

---

# **A Privacy-aware and Secure System for Human Memory Augmentation**

Doctoral Dissertation submitted to the  
Faculty of Informatics of the Università della Svizzera italiana  
in partial fulfillment of the requirements for the degree of  
Doctor of Philosophy

presented by  
**Agon Bexheti**

under the supervision of  
Prof. Marc Langheinrich

September 2019



---

## Dissertation Committee

**Prof. Antonio Carzaniga** Università della Svizzera italiana, Switzerland  
**Prof. Fernando Pedone** Università della Svizzera italiana, Switzerland  
**Prof. Cecilia Mascolo** University of Cambridge, United Kingdom  
**Prof. Claudio Bettini** Università degli Studi di Milano, Italy

Dissertation accepted on 06 September 2019

---

Research Advisor

**Prof. Marc Langheinrich**

---

PhD Program Director

**Prof. Walter Binder and Prof. Silvia Santini**

---

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

---

Agon Bexheti  
Lugano, 06 September 2019



# Abstract

The ubiquity of digital sensors embedded in today’s mobile and wearable devices (e.g., smartphones, wearable cameras, wristbands) has made technology more intertwined with our life. Among many other things, this allows us to seamlessly log our daily experiences in increasing numbers and quality, a process known as “lifelogging”. This practice produces a great amount of pictures and videos that can potentially improve human memory. Consider how a single photograph can bring back distant childhood memories, or how a song can help us reminisce about our last vacation.

Such a vision of a “memory augmentation system” can offer considerable benefits, but it also raises new security and privacy challenges. Maybe obviously, a system that captures everywhere we go, and everything we say, see, and do, is greatly increasing the danger to our privacy. Any data breach of such a memory repository, whether accidental or malicious, could negatively impact both our professional and private reputation. In addition, the threat of memory manipulation might be the most worrisome aspect of a memory augmentation system: if an attacker is able to remove, add, or change our captured information, the resulting data may implant memories in our heads that never took place, or, in turn, accelerate the loss of other memories.

Starting from such key challenges, this thesis investigates how to design *secure* memory augmentation systems. In the course of this research, we develop tools and prototypes that can be applied by researchers and system engineers to develop pervasive applications that help users capture and later recall episodic memories in a secure fashion. We build trusted sensors and protocols to securely capture and store experience data, and secure software for the secure and privacy-aware exchange of experience data with others. We explore the suitability of various access control models to put users in control of the plethora of data that the system captures on their behalf. We also explore the possibility of using in situ physical gestures to control different aspects regarding the capturing and sharing of experience data. Ultimately, this thesis contributes to the design and development of secure systems for memory augmentation.



# Acknowledgements

I acknowledge the financial support of the Future and Emerging Technologies (FET) programme within the 7th Framework Programme for Research of the European Commission, under FET Grant Number: 612933 (RECALL).



# Contents

Contents	vii
List of Figures	xi
List of Tables	xiii
<b>I Introduction and Background</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Vision: Pervasive Memory Augmentation . . . . .	5
1.1.1 Memory Cue Sources . . . . .	7
1.1.2 Cue Presentation . . . . .	9
1.1.3 Sample Scenarios . . . . .	9
1.2 Research Questions . . . . .	11
1.3 Research Context: The RECALL Project . . . . .	13
1.4 Thesis Goals . . . . .	13
1.5 Methodology . . . . .	14
1.5.1 Design of Security Mechanisms . . . . .	15
1.5.2 Benchmarking and Performance Evaluation . . . . .	17
1.5.3 Design of a Memory Control Interface . . . . .	18
1.6 Thesis Outline . . . . .	19
1.7 Publication Overview . . . . .	21
<b>2 Background</b>	<b>25</b>
2.1 Pervasive and Context-aware Computing . . . . .	25
2.2 Lifelogging . . . . .	27
2.2.1 Lifelogs as a Surrogate Memory . . . . .	28
2.2.2 Lifelogging Devices . . . . .	30
2.2.3 Extracting Useful Information from Lifelogs . . . . .	35

