Looks Good To Me: Authentication for Augmented Reality

Ethan Gaebel
Virginia Tech
Falls Church, VA
egaebel@vt.edu

Ning Zhang
Virginia Tech
Falls Church, VA
ningzh@vt.edu

Wenjing Lou
Virginia Tech
Falls Church, VA
wjlou@vt.edu

Tom Hou
Virginia Tech
Blacksburg, VA
thou@vt.edu

ABSTRACT

Augmented reality is poised to become the next dominant computing paradigm over the next decade. With promises of three-dimensional graphics and interactive interfaces, augmented reality experiences will rival the very best science fiction novels. This breakthrough also brings in unique challenges on how users can authenticate one another to share rich content between augmented reality headsets. Traditional authentication protocols fall short when there is no common central entity or when access to the central authentication server is not available or desirable. Looks Good To Me (LGTM) is an authentication protocol that leverages the unique hardware and context provided with augmented reality headsets to bring innate human trust mechanisms into the digital world to solve authentication in a usable and secure way. LGTM works over point to point wireless communication so users can authenticate one another in any circumstance and is designed with usability at its core, requiring users to perform only two actions: one to initiate and one to confirm. Users intuitively authenticate one another, using seemingly only each other's faces, but under the hood LGTM uses a combination of facial recognition and wireless localization to bootstrap trust from a wireless signal, to a location, to a face, for secure and extremely usable authentication.

1. INTRODUCTION

Augmented reality (AR) is one of the most exciting new technologies on the horizon, promising three-dimensional interfaces that one or more users can directly interact with. Users will be able to interact with these rich digital interfaces while remaining aware of and responsive to their physical surroundings. These semi-immersive interfaces are commonly provided via head-mounted displays (HMDs) with translucent lenses that render digital content in two or three dimensions on top of the real world. Many of these HMDs will be stand-alone computers equipped with powerful processors, wireless communications, one or more high resolution cameras, and depth sensors. This sophisticated array of hardware is used to provide fully interactive AR which requires: hand tracking, intense graphics processing, and environment mapping.

Users of AR headsets have the same need that users of any other consumer electronic device have today: to send content to others. With AR, this content will usually be far richer, and thus larger, than we've seen on other platforms, since users will be creating, viewing, and working with three-dimensional objects, which inherently have more information associated with them than their equivalent two-dimensional representations. Users will be sharing three-dimensional scene captures, three-dimensional engineering part diagrams, and entire rooms full of three-dimensional digital objects. It's also expected that users will share a large amount of content face-to-face since this will provide 3D objects that both users can see and/or interact with in real time. The current parallel to face-to-face digital sharing in AR is using a smart phone to show a video or picture to someone. The sharing action is the same, but the sharing medium has switched from physically displaying one's screen to sharing the content across two users' devices. We expect this type of sharing to increase with AR as much of the physical world's in-person content exchanges transition to the digital world, like so many other content streams in the past.

Content sharing between users in close proximity is an interesting and well-studied problem [25, 29, 39, 44]. Local sharing stands in contrast to today's typical content sharing schemes which take advantage of preexisting infrastructures such as cell towers, wireless access points, and the Internet. This makes sense considering the most common use case for digitally sharing content currently is to share it with those who aren't present. Today, you wouldn't send someone standing next to you a video, you would show it to them on your device. But with AR you would send someone sitting next to you a video so that the two of you can watch it together through your respective headsets.

This type of local sharing deserves separate consideration from the general content sharing problem, it will be worthwhile to explore and design for the specific conditions that separate local sharing from remote sharing. By sharing con-

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

tent locally, network resources are saved and money may be saved if charge-by-data usage plans are being used for Internet access. The Microsoft HoloLens [5], a new AR headset, recognizes the value in this and has separate support for local sharing vs. remote sharing [37].

When working in a localized communications scenario where there is no central authority, authentication can be tricky. It will often be the case that two users have not used their devices to communicate before, meaning there will be no preexisting security context between the two devices such as a pin, key, token, etc. This problem is commonly known as device pairing and it is well studied [14, 15, 17, 18, 26, 34, 35, 41, 44, 46, 47, 52]. In this scenario authentication must be bootstrapped, which leads to well-known protocols, including Bluetooth, being vulnerable to man-in-the-middle (MITM) attacks). Furthermore, these device pairing protocols are often not usable [50] which can lead to further security problems and user frustration. We go into more detail about existing device pairing work in the background section.

In this paper we examine device pairing in the context of AR headsets, and develop a novel solution to pair two headsets that is quick and intuitive while providing a high degree of security. Our solution leverages the unique hardware capabilities required for AR and a unique combination of contextual information about AR headset use, facial recognition, and very recent advances in wireless localization.

The core idea of the scheme relates to how humans authenticate other human speakers. When trying to determine whom is speaking, our brains localize the source of the audio signal heard and then we match up that origin with a face that our brain has also recognized [11, 36]. Humans take a wave-signal source, localize it, then pair it with a face. Our work uses this idea, except instead of localizing an audio signal our system localizes a wireless signal that is adjacent to a face and recognizes the face automatically using facial recognition. By doing this, we create an authentication scheme robust against man-in-the-middle (MITM) attacks that reduce the problem of pairing for two users to simply looking at each other's faces and indicating to their AR headset that it "Looks Good To Me", thus establishing a secure connection. This system is aptly called "Looks Good To Me" or LGTM for short. In this paper we contribute:

- The LGTM protocol which brings innate human trust mechanisms into the digital world, allowing users to share any type of content with one-another face-to-face just by selecting a person to share with

- Analysis where we demonstrate that LGTM is secure against MITM attacks and highly usable

- We publish a full open-source implementation of LGTM under the MIT software license as a prototype and a building point for further improvements

In section 2 we provide context by surveying relevant areas including AR, device pairing, and wireless localization. In section 3 we present the system model, threat model, security objectives, and the LGTM protocol. In section 4 we present analyses of LGTM covering security, usability, and privacy implications. In Section 5 we present the details of our LGTM implementation, including a link to the complete open-source code-base. In section 6 we present experimental results involving localization accuracy and performance. In section 7 we discuss the potential for future work surrounding LGTM. In section 8 we conclude.

