018).

[141] Bendik B Mjaaland, Patrick Bours, and Danilo Gligoroski. "Walk the walk: attacking gait biometrics by imitation". In: *International Conference on Information Security*. Springer. 2010, pp. 361–380.

[142] Michael Sherman et al. “User-generated free-form gestures for authentication: Security and memorability”. In: *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM. 2014, pp. 176–189.

[143] Napa Sae-Bae et al. “Biometric-rich gestures: a novel approach to authentication on multi-touch devices”. In: *proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2012, pp. 977–986.

[144] Kam Lai, Janusz Konrad, and Prakash Ishwar. “Towards gesture-based user authentication”. In: *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*. IEEE. 2012, pp. 282–287.

[145] Microsoft Kinect. *Kinect for Windows*. Accessed Decermber 2018. 2018. URL: `https://developer.microsoft.com/en-us/windows/kinect`.

[146] Jing Tian et al. “KinWrite: Handwriting-Based Authentication Using Kinect.” In: *Annual Network & Distributed System Security Symposium (NDSS)*. Vol. 93. 2013, p. 94.

[147] Leap Motion. *The Most Advanced Hand Tracking on Earth*. Accessed December 2018. 2018. URL: `https://www.leapmotion.com/`.

[148] Ilhan Aslan et al. “Design and exploration of mid-air authentication gestures”. In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 6.3 (2016), p. 23.

[149] Md Tanvir Islam Aumi and Sven Kratz. “Airauth: evaluating in-air hand gestures for authentication”. In: *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. ACM. 2014, pp. 309–318.

[150] Wenyuan Xu et al. “Which One to Go: Security and Usability Evaluation of Mid-Air Gestures”. In: *arXiv preprint arXiv:1811.10168* (2018).

[151] John Brooke. “SUS: A quick and dirty usability scale”. In: *Usability Evaluation in Industry* (1996).

[152] Dennis Guse. *Gesture-based User Authentication for Mobile Devicesusing Accelerometer and Gyroscope*. Tech. rep. Berlin Institute of Technology, 2011.

[153] Imtiaj Ahmed et al. “Checksum gestures: continuous gestures as an out-of-band channel for secure pairing”. In: *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2015, pp. 391–401.

[154] Jozef Kasprowski, Pawel; Ober. “Eye Movements in Biometrics”. In: *Biometrics* 3087 / 200 (2003), pp. 248–258.

[155] Pawel Kasprowski, Oleg V. Komogortsev, and Alex Karpov. “First eye movement verification and identification competition at BTAS 2012”. In: *2012 IEEE 5th International Conference on Biometrics: Theory, Applications and Systems, BTAS 2012* Btas (2012), pp. 195–202.

[156] Simon Eberz et al. “Preventing Lunchtime Attacks: Fighting Insider Threats With Eye Movement Biometrics”. In: *Proceedings of the 2015 Networked and Distributed System Security Symposium*. 2015.

[157] William Welby Abbott and Aldo Ahmed Faisal. “Ultra-low-cost 3D gaze estimation: an intuitive high information throughput compliment to direct brain–machine interfaces”. In: *Journal of neural engineering* 9.4 (2012), p. 046016.

[158] Katharine Byrne. *MSI & Tobii join forces to create eye-tracking gaming laptop.* Accessed June 2015. 2015. URL: http://www.expertreviews.co.uk/laptops/1403340/msi-tobii-join-forces-to-create-eye-tracking-gaming-laptop.

[159] Emiliano Miluzzo, Tianyu Wang, and Andrew T Campbell. "EyePhone: Activating Mobile Phones with Your Eyes". In: *Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld)* (2010).

[160] Sally Davies. *GM to launch cars that can pick up on distracted driving.* Accessed March 2015. New York, New York, USA, Sept. 2014.

[161] Ami Klin et al. "Visual fixation patterns during viewing of naturalistic social situations as predictors of social competence in individuals with autism." In: *Archives of general psychiatry* 59 (2002), pp. 809–816.