2.2.4	Legal and Ethical Issues . . . . .	39
2.3	Human Memory . . . . .	46
2.3.1	Human Memory Manipulation . . . . .	49
2.4	The Envisioned Threat Model . . . . .	50
 <b>II Manipulation Resistant Memory Capture and Sharing</b>		<b>53</b>
 <b>3 Secure Memory Capture and Storage</b>		<b>55</b>
3.1	Threat Model and Requirements . . . . .	57
3.2	Related Work . . . . .	59
3.3	Secure Memory Capture Using Trusted Sensors . . . . .	61
3.3.1	Camera Implementation . . . . .	62
3.3.2	Trusting the Camera . . . . .	62
3.3.3	Provisioning and Protecting Camera Key Material . . . . .	65
3.4	A Storage Protocol for Securely Linking Data . . . . .	65
3.4.1	Protocol Description . . . . .	66
3.4.2	Checking for Missing Images . . . . .	69
3.4.3	Generating the MAC token key TK . . . . .	69
3.4.4	Security Analysis . . . . .	70
3.4.5	Evaluation . . . . .	72
3.5	Limitations . . . . .	73
3.6	Chapter Summary . . . . .	74
 <b>4 Secure Memory Sharing</b>		<b>77</b>
4.1	Threat Model and Requirements . . . . .	79
4.2	Related Work . . . . .	81
4.3	A Systematic Approach for Memory Sharing . . . . .	84
4.3.1	System Requirements . . . . .	84
4.3.2	System Description . . . . .	85
4.4	Verifying Shared but Modified Image Cues . . . . .	91
4.4.1	Variant 1: A Practical Protocol Based on Hash Schemes . . . . .	92
4.4.2	Variant 2: A Protocol Based on Homomorphic Encryption . . . . .	95
4.5	Implementation . . . . .	101
4.6	Security Analysis . . . . .	102
4.6.1	Experience Sharing System . . . . .	103
4.6.2	Image Verification Protocol . . . . .	104
4.7	Evaluation . . . . .	105
4.7.1	Beacon Reception Rates and Proximity Range . . . . .	106

---

4.7.2	Runtime Overhead and Energy Consumption . . . . .	108
4.8	Chapter Summary . . . . .	110
<b>III</b>	<b>Memory Capture and Access Control</b>	<b>113</b>
<b>5</b>	<b>A Tangible Interface for Controlling Memory Capture and Sharing</b>	<b>115</b>
5.1	Related Work . . . . .	116
5.2	MemStone Interface . . . . .	119
5.2.1	Design Principles . . . . .	119
5.2.2	Interface Description . . . . .	121
5.2.3	Gestures and Control Actions . . . . .	122
5.2.4	Envisioned Usage Scenario . . . . .	123
5.2.5	Using MemStone to Control Infrastructure Sensors . . . . .	125
5.3	Phone-app as an Alternative Interface . . . . .	126
5.4	User Study . . . . .	127
5.4.1	Study Design and Procedure . . . . .	127
5.4.2	Recorded Meetings and Tasks . . . . .	129
5.5	Results: Interface Comparison . . . . .	130
5.5.1	Efficiency and Effectiveness . . . . .	130
5.5.2	Perceived Usability and Learnability . . . . .	132
5.5.3	Perceived Intuitiveness and Enjoyment . . . . .	133
5.6	Results: Long-term Gesture Memorability . . . . .	134
5.6.1	Follow-up Study . . . . .	135
5.6.2	Results . . . . .	136
5.7	User Perceptions of the MemStone Device . . . . .	138
5.7.1	Suggested Design Improvements . . . . .	138
5.7.2	Interaction Techniques . . . . .	139
5.7.3	Role of Device Visibility . . . . .	139
5.8	Discussion . . . . .	140
5.9	Chapter Summary . . . . .	144
<b>6</b>	<b>Access Control for Memory Augmentation Systems</b>	<b>147</b>
6.1	Control Requirements . . . . .	149
6.1.1	Controlling Experience Capture . . . . .	149
6.1.2	Memory Access Control . . . . .	151
6.2	Evaluation of Access Control Models . . . . .	153
6.2.1	Support for context-based capture and sharing . . . . .	153
6.2.2	Support for sharing based on interpersonal relations . . . . .	155

---

6.2.3	Support for multi-user governance . . . . .	156
6.2.4	Data obfuscation and sharing granularity . . . . .	157
6.3	Evaluation Summary . . . . .	161
6.4	Research Challenges . . . . .	164
6.5	Chapter Summary . . . . .	166
<b>IV</b>	<b>Conclusion and Future Work</b>	<b>169</b>
<b>7</b>	<b>Conclusion and Future Work</b>	<b>171</b>
7.1	Summary of Contributions and Results . . . . .	173
7.1.1	Securely Capturing and Storing of Experience Data . . . . .	173
7.1.2	Secure Memory Sharing with Co-located Others . . . . .	174
7.1.3	Verifying Shared but Modified Visual Cues . . . . .	174
7.1.4	In-situ Controls for Memory Capture and Sharing . . . . .	175
7.1.5	A Review of the Suitability of Access Control Models for Memory Augmentation Systems . . . . .	176
7.2	Future Work . . . . .	176
7.2.1	Additional Memory Manipulation Threats . . . . .	176
7.2.2	Verifying More Image Modifications Beyond Blurring . . . . .	178
7.2.3	Specifying Control Policies Through Abstractions . . . . .	178
	<b>Bibliography</b>	<b>181</b>