2. BACKGROUND

2.1 Augmented Reality

AR has been around as a research topic since as early as 1993 [54], but only recently have AR headsets been pursued as consumer and business products [1, 2, 4, 5]. AR at its core is taking what we see in reality and overlaying additional digital objects on top of it in the form of two-dimensional or three-dimensional renderings. Additionally, users can directly interact with these digital objects using their hands or other parts of their body.

The vision for future AR experiences lies in the HMD, a large pair of glasses with a mechanism to deliver light encoding specific digital objects directly into the user's field of vision. There is a wave of AR HMDs coming to market including the Meta 2 [4], the Microsoft HoloLens [5], and Magic Leap [2], all of which sport immersive and interactive experiences.

AR headsets are trending towards being stand-alone computers. The Microsoft HoloLens is already a fully functioning computer, and the company Meta has stated that it intends to move toward a full stand-alone computer headset [4]. This trend implies that AR users will be able to enjoy all the current advantages of computers with the addition of the powerful sensors and paradigm changing user interfaces that come with AR. This allows us to look at traditional problems that arise in the context of computers and computing and reexamine these existing problems and their existing solutions in the context of AR computers with an eye to improve upon the current state-of-the-art by using the additional functionality provided via the hardware and abilities from AR.

One of these problems is device pairing, which is a common procedure that must be done to connect two devices together such as a headset and a smart phone, a speaker and a smart phone, two smart phones, or any other wireless peripheral with a smart phone or computer.

2.2 Device Pairing

Device pairing is the area dedicated to authenticating devices without prior security context, and there have been many schemes introduced to address this problem on devices with myriad hardware features and constraints [14, 18, 34, 35, 41, 44–46]. Virtually all of these methods rely on communication over one or more out-of-band (OOB) channels. An OOB channel is any channel which is not the primary communications channel being used. This secondary channel can be defined quite broadly and many schemes use human sensory capabilities or human activity as the OOB channel [27, 28, 38]. The primary channel that the secondary channel is used to authenticate is a human imperceptible channel, most often the wireless channel. Most device pairing schemes proceed as follows. Some information is transmitted over the OOB channel, which is then used for authentication in the primary channel via some specialized authentication procedure. These authentication procedures can vary widely depending on what sort of OOB channel is used and what sort of information is transmitted across it. The simplest and most common example is for a numerical pin to be exchanged or verified between two devices via one or more human actors. Depending on the scheme, the pin may need to be entered on both devices, transmitted between two devices and then verified as matching by one or more users, or some combination of the two [3,19,27]. When human actors are involved, the exchanged information is often short to improve usability [3, 19], and thus possesses a low level of entropy, reducing its security.

Usability in device pairing schemes has accumulated its own body of research [14, 21, 24, 26, 28, 50] through protocol comparisons, usability studies, and analyses of security issues that usability can affect. Usability is important because it can affect protocol adoption and even the security of the protocol. If users are prompted to confirm a numeric string that appears on a pair of devices, and they confirm it without thorough checking, it is easy for incorrect devices to be authenticated, which is a huge security leak [50].

2.3 Wireless Localization

Wireless localization seeks to determine the exact point of origin of a wireless signal using only the signal itself. Until very recently wireless localization systems were infeasible to implement on commodity hardware due to high per-node antenna-count requirements [22] and the high number of nodes required, since many locations where localization would be beneficial simply do not have the required number of access points to provide accurate localization [16, 22, 23]. Yet another barrier to implementing localization on commodity devices is the lack of availability of granular information on the wireless channel on commodity hardware. Many commodity hardware devices have only provided received signal strength information (RSSI) as a measure of wireless signals, but recently there has been a trend towards providing more granular channel state information (CSI) on commodity devices [20, 55].

These past limitations and recent developments have lead to a recent influx of wireless localization work with techniques that both improve precision and decrease hardware requirements [16, 33, 48, 51]. Much of this work has only occurred in the past year, opening up new applications of wireless localization that did not previously exist. The most recent of these works reduces the hardware required for precise localization to a single access point with three antennas [51], which is unparalleled given that previous single access point localization was simply not feasible with high precision. We leverage these very recent advances in localization in LGTM's development, and expect LGTM to improve as wireless localization continues to become more robust and more precise.

3. LGTM PROTOCOL

3.1 System Model and Assumptions

Consider this scenario: Alice and Bob meet for the first time and want to pair their AR headsets to exchange some three-dimensional content. Alice and Bob both trigger the pairing protocol on their devices. Moments later they are both prompted to confirm one-another's faces, which are outlined on their respective AR displays. Alice and Bob confirm one-another's faces and now they are free to communicate over an encrypted channel. What could be simpler than that? This is what LGTM promises.

LGTM is specifically designed for authenticating two users, Alice and Bob. Alice and Bob are assumed to be users equipped with AR HMDs which are also stand-alone computers equipped with: wireless communications hardware with support for a point to point communications protocol, software and hardware support for wireless localization using the same wireless hardware used for communication, a high-definition video camera, and a translucent display directly in front of the users' eyes that is capable of displaying digital objects on top of the physical world.

An important piece of information that each headset is assumed to possess is a facial recognition model capable of recognizing the user of the headset. Alice's headset has a facial recognition model trained to recognize her, and likewise for Bob. Training these models can be done in a mirror.

The two hardware devices are assumed to have no prior security context and we assume that Internet access is not reliably accessible or not desirable.

As for the users themselves, Alice and Bob are assumed to be in sight of each other and would like to share some sort of digital content with one another, such as shared holograms or basic messaging. No assumptions are made of Alice and Bob's relationship with each other: they may be friends or strangers.

3.2 Threat Model and Protocol Objectives

Attackers have quite a bit of power when dealing with a wireless channel since the communications medium is both public and localized. Attackers have the capability to: eavesdrop on all packets transmitted, and replay packets collected from eavesdropping in any order. Further, the attacker may have multiple transceivers at her disposal, so attacks can be coordinated between multiple nodes. However, it is necessary for an attacker to be physically present, either personally or via a device controlled remotely, in order to modify the wireless channel.