[162] Tony Poitschke et al. "Gaze-based interaction on multiple displays in an automotive environment". In: *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on.* IEEE. 2011, pp. 543–548.

[163] Richard A Abrams, David E Meyer, and Sylvan Kornblum. "Speed and accuracy of saccadic eye movements: characteristics of impulse variability in the oculomotor system." In: *Journal of experimental psychology. Human perception and performance* 15.3 (1989).

[164] Kenneth Holmqvist, Marcus Nyström, and Richard Andersson. *Eye tracking: A comprehensive guide to methods and measures.* Oxford University Press, 2011, pp. 1–702.

[165] Michael F Land. "Oculomotor behaviour in vertebrates and invertebrates". In: *The Oxford handbook of eye movements* 1 (2011).

[166] Robin Walker et al. "Control of voluntary and reflexive saccades". In: *Experimental Brain Research* 130.4 (Feb. 2000), pp. 540–544.

[167] Jennie E S Choi, Pavan a Vaswani, and Reza Shadmehr. "Vigor of movements and the cost of time in decision making." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 34.4 (2014), pp. 1212–23.

[168] T. Bahill, M. R. Clark, and L. Stark. "The main sequence, a tool for studying human eye movements". In: *Mathematical Biosciences* 24.3-4 (1975), pp. 191–204.

[169] Monica S. Castelhano and John M. Henderson. "Stable individual differences across images in human saccadic eye movements." In: *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale* 62.1 (2008), pp. 1–14.

[170] Francesco Di Russo, Sabrina Pitzalis, and Donatella Spinelli. "Fixation stability and saccadic latency in elite shooters". In: *Vision Research* 43.17 (2003), pp. 1837–1845.

[171] Terry Bahill and Tom Laritz. "Why Can't Batters Keep Their Eyes on the Ball?" In: *American Scientist* May - June (1984), pp. 249–253.

[172] Lawrence R Gottlob, Mark T Fillmore, and Ben D Abroms. "Age-group differences in saccadic interference." In: *The journals of gerontology. Series B, Psychological sciences and social sciences* 62.2 (2007), P85–P89.

[173] O. V. Kolesnikova et al. “Effects of visual environment complexity on saccade performance in humans with different functional asymmetry profiles”. In: *Neuroscience and Behavioral Physiology* 40.8 (2010), pp. 869–876.

[174] Petroc Sumner. “Determinants of saccade latency”. In: *Oxford handbook of eye movements*. Vol. 22. March. 2011, pp. 411–424.

[175] Yun Zhang, Zheru Chi, and Dagan Feng. “An Analysis of Eye Movement Based Authentication Systems”. In: *International Conference on Mechanical Engineering and Technology (ICMET-London 2011)* (2011), pp. 799–802.

[176] Usman Saeed. “Eye movements during scene understanding for biometric identification”. In: *Pattern Recognition Letters* (2015).

[177] Manu Kumar et al. “Reducing Shoulder-surfing by Using Gaze-based Password Entry”. In: *Proceedings of the 3rd Symposium on Usable Privacy and Security*. SOUPS ’07. Pittsburgh, Pennsylvania: ACM, 2007, pp. 13–19.

[178] Justin Weaver, Kenrick Mock, and Bogdan Hoanca. “Gaze-based password authentication through automatic clustering of gaze points”. In: *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics* (2011), pp. 2749–2754.

[179] Andreas Bulling, Florian Alt, and Albrecht Schmidt. “Increasing the security of gaze-based cued-recall graphical passwords using saliency masks”. In: *CHI*. 2012.

[180] 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. Mountain View, California: ACM, 2009, 7:1–7:12.

[181] Tomasz Kocejko and Jerzy Wtorek. “Gaze pattern lock for elders and disabled”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 7339 LNBI. 2012, pp. 589–602.