An attacker with powerful equipment may attempt to impersonate Alice to trick Bob into sending sensitive content. She may also impersonate Bob to send Alice malicious data, such as malware embedded in a PDF file. Furthermore, an attacker may launch a MITM attack to eavesdrop on Alice and Bob's communications.

We do not include denial-of-service (DOS) attacks in our threat model since these attacks are present across all wireless and networked devices and cannot be completely defended against. However we do briefly discuss how DOS attacks can be executed against LGTM in section 4.

Lastly, the attacker may want to prevent communication between Alice and Bob by either jamming the wireless communication or draining the battery of the AR devices. We do not consider this problem in our threat model as denial-of-service attacks are pervasive across all networked devices, but we do include a brief discussion after the protocol description regarding potential mitigation strategies.

The primary security objective is for Alice and Bob to correctly authenticate one another's wireless signals so that they can engage in secure point to point wireless communication. Since Alice and Bob do not share any security context, this authentication must be bootstrapped.

A secondary objective is for Alice and Bob to correctly select which user they want to communicate with. This objective is directly related to the first but it warrants distinction as a worthy problem on its own merit since there are many practical attacks that rely upon tricking users into selecting the wrong thing [12,42]. There has even been work done on these attacks that specifically look at AR systems [40]. We refer to this pervasive issue as the user-selection problem.

3.3 Protocol

To begin the protocol, Alice presses a button which may be digitally rendered over reality or physically located on the AR headset. At this point Alice's headset begins listening on the local wireless channel and broadcasts an initialization message. Since Bob is not listening yet, Alice's message goes unanswered. Bob presses a button to initiate the protocol, opening his device up to listen to the wireless channel and generating an initialization packet containing Diffie-Hellman parameters g and p, Bob's private key b, Bob's public key computed as: $B = g^b \, mod \, p$, and a randomly generated number, R_{B1}, to serve as half of a session identifier. Bob broadcasts his message $g||p||B||R_{B1}$, where $||$ denotes concatenation. Alice receives Bob's initialization message, generates her private key a and computes her public key in relation to the received Diffie-Hellman parameters g and p as: $A = g^a \, modp$. Alice generates a random number R_{A1}, to serve as the second half of a session identifier and broadcasts her public key A concatenated with the random number R_{A1}, and $R_{B1}||R_{A1}$, which is the session identifier for Alice and Bob's current session. This makes Alice's full message: $A||R_{A1}||R_{B1}$. Bob receives Alice's message and verifies that it is in response to his initial message by checking $R_{B1}||R_{A1}$. Alice and Bob both compute the shared key $K = B^a \, modp = A^b \, modp$. At this point Alice and Bob have established a shared symmetric key, but have not established the authenticity of each other's identities.

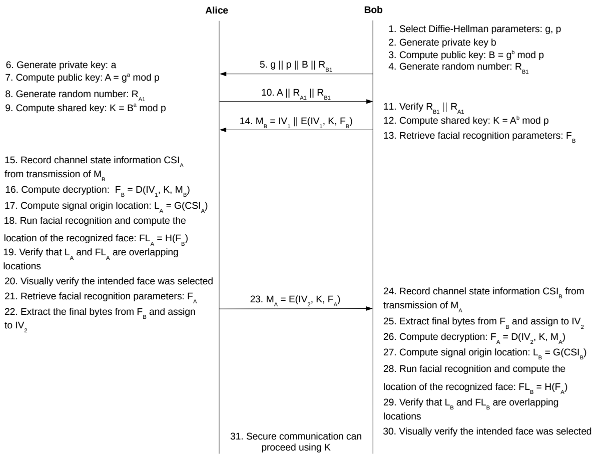
Alice
Bob
1. Select Diffie-Hellman parameters: g, p
2. Generate private key b
3. Compute public key: $B = g^b$ mod p
4. Generate random number: R_{B1}
5. $g || p || B || R_{B1}$
6. Generate private key: a
7. Compute public key: $A = g^a$ mod p
8. Generate random number: R_{A1}
9. Compute shared key: $K = B^a$ mod p
10. $A || R_{A1} || R_{B1}$
11. Verify $R_{B1} || R_{A1}$
12. Compute shared key: $K = A^b$ mod p
13. Retrieve facial recognition parameters: F_B
14. $M_B = IV_1 || E(IV_1, K, F_B)$
15. Record channel state information CSI_A from transmission of M_B
16. Compute decryption: $F_B = D(IV_1, K, M_B)$
17. Compute signal origin location: $L_A = G(CSI_A)$
18. Run facial recognition and compute the location of the recognized face: $FL_A = H(F_B)$
19. Verify that L_A and FL_A are overlapping locations
20. Visually verify the intended face was selected
21. Retrieve facial recognition parameters: F_A
22. Extract the final bytes from F_B and assign to IV_2
23. $M_A = E(IV_2, K, F_A)$
24. Record channel state information CSI_B from transmission of M_A
25. Extract final bytes from F_B and assign to IV_2
26. Compute decryption: $F_A = D(IV_2, K, M_A)$
27. Compute signal origin location: $L_B = G(CSI_B)$
28. Run facial recognition and compute the location of the recognized face: $FL_B = H(F_A)$
29. Verify that L_B and FL_B are overlapping locations
30. Visually verify the intended face was selected
31. Secure communication can proceed using K

Figure 1: Protocol diagram of Alice and Bob's communications and computations during LGTM. Note that $H(\cdot)$ denotes a facial recognition algorithm, $G(\cdot)$ denotes a wireless localization algorithm, and $||$ denotes concatenation.

Bob retrieves facial recognition parameters that define a facial recognition model to recognize his face, F_B. Bob encrypts the facial recognition parameters using the shared key and an initialization vector, IV, which Bob concatenates with his encrypted facial recognition parameters, giving us $IV||E(IV, K, F_B)$, and broadcasts this message. Alice receives Bob's encrypted message with his facial recognition parameters and also records the channel state information, CSI_A, associated with the packets carrying Bob's message. Alice decrypts the message and retrieves F_B. Alice computes Bob's location using a wireless localization algorithm denoted by $G(\cdot)$ to get Bob's location: $L_A = G(CSI_A)$. Alice runs a facial recognition algorithm denoted by $H(\cdot)$, using Bob's facial recognition parameters to obtain a location for any faces that match F_B, denoted by $FL_A = H(F_B)$. Alice compares L_A and FL_A to see if the coordinates overlap. If they do not match then they are thrown out. If there are no matches among all the options for L_A and FL_A then the protocol is aborted. If there are matches, then Alice's device renders a box around Bob's face, matched by F_B, and whose location, FL_A, overlaps with the corresponding L_A. Alice is prompted to verify that the protocol has selected the correct face. In practice, there may be many pairs of faces and locations to check at once since multiple users may be carrying out LGTM at the same time. In this case, Alice selects Bob's face from the available faces, if it is present. If Alice fails to select Bob's, face then the protocol aborts at this point.

Otherwise, Alice retrieves facial recognition parameters that define a facial recognition model to recognize her face, F_A. Alice uses the shared key, K and the last several bytes of F_B as an initialization vector, IV, to compute the encryption, $E(IV, K, F_A)$ and broadcasts it. Bob receives Alice's encrypted message with her facial recognition parameters and also records the channel state information, CSI_B, associated with the packets carrying Alice's message. Bob decrypts the message using the shared key, K, and the initialization vector derived from the last several bytes of F_B and retrieves F_A.

Bob computes Alice's location using the wireless localization algorithm to get: $L_B = G(CSI_B)$. Bob runs the facial recognition algorithm, using Alice's facial recognition parameters to obtain a location for any faces that match F_A, denoted by $FL_B = H(F_A)$. Bob compares L_B and FL_B to see if the coordinates overlap. If they do not match, then they are thrown out. If there are no matches among all the options for L_B and FL_B, then the protocol is aborted. If there are matches, then Bob has a box drawn around Alice's face, matched by F_A, and whose location, FL_B, overlaps with the corresponding L_B. Bob is prompted to verify that the protocol has selected the correct face. In practice, there may be many pairs of faces and locations to check at once since multiple users may be carrying out LGTM at the same time. In this case, Bob selects Alice's face from the available faces, if it is present. If Bob fails to select Alice's face then the protocol aborts. Otherwise, the protocol has successfully completed, Alice and Bob have established a secure key K, and they have authenticated each other's wireless signals ensuring that they are communicating with who they think they are. They are now free to send encrypted content back and forth.

4. PROTOCOL ANALYSIS

4.1 Protocol Components

Key Exchange and Channel Security.

LGTM uses Diffie-Hellman key exchange [13] for establishing a shared key between Alice and Bob. High-speed key generation makes it feasible to generate a new key pair every time pairing is performed. This provides LGTM with perfect forward secrecy, meaning that if an attacker obtains one of the private keys used in a single pairing the other public-private key pairs along with the symmetric keys derived from them remain secure.

LGTM uses symmetric key encryption to protect the confidentiality of the facial recognition parameters. These parameters are not necessarily private, but they are not easily obtainable by arbitrary individuals, so encrypting them provides protection from attackers that do not have easy access to them. Furthermore, encryption serves as a form of authentication so that Alice and Bob can properly keep track of the potentially many message streams from LGTM being performed between multiple parties in the same area at once. Initialization vectors are used with symmetric encryption to protect against known plain text attacks.

Wireless Localization.

LGTM's security rests atop wireless localization's accuracy. Localization connects a wireless signal to a physical location by computing the signal's point of origin. This location can be matched with a device or person occupying that location, and with human assistance, this device or person can be authenticated. Wireless localization remains a difficult and unsolved problem and many current schemes are too inaccurate to be suitable for security applications. However, very recent and promising work achieves a median error of 65 cm in line of sight conditions with a single laptop equipped with three antennas [51]. Consider now that LGTM's localization scenario is a constrained one. If two users are performing LGTM, they must have a line of sight between each other and in the most common use-case, they will be quite close to one another, likely within four meters. If we look at the results in [51] again with these constraints in mind, we see that this localization technique achieves accurate localization down to less than 15 cm median error, which is sufficient for security applications.

Facial Recognition.

This brings us to facial recognition. We state above that wireless localization is what allows LGTM to authenticate a device or person, so of what use is facial recognition? Having users authenticate a physical location computed using wireless localization, identified by say, a sphere, requires users to search for the sphere and verify that it intersects with the person they're attempting to communicate with. This will work, but it is not a very usable way for a user to choose whom to communicate with (I.E. to solve the user-selection problem). The facial recognition parameters exchanged in LGTM serve to make it simple for a user to select whom they are to communicate with.

Humans are phenomenal at facial recognition. It was found in [49] that infants as young as 1-3 days old are able to recognize faces. By using facial recognition to effectively preprocess what the user is currently seeing, locations obtained via localization that do not match up with the location of the face identified by the facial recognition parameters are automatically eliminated. This reduces the chances of a user selecting an invalid location either accidentally or through an attacker's careful misdirection. Once the invalid selections are eliminated, there may still be multiple valid face-location pairs to choose from, but selecting a face remains a far simpler task than picking out a sphere or another sort of location representation.

Human Verification.

The final step in the protocol is for the actual human users to select a face from the options LGTM provides. A user of any sharing system must tell that system whom they wish to communicate with, so this step is a requirement regardless of whether LGTM is in use or not. This step in the LGTM protocol serves to select a user to share with as well as authenticating that user, in a process seamless to the user. From the user's perspective selecting who to share with is the same as authenticating them, which is possible thanks to the preprocessing done using wireless localization and facial recognition.

On top of this, humans are great at recognizing the difference between a mask and a true human face. Without the human verification step, LGTM could be fooled by an attacker wearing a mask. Human verification serves as a final check against spoofing attempts, but doesn't add any extra steps for the user to jump through.

4.2 Security Against Attacks

We have spoken at length about the security properties the techniques used in LGTM possess; now we discuss how LGTM performs against attacks.