[182] Virginio Cantoni et al. “GANT: Gaze analysis technique for human identification”. In: *Pattern Recognition* (2014).

[183] Chiara Galdi et al. “A new gaze analysis based soft-biometric”. In: *Lecture Notes in Computer Science* 7914 LNCS (2013), pp. 136–144.

[184] Ioannis Rigas, George Economou, and Spiros Fotopoulos. “Biometric identification based on the eye movements and graph matching techniques”. In: *Pattern Recognition Letters* 33.6 (2012), pp. 786–792.

[185] Ioannis Rigas and Oleg V Komogortsev. “Biometric Recognition via Probabilistic Spatial Projection of Eye Movement Trajectories in Dynamic Visual Environments”. In: *IEEE TIFS* 9.10 (2014), pp. 1743–1754.

[186] Oleg V Komogortsev et al. “Biometric Identification via an Oculomotor Plant Mathematical Model”. In: *Eye Tracking Research & Applications (ETRA) Symposium* (2010), pp. 57–60.

[187] Corey D Holland and Oleg V Komogortsev. “Biometric identification via eye movement scanpaths in reading”. In: *2011 International Joint Conference on Biometrics, IJCB 2011* (2011).

[188] Corey D Holland and Oleg V Komogortsev. "Complex eye movement pattern biometrics: The effects of environment and stimulus". In: *IEEE Transactions on Information Forensics and Security* 8.12 (2013), pp. 2115–2126.

[189] Oleg V Komogortsev, Alexey Karpov, and Corey D Holland. "Attack of Mechanical Replicas : Liveness Detection With Eye Movements". In: *IEEE TIFS* 10.4 (2015), pp. 716–725.

[190] Youming Zhang, Jorma Laurikkala, and Martti Juhola. "Biometric Verification of a Subject with Eye Movements". In: *Int. J. Biometrics* 6.1 (Mar. 2014), pp. 75–94.

[191] Pawel Kasprowski. "Human Identification Using Eye Movements". In: *Institute of Computer Science* (2004), pp. 1–111.

[192] Pawel Kasprowski. "The Second Eye Movements Verification and Identification Competition". In: *IEEE & IAPR International Joint Conference on Biometrics*. 2014.

[193] Pawel Kasprowski and Katarzyna Harezlak. "Fusion of eye movement and mouse dynamics for reliable behavioral biometrics". In: *Pattern Analysis and Applications* 21.1 (2018), pp. 91–103.

[194] Yongtuo Zhang et al. "Continuous Authentication Using Eye Movement Response of Implicit Visual Stimuli". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.4 (2018), p. 177.

[195] Simon Eberz et al. "Looks Like Eve: Exposing Insider Threats Using Eye Movement Biometrics". In: *ACM Transactions on Privacy and Security* 19.1 (June 2016), 1:1–1:31.

[196] Marcus Nyström and Kenneth Holmqvist. "An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data". English. In: *Behavior Research Methods* 42.1 (2010), pp. 188–204.

[197] Mario Frank et al. "Touchalytics: On the Applicability of Touchscreen Input As a Behavioral Biometric for Continuous Authentication". In: *Trans. Info. For. Sec.* 8.1 (Jan. 2013), pp. 136–148.

[198] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine Learning* 20 (1995), pp. 273–297.

[199] SensoMotoric Instruments GmbH. *SMI RED500 Technical Specification*. Tech. rep. Teltow, Germany: SensoMotoric Instruments GmbH, 2011, p. 1. URL: `http://medcontent.metapress.com/index/A65RM03P4874243N.pdf`.

[200] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823.

[201] Gary B. Huang et al. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. rep. 07-49. University of Massachusetts, Amherst, Oct. 2007.

[202] Jiankang Deng et al. "Arcface: Additive angular margin loss for deep face recognition". In: *arXiv preprint arXiv:1801.07698* (2018).