Man-in-the-Middle Attacks.

MITM attacks plague virtually every pairing protocol [26]. The very nature of pairing opens these protocols up to MITM attacks since there is no prior authentication information. LGTM, however, has protection against MITM attacks not afforded to other pairing protocols thanks to wireless localization. Localization pairs a signal to a location so attackers must occupy the same physical location as both users to successfully launch a MITM attack. This is still theoretically possible using two extremely small coordinated devices physically located on Alice and Bob impersonating each of them respectively, but this is very difficult since Alice and Bob are likely to notice a wireless device placed on their person.

A more realistic MITM attack for the present is one reliant on inaccurate localization results. Localization can not be called a solved problem yet, and any implementation of LGTM will have to deal with the currently accepted localization techniques, errors and all. The attack would still require two coordinated devices, but they could be placed adjacent to Alice and Bob instead of on their person. One device would impersonate Alice, the other would impersonate Bob, and the two devices would communicate the facial recognition parameters back and forth to successfully impersonate Alice and Bob. However as wireless localization becomes more and more precise, as it has been over the field's entire history, these attacks will become harder and harder.

User Impersonation Attacks.

Inaccurate localization results can also be used in one-way impersonation. Eve might want to impersonate Alice to Bob, so she sets up a device near enough to Alice to fool the localization procedure and transmits Alice's facial recognition parameters.

User impersonation will usually be protected against even further than localization goes by context. LGTM's most likely use-case is that Alice and Bob want to share digital content via their AR headsets and that they are already interacting and conversing. If Eve impersonates Alice to Bob, then Alice will not be sharing anything with Bob and vice-versa, which will not go unnoticed. Bob will likely terminate the connection and try again since it will be apparent that the protocol failed. It is possible though to imagine scenarios where Bob might not notice that he is paired with someone besides Alice and in these cases the only defense is to improve localization accuracy.

Denial-of-Service Attacks.

DOS attacks are a common attack across all devices with network connectivity. Here we discuss two different ways that DOS attacks can be launched against LGTM.

The first is common to all wireless devices, jamming the wireless channel so that messages are not successfully received at all. Defending against jamming is still an active area of research. In LGTM, we do not have any built-in defenses against jamming as jamming requires expensive equipment and is illegal in many countries and states.

The second avenue is through the protocol itself. LGTM's localization step can be fairly expensive, and the rest of the protocol doesn't just happen for free. If Eve spams Alice or Bob's device with LGTM requests, they will waste time processing these requests. But this attack goes further. Since these AR headsets are stand-alone and mobile, they will be battery-powered. By spamming Alice or Bob's device and forcing computation, Eve can execute a battery drain attack. The best defense against an attack like this have monitoring LGTM protocol messages, and if too many are received, the protocol is aborted, similar to TCP's congestion control mechanism [32]. This control mechanism defends against battery drain attacks, but not general denial of service.

4.3 Usability

Usability is a great thing to have for any piece of technology. It increases the likelihood of adoption, decreases avoidable user error, and reduces user frustration, which is linked to the other two points.

Besides being good for user satisfaction, usability can be an important factor in increasing security by reducing user errors that lead to security issues. It is no surprise that usability is increasingly becoming a focus for security schemes, especially in pairing [14, 21, 24, 26, 28, 50]. One study of device pairing methods [50] found that increasing the quality and usability of a user interface in a security scheme decreased errors that lead to a security failure in one pairing method by 20 percentage points and another by 7.5 percentage points. These two pairing methods were not seemingly complex schemes either. The first one required comparing two short alphanumeric strings on each device and confirming that they matched. The second one required selecting matching alphanumeric strings from a list on each device.

There are established pairing protocols using short-range peripherals in most consumer devices today. However, there are still frequent user errors in the currently deployed technologies. Consider Bluetooth, there are three pairing methods in the standard which protect against MITM attacks: numeric comparison, passkey entry, and OOB communication (most commonly provided via tap-to-pair, where users bump two devices together to authenticate them) [3]. The device pairing usability study in [50] showed that users commit security errors up to 20 percent of the time when verifying numeric strings; the consequence of a security error is that the user authenticates the wrong user. Passkey entry was even worse, resulting in security errors 42.5 percent of the time [50]. These statistics indicate serious failures in usability present in the Bluetooth standard which can lead to security breaches far too often.

Tap-to-pair is a more usable alternative for smart phones, but it would make for a very comical display when used with headsets. Either users would have to each remove their headsets and tap them together every time they authenticate someone new, a process certain to cause disdain from users or, even more comically, users would need to tap their heads together to authenticate one another.

To improve user satisfaction and increase security, LGTM must do better in designing a usable authentication system. LGTM uses the combination of facial recognition and wireless localization to reduce user-device authentication to two actions. By adding facial recognition on top of wireless localization LGTM can auto-remove choices whose wireless signal location and face location do not match up. Users outside the device user's field of view are also auto-eliminated by context since the user is not facing that direction. Reducing the number of choices available to a user is a fantastic way to increase usability, since it inherently reduces cognitive load for the user. But the core usability gain from LGTM comes from combining the process of selecting which user to share content with with the process of authentication. This makes authentication seamless for the user and makes it less of an abstract concept.

4.4 Privacy

A common misconception is that most people are not concerned about digital privacy. However, a 2015 study by the Pew Research Center [30] found that 86% of Internet users have taken steps to remove or mask their digital footprints online, and 55% of Internet users have taken steps to avoid surveillance by specific individuals, organizations, or governments. When it comes to user privacy, LGTM delivers. Since LGTM relies on point to point wireless communication, attackers must be co-located with their targets, significantly diminishing the reach that arbitrary attackers can exert. Prolonged digital surveillance of an individual's point to point content sharing would require something akin to stalking, which is unlikely to occur as often as hacking does on the Internet.

Furthermore, by sending data from point to point, we can avoid empowering a central authority with hordes of user data: who users talk to, who users are close to, what users are sending to one another, and when. This is data that many users would like to keep private, but that many large companies see as a gold mine. The safest way to keep this data from being abused to is not make it available to third-parties. With LGTM, this data is as private as any untapped conversation between two people.

5. IMPLEMENTATION

5.1 Testbed

Real world AR headsets are still fresh on the market and are very difficult to obtain. On top of this, the limited headsets on the market do not provide sufficient access to information about the wireless channel that is necessary for wireless localization. Depending on the technique, modern localization techniques may require access to physical layer channel state information [33], channel switching and synchronization [51], or some other type of wireless information. These two factors make implementing LGTM on real AR headsets infeasible.

However, we have stripped down LGTM's hardware and contextual requirements to the bare minimum leaving us in need of: a high-definition video camera, wireless point to point communication capabilities, wireless localization capabilities, a display, reasonable computational abilities, and close proximity of a face to wireless antennas.

All of these requirements are satisfied by equipping two Fujitsu LifeBook T900 laptops running Linux Mint 17 Qiana with a Logitech C270 720p web cam, an Intel 5300 wireless card with custom firmware for Linux from Halperin et al. [20] that enables retrieval of channel state information and point to point wireless communication using 802.11n, an array of three antennas affixed to the back of the laptop screen at 10 centimeter intervals, and finally by attaching printed pictures of faces from the Yalefaces dataset [8] to the back of the laptop over the antenna array. This last point is important since AR headset users will have their face directly adjacent to the wireless transmitter which LGTM exploits to couple the wireless signal and the user together. This context is an extremely important enabler of LGTM. A pair of images with the face photograph on and off is shown in Figure 2. To be sure, the testbed itself is not a practical setup, but it perfectly emulates the requisite hardware of real AR headsets.

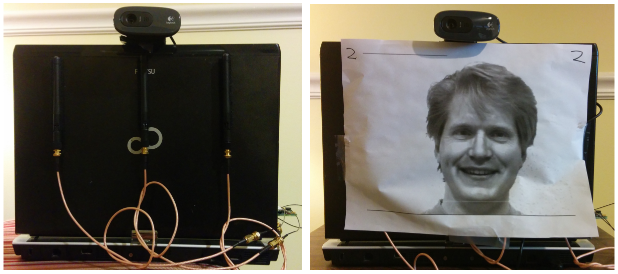

Figure 2: Testbed laptop with and without a printed face attached.

5.2 Technologies and Software

LGTM employs many technologies in service of bootstrapping secure communication and these technologies have several different implementation choices.

Wireless Capabilities.

We use custom firmware on the Intel 5300 wireless card and supplementary software tools to collect channel state information, send packets, receive packets, and process channel state information [20]. We use the 5 GHz channel of 802.11n due to known issues that arise with the Intel 5300 card when it is used on the 2.4 GHz band [16, 33].

Cryptography.

To perform the key exchange, we employ elliptic curve Diffie-Hellman key exchange instead of standard Diffie-Hellman key exchange to reduce the number of bits required for the public-private key pairs, thus increasing performance [9]. For symmetric encryption, we use the advanced encryption standard (AES) in Galois Counter Mode (GCM). GCM provides both authentication and encryption for messages using a keyed hash as a message authentication code or HMAC. To implement LGTM's cryptographic components we use the popular C++ library, Crypto++, which provides functions and classes to support elliptic-curve Diffie-Hellman, random number generation, AES, and GCM.

Facial Recognition.

Facial recognition is a multi-step process. Before recognizing a face, all the faces must be detected so they can be fed to the facial recognition algorithm. Our implementation's facial detection relies upon Haar feature-based cascade classifiers [53]. This method was designed specifically for performance. It uses many feature detectors which are run in a tree-like fashion, with simpler, faster feature detectors being run first, followed by more complex, more expensive, more discriminating feature detectors.

For facial recognition, we use local binary pattern histograms (LBPH) for facial recognition [7, 31]. We selected LBPH because it is known to be robust in the face of less than ideal situations like the kind we face in our testbed. Furthermore, the models are of reasonable size, in our experiments they totaled between 1000 and 1100 packets.

We used the OpenCV 3.0 [10] C++ library in our implementation to implement LBPH for facial recognition and also for facial detection using Haar feature-based cascade classifiers. OpenCV provides out-of-the-box implementations for all of these methods as well as support for additional image preprocessing and real-time video annotations.

Wireless Localization.

The most intricate of the technologies involved in LGTM is by far wireless localization. The recent advances in wireless localization [33, 48, 51] are indeed what make LGTM feasible. For our testbed we implemented a modified version of SpotFi, adapted to run from a single device instead of coordinated devices.

SpotFi is an angle of arrival based localization technique that relies on the classical MUSIC algorithm [43] to compute the angle that a wireless signal originates from relative to an array of antennas. The key to MUSIC is the existence of a phase difference between antennas as a signal arrives at each one in an array. These phase differences occur because the antennas are at different locations, meaning a single wireless signal must travel different distances to arrive at each antenna in an array.

However, MUSIC can be rendered inaccurate as a result of multipath effects, caused by a single signal reflecting off objects in the environment and thus arriving at different angles, making the direct path angle of arrival difficult to determine. Multipath can be resolved in MUSIC by having more antennas. Specifically, the number of antennas must be greater than the number of multipath components. However average indoor conditions usually have five paths resulting from multipath effects [33] and having five antennas on a consumer device is currently both impractical and expensive.

SpotFi presents techniques to resolve multipath without increasing the number of antennas. The key insights are using channel state information measurements from each subcarrier in 802.11n wireless combined with a method of creating more sensors using clever data restructuring. To use these two insights SpotFi modifies MUSIC to consider time of flight as well as angle of arrival. Leveraging these

Table 1: Modified SpotFi localization results. Percent correct for the top 1 through 5 positions are reported along with mean error and median error for distances of: 1 m, 2 m, and 3 m.

Distances:	1 m	2m	3m
Correct in Top 1:	60.0 %	30.0 %	23.0 %
Correct in Top 2:	83.0 %	47.0 %	38.0 %
Correct in Top 3:	95.0 %	65.0 %	45.0 %
Correct in Top 4:	100.0 %	75.0 %	50.0 %
Correct in Top 5:	100.0 %	75.0 %	50.0 %
Mean Error:	0.722 m	1.654 m	2.321 m
Median Error:	0.664 m	1.526 m	1.991 m

insights allows SpotFi to use MUSIC effectively with only three antennas. For in-depth technical details we refer the reader to the full SpotFi paper [33].