[203] Apple Security. *Face ID Security Overview*. Accessed December 2018. 2017. URL: `https://www.apple.com/business/site/docs/FaceID_Security_Guide.pdf`.

[204] Jee-weon Jung et al. "RawNet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification". In: *arXiv preprint arXiv:1904.08104* (2019).

[205] A. Nagrani, J. S. Chung, and A. Zisserman. "VoxCeleb: a large-scale speaker identification dataset". In: *INTERSPEECH*. 2017.

[206] Apple Siri Team. *Personalized Hey Siri*. Accessed December 2018. Apr. 2018. URL: `https://machinelearning.apple.com/2018/04/16/personalized-hey-siri.html`.

[207] Bernadette Dorizzi et al. "Fingerprint and on-line signature verification competitions at ICB 2009". In: *International Conference on Biometrics*. Springer. 2009, pp. 725–732.

[208] Mahmood Sharif et al. "Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: ACM, 2016, pp. 1528–1540.

[209] Yi Xu et al. "Virtual U: Defeating Face Liveness Detection by Building Virtual Models from Your Public Photos". In: *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, 2016, pp. 497–512.

[210] 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. Mountain View, California, USA: ACM, 2009, 11:1–11:12.

[211] Nicholas Carlini et al. "Hidden Voice Commands". In: *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, 2016, pp. 513–530.

[212] Danny Dolev and Andrew Yao. "On the security of public key protocols". In: *IEEE Transactions on information theory* 29.2 (1983), pp. 198–208.

[213] Serge Vaudenay. "Secure Communications over Insecure Channels Based on Short Authenticated Strings". In: *Advances in Cryptology – CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 309–326.

[214] Joshua Tan et al. "Can Unicorns Help Users Compare Crypto Key Fingerprints?" In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA, 2017, pp. 3787–3798.

[215] Matej Serbec. *Application Development for Augmented Reality Platforms*. Tech. rep. Faculty of Engineering and Computing, University of Zagreb, 2017.

[216] Michael Farb et al. "SafeSlinger: easy-to-use and secure public-key exchange". In: *The 19th Annual International Conference on Mobile Computing and Networking, MobiCom'13, Miami, FL, USA, September 30 - October 04, 2013*. 2013, pp. 417–428.

[217] Aaron Bangor, Philip Kortum, and James Miller. "Determining what individual SUS scores mean: Adding an adjective rating scale". In: *Journal of usability studies* 4.3 (2009), pp. 114–123.

[218] Alfred Kobsa et al. “Serial Hook-ups: A Comparative Usability Study of Secure Device Pairing Methods”. In: *Proceedings of the 5th Symposium on Usable Privacy and Security*. SOUPS ’09. Mountain View, California, USA, 2009, 10:1–10:12.

[219] R. Mayrhofer and H. Gellersen. “Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices”. In: vol. 8. 6. June 2009, pp. 792–806.

[220] Yeongjin Jang et al. “Gyrus: A Framework for User-Intent Monitoring of Text-based Networked Applications.” In: *Annual Network & Distributed System Security Symposium (NDSS)*. 2014.

[221] Weidong Cui, R. H. Katz, and Wai-tian Tan. “Design and implementation of an extrusion-based break-in detector for personal computers”. In: *21st Annual Computer Security Applications Conference (ACSAC’05)*.

[222] Ramakrishna Gummadi et al. “Not-a-Bot: Improving Service Availability in the Face of Botnet Attacks”. In: *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation 2009*.

[223] Google, Ray Smith, and Hewlett-Packard. *Tesseract Open Source OCR Engine*. accessed July 2018. URL: `https://github.com/tesseract-ocr/`.

[224] Xiang Zhang, Stephan Fronz, and Nassir Navab. “Visual marker detection and decoding in ar systems: A comparative study”. In: *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*. IEEE Computer Society. 2002, p. 97.

[225] Google. *Text Recognition API Overview*. accessed June 2018. 2017. URL: `https://developers.google.com/vision/android/text-overview`.