The SpotFi system runs the modified version of MUSIC [43] on multiple wireless access points and then combines the results from each to come up with a precise location. In our implementation however we are working from a single laptop, and so we do no such combination, relying upon the angle of arrival computed on our single device. This will certainly decrease our localization accuracy, but there is little other work dealing with wireless localization from a single device with a limited number of antennas.

We implement our altered SpotFi along with the altered MUSIC algorithm from scratch in MATLAB. To our knowledge there is no open-source implementation of SpotFi yet, despite apparent high demand [6], so our code release of SpotFi is a valuable contribution in and of itself.

Bringing It All Together.

The techniques described above are implemented in a combination of: C, C++, MATLAB, and Linux shell commands. To tie all this software together and construct a data pipeline for transmission, reception, processing, and user input we use a Bash script to coordinate the flow of control. For passing state variables and data between separate programs we use files for simplicity. The source code for the entire implementation of LGTM is available at: www.github.com/egaebel/lgtm.

6. EXPERIMENTS

We found it prudent to run a few experiments to validate our implementation. Our evaluation deals with the accuracy of our SpotFi implementation adapted for point to point use since this is the primary component responsible for the security of this LGTM prototype, and our modifications are completely untested.

6.1 Accuracy

SpotFi uses a likelihood technique to make the final selection of angle of arrival. Prior to this final selection SpotFi has several candidate angle of arrivals to select from. This can lead to an incorrect angle of arrival being selected over the correct one by a slim margin. To evaluate the modified SpotFi's localization accuracy, we focus on the accuracy of the top-5 angle of arrival selections as well as the mean error and median error. To account for acceptable levels of error in angle of arrival computations we introduce a slight error tolerance of 40 cm on either side of the face, so our top-5 error rates are computed after having taken this tolerance into account, but mean error and median error do not consider this tolerance. The data used was gathered with the two testbed laptops at angles of: -20, -10, 0, 10, and 20 degrees, where 0 degrees is when the laptops are directly facing one another, positive degrees are to the right, and negative degrees are to the left. For each angle, we performed LGTM 10 times and saved off the channel state information used on each laptop, yielding 20 samples for each angle at each distance for a total of 100 samples for each distance. We repeated this at distances of: 1 m, 2 m, and 3 m. Top-5, mean error, and median error accuracy reports can be found in Table 1.

Localization accuracy is strongly coupled with distance between devices. Percent correct in the top-1 position is halved when moving from 1 meter to 2 meters and beyond 2 meters the percent correct in the top-5 positions drops below 50%. This shows that we must rely on the localization community to further increase the precision of localization from a single access point. From the data we present modified SpotFi is only feasible for security purposes within 1 m, which still has its uses. However, it is a step towards a more secure system. Further improvements to wireless localization will continue to improve the security of LGTM, and indeed this has already happened since this work began thanks to the work in [51].

7. CONCLUSION

The future of computing lies in AR and beyond. The digital interfaces that we all interact with on a daily basis are going to become richer, more intuitive, and more natural. It's only right that authentication grow more intuitive and natural as well. In this paper we have presented LGTM: authentication for AR, which promises to make authenticating a wireless signal as simple as figuring out who's talking to you. LGTM achieves this apparent simplicity by leveraging advances in wireless localization and facial recognition, combining wireless signal origin with a user's face location to create this simplicity.

We have implemented LGTM using a diverse set of software, hardware, and technologies. This implementation has been open-sourced, so that it can be built upon, improved upon, and scrutinized by other interested parties. After performing extensive empirical and theoretical analysis on our implementation, we have found it to be a promising security scheme that will only improve as time goes on and we get closer to the future of computing.

Acknowledgments

This work was supported in part by the National Science Foundation under Grants CNS-1443889, CNS-1405747, CNS-1446478, and CNS-1343222.

8. REFERENCES

[1] Google glass website.
[2] Magic leap website.
[3] Bluetooth specification version 4.0, 2010.
[4] Augmented reality meta website, 2016.
[5] Microsoft hololens website, 2016.
[6] relative phase is not right? On Issues Page, April 2016.

[7] T. Ahonen, A. Hadid, and M. Pietikäinen. *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I*, chapter Face Recognition with Local Binary Patterns, pages 469–481. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[8] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, July 1997.

[9] D. J. Bernstein. *Public Key Cryptography - PKC 2006*, chapter Curve25519: New Diffie-Hellman Speed Records, pages 207–228. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[10] G. Bradski. Opencv. *Dr. Dobb's Journal of Software Tools*, 2000.

[11] S. Campanella and P. Belin. Integrating face and voice in person perception. *Trends in Cognitive Sciences*, 11(12):543, April 2016.

[12] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 581–590, New York, NY, USA, 2006. ACM.

[13] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654, Sept. 2006.

[14] A. Gallego, N. Saxena, and J. Voris. Playful security: A computer game for secure wireless device pairing. In *Computer Games (CGAMES), 2011 16th International Conference on*, pages 177–184, July 2011.

[15] C. Gehrmann and K. Nyberg. Manual authentication for wireless devices. *RSA Cryptobytes*, 7:2004, 2004.

[16] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson. Phaser: Enabling phased array signal processing on commodity wifi access points. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 153–164, New York, NY, USA, 2014. ACM.

[17] S. Gollakota, N. Ahmed, N. Zeldovich, and D. Katabi. Secure in-band wireless pairing. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, pages 16–16, Berkeley, CA, USA, 2011. USENIX Association.

[18] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and clear: Human-verifiable authentication based on audio. In *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, pages 10–10, 2006.

[19] U. E. Group. Bluetooth user interface flow diagrams for bluetooth secure simple pairing devices. Technical Report v1.0, Bluetooth, September 2007.

[20] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Tool release: Gathering 802.11n traces with channel state information. *SIGCOMM Comput. Commun. Rev.*, 41(1):53–53, Jan. 2011.