[226] OpenCV. *Open Source Computer Vision Library*. accessed June 2018. 2018. URL: `https://opencv.org`.

[227] OpenCV. *Perspective Transform*. Accessed May 2018. URL: `https://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html#perspectivetransform`.

[228] Apache. *Apache Tomcat*. accessed July 2018. URL: `https://tomcat.apache.org/`.

[229] Selenium. *Selenium WebDriver: A browser automation framework and ecosystem*. accessed July 2018. URL: `https://github.com/SeleniumHQ/selenium`.

[230] Anna Pereira et al. “The effect of keyboard key spacing on typing speed, error, usability, and biomechanics: Part 1”. In: *Human factors* 55.3 (2013), pp. 557–566.

[231] Anna Maria Feit, Daryl Weir, and Antti Oulasvirta. “How we type: Movement strategies and performance in everyday typing”. In: *Proceedings of the 2016 chi conference on human factors in computing systems*. ACM. 2016, pp. 4262–4273.

[232] Kyle Krafka et al. “Eye tracking for everyone”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2176–2184.

[233] Thomas Weigold and Alain Hiltgen. “Secure confirmation of sensitive transaction data in modern Internet banking services”. In: *Internet Security (WorldCIS), 2011 World Congress on*. IEEE. 2011, pp. 125–132.

[234] S. Kiljan, H. Vranken, and M. Van Eekelen. "What You Enter Is What You Sign: Input Integrity in an Online Banking Environment". In: *2014 Workshop on Socio-Technical Aspects in Security and Trust.*

Appendices

A
Usability Evaluation

A.1 User Study Instructions

The instructions that were given to participants of the *IntegriScreen* system are as follows:

"Thank you for accepting to take part in our experimental user study.

In this study, you will simulate several online banking payments (using provided test data). While you do this, your input will be protected by *IntegriScreen*, an application that we are currently developing.

IntegriScreen aims to secure users who are sending data to remote servers against the malware that might be present on their computers. In essence, the app recognizes what you type on the computer screen and sends this data directly to the remote server (a test banking server in this experiment). By comparing the data stream received from the computer and the stream from the mobile app, the server can verify if the computer is acting honestly. This system can thus prevent adversaries from, e.g. modifying your transaction details or creating arbitrary transactions."

***IntegriScreen* usage instructions:**

1. Press "START" after loading the form on the computer.
2. When the app shows "Everything OK", you can input or modify the values on the computer.
3. If the app detects unexpected behavior on the screen, it will make a sound, show "Stop input!", and indicate the offending element in red rectangle.
    - The "Expected" and "Detected" values will be shown in different colors on the screen.
    - If you believe this is a result of misrecognition, simply correct the values.
    - Otherwise, abort your input by clicking "RESET"
4. When you finish with input, click "Submit" on the computer, and then click "SUBMIT" on the mobile device.

A.2 System Usability Scale Questions

The System Usability Scale [151] is a widely used 10 statement questionnaire, with Likert-scale answers (0 - "Strongly disagree" to 4 - "Strongly agree"). The overall usability score for an individual user is computed by summing the answers to odd-positioned questions (Q1, Q3, ...) and subtracting the answers on even-positioned questions (Q2, Q4, ...). The score is finally centered and scaled to [0, 100] by adding 20 and multiplying the resulting value with 2.5.

The 10 SUS questions are:

Q1: I think that I would like to use this system frequently.

Q2: I found the system unnecessarily complex.

Q3: I thought the system was easy to use.

Q4: I think that I would need the support of a technical person to be able to use this system.

Q5: I found the various functions in this system were well integrated.

Q6: I thought there was too much inconsistency in this system.

Q7: I would imagine that most people would learn to use this system very quickly.

Q8: I found the system very cumbersome to use.

Q9: I felt very confident using the system.

Q10: I needed to learn a lot of things before I could get going with this system.