[21] I. Ion, M. Langheinrich, P. Kumaraguru, and S. Čapkun. Influence of user perception, security needs, and social factors on device pairing method choices. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, SOUPS '10, pages 6:1–6:13, New York, NY, USA, 2010. ACM.

[22] K. J. Jie Xiong. Arraytrack: a fine-grained indoor location system. In *10th USENIX conference on Networked Systems Design and Implementation*, 2013.

[23] K. Joshi, S. Hong, and S. Katti. Pinpoint: Localizing interfering radios. In *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, pages 241–253, Lombard, IL, 2013. USENIX.

[24] R. Kainda, I. 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, pages 11:1–11:12, New York, NY, USA, 2009. ACM.

[25] A. Klemm, C. Lindemann, and O. P. Waldhorst. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 4, pages 2758–2763 Vol.4, Oct 2003.

[26] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun. A comparative study of secure device pairing methods. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1–10, March 2009.

[27] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun. A comparative study of secure device pairing methods. *Pervasive Mob. Comput.*, 5(6):734–749, Dec. 2009.

[28] A. Kumar, N. Saxena, and E. Uzun. Alice meets bob: A comparative usability study of wireless device pairing methods for a "two-user" setting. *CoRR*, abs/0907.4743, 2009.

[29] K. C. Lee, S. h. Lee, R. Cheung, U. Lee, and M. Gerla. First experience with cartorrent in a real vehicular ad hoc network testbed. In *2007 Mobile Networking for Vehicular Environments*, pages 109–114, May 2007.

[30] R. K. LEE RAINIE, SARA KIESLER and M. MADDEN. Anonymity, privacy, and security online. Web article, September 2015.

[31] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li. Learning multi-scale block local binary patterns for face recognition. In *Proceedings of the 2007 International Conference on Advances in Biometrics*, ICB'07, pages 828–837, Berlin, Heidelberg, 2007. Springer-Verlag.

[32] I. M. Allman, V. Paxson and E. Blanton. Tcp congestion control.

[33] D. B. Manikanta Kotaru, Kiran Joshi and S. Katti. Spotfi: Decimeter level localization using wifi. *SIGCOMM Comput. Commun. Rev.*, 45(5):269–282, Aug. 2015.

[34] R. Mayrhofer and H. Gellersen. Shake well before use: Intuitive and secure pairing of mobile devices. *IEEE Transactions on Mobile Computing*, 8(6):792–806, June 2009.

[35] J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: using camera phones for human-verifiable authentication. In *2005 IEEE Symposium on Security and Privacy (S P'05)*, pages 110–124, May 2005.

[36] J. C. Middlebrooks and D. M. Green. Sound localization by human listeners. *Annual Review of Psychology*, 42(1):135–159, 1991. PMID: 2018391.

[37] W. H. D. F. moderator. Hologram sharing, April 2016.

[38] R. C.-W. Phan and P. Mingard. Analyzing the secure simple pairing in bluetooth v4.0. *Wireless Personal Communications*, 64(4):719–737, 2010.

[39] B. Qureshi, G. Min, D. Kouvatsos, and M. Ilyas. An adaptive content sharing protocol for p2p mobile social networks. In *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*, pages 413–418, April 2010.

[40] F. Roesner, T. Kohno, and D. Molnar. Security and privacy for augmented reality systems. *Commun. ACM*, 57(4):88–96, Apr. 2014.

[41] N. Saxena, J. E. Ekberg, K. Kostiainen, and N. Asokan. Secure device pairing based on a visual channel. In *2006 IEEE Symposium on Security and Privacy (S P'06)*, pages 6 pp.–313, May 2006.

[42] N. Saxena and M. B. Uddin. Secure pairing of "interface-constrained" devices resistant against rushing user behavior. In *Proceedings of the 7th International Conference on Applied Cryptography and Network Security*, ACNS '09, pages 34–52, Berlin, Heidelberg, 2009. Springer-Verlag.

[43] R. Schmidt. Multiple emitter location and signal parameter estimation. *Antennas and Propagation, IEEE Transactions on*, 34(3):276–280, Mar 1986.

[44] D. B. Smetters, D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *None*, 2002.

[45] C. Soriente, G. Tsudik, and E. Uzun. Beda: Button-enabled device association, 2007.

[46] C. Soriente, G. Tsudik, and E. Uzun. *Information Security: 11th International Conference, ISC 2008, Taipei, Taiwan, September 15-18, 2008. Proceedings*, chapter HAPADEP: Human-Assisted Pure Audio Device Pairing, pages 385–400. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[47] F. Stajano and R. J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proceedings of the 7th International Workshop on Security Protocols*, pages 172–194, London, UK, UK, 2000. Springer-Verlag.

[48] D. K. Swarun Kumar, Stephanie Gil and D. Rus. Accurate indoor localization with zero start-up cost. In *Proceedings of the 20th annual international conference on Mobile computing and networking (Mobicom)*, 2014.

[49] S. F. Turati C, Macchi Cassia V and L. I. Newborns' face recognition: role of inner and outer facial features. *Child Development*, 77:297–311, March 2006.

[50] E. Uzun, K. Karvonen, and N. Asokan. *Financial Cryptography and Data Security*, chapter Usability Analysis of Secure Pairing Methods, pages 307–324. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[51] D. Vasisht, S. Kumar, and D. Katabi. Decimeter-level localization with a single wifi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 165–178, Santa Clara, CA, Mar. 2016. USENIX Association.

[52] S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *Proceedings of the 25th Annual International Conference on Advances in Cryptology*, CRYPTO'05, pages 309–326, Berlin, Heidelberg, 2005. Springer-Verlag.

[53] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511–I–518 vol.1, 2001.

[54] P. Wellner, W. Mackay, and R. Gold. Back to the real world. *Commun. ACM*, 36(7):24–26, July 1993.

[55] Z. Yang, Z. Zhou, and Y. Liu. From rssi to csi: Indoor localization via channel response. *ACM Comput. Surv.*, 46(2):25:1–25:32, Dec. 2013.