# Figures

1.1	Three steps memory augmentation process. . . . .	6
1.2	Prototypes of ambient displays for cue presentation. . . . .	8
1.3	Security design process. . . . .	15
2.1	Commercial lifelogging devices. . . . .	30
2.2	Experience capturing with wearable and fixed cameras. . . . .	33
3.1	Threat model for captured experience data. . . . .	57
3.2	System overview for secure capture and storage. . . . .	61
3.3	Our prototypical camera setup used for performance evaluation. . . . .	63
3.4	Overview of the file chain scheme. . . . .	66
3.5	Pseudocode of the storage protocol. . . . .	68
3.6	Camera runtime overhead. . . . .	73
4.1	Threat model when exchanging data among co-located peers. . . . .	80
4.2	Capturing and sharing lifelog data between co-located peers. . . . .	86
4.3	State diagram of the memory cue sharing process. . . . .	89
4.4	Pseudocode of the image verification protocol (variant 1). . . . .	94
4.5	The process of “blinding” a region of an image before sharing it. . . . .	95
4.6	Pseudocode of the image verification protocol (variant 2). . . . .	98
4.7	The beacon protocol data unit. . . . .	102
4.8	Attack tree for unauthorized access to peers’ captured data. . . . .	103
4.9	Beacon reception rates for three token-pubkey ratios. . . . .	106
4.10	Runtime overhead of Protocol 2. . . . .	108
4.11	Camera estimated operational time on a single battery charge. . . . .	109
4.12	System overview for secure exchange of cues. . . . .	111
5.1	Overview of the developed MemStone prototype. . . . .	121
5.2	MemStone gestures. . . . .	123
5.3	MemStone usage scenario. . . . .	124

---

5.4	A phone app as an alternative of MemStone. . . . .	126
5.5	Device average task completion rate. . . . .	131
5.6	Device average task completion time. . . . .	132
5.7	Illustration of a question from the follow-up study. . . . .	135
5.8	MemStone long-term gesture memorability results. . . . .	137
6.1	A secure architecture for memory augmentation. . . . .	148

# Tables

1.1	Overview of research questions in this thesis. . . . .	12
1.2	Overview of contributions of this thesis. . . . .	14
1.3	Publication overview. . . . .	23
3.1	List of variables used for the storage protocol. . . . .	67
4.1	Variables used in the system for sharing memory cues. . . . .	87
4.2	Variables used in the protocol for verifying shared images. . . . .	92
5.1	List of tasks and corresponding device actions. . . . .	129
5.2	Task sequence for both videos. . . . .	130
6.1	Elicited control requirements. . . . .	150
6.2	Evaluation of access control systems against control requirements. . . . .	162



# **Part I**

## **Introduction and Background**



# Chapter 1

## Introduction

For millennia humans have been recording, storing, and passing on information in order to advance knowledge, as well as to preserve our most valuable memories. Whether carving on walls, painting on paper, taking a photograph, storytelling, or writing, we have always attempted to create various *memory aids* (or cues) that would allow us to conserve our fading memories over time. With the dawn of the “information era”, exhibited by pervasive smart computers and sensors that are connected all the time<sup>1</sup>, we have seen remarkable changes towards how we can capture, store, and share information. Taking a high-resolution photograph, tracking our location, counting our steps, or going live on social media, are only a few examples of the *digital memory aids* that we can create and disseminate in a very straightforward fashion nowadays.

However, the transition into an information and knowledge society has also changed our lifestyle towards a more fast-paced and stressful one, requiring us to handle more and more information at a time. Not surprisingly, this has had a negative impact on our ability to remember and recall many aspects of our mundane activities. A recent study [1] suggests that chronic stress might be one of the reasons why we often fail to remember e.g., “where we left our car keys”, or “if we have to pickup our kids from school today”. Despite such consequences, technology might still be a means of supporting and augmenting our memories in this rich-information era.

Early ideas to designing technological artifacts to support human memory date back to the concept of the “Memex”, proposed by Vannevar Bush in his seminal article “As we may think” from 1945 [2]. Irritated by the fact that for years

---

<sup>1</sup>According to a report by Cisco IBSG 2011, the number of Internet-connected devices has seen a massive growth in the last 13 years (going from 500 million in 2003, to 12.5 billion in 2010, to 50 billion by 2020) clearly making the number of connected devices per person more than 5.

inventors had focused on extending human's physical powers rather than their "powers of mind", Bush urged scientists to design solutions that will improve the accessibility of the bewildering store of knowledge of that time. His article introduced a straw-man description of his Memex machine to both capture and store knowledge from different scientific articles and books. Fast forward some 50 years, "lifelogging" pioneers Bell and Gemmell started "MyLifeBits" [3], initially envisioning to create a modern Memex system that would capture archival materials (e.g., computer files, scanned books, and digitized music), but later also capturing *real-time and continuous content streams* (e.g., phone calls, meetings conversations, and first-person photos as produced by wearable cameras).

Fast forward another 20 years, and we can take the idea of "augmented memory" to a completely different new level. We are able to capture more information than MyLifeBits could, e.g., contextual information, location traces, physiological state, to name but few. Recent advances in big data analytics (e.g., deep learning), as well as in data visualization (e.g., ambient displays) provide us with cutting-edge tools for designing novel approaches to memory augmentation. The principle idea is to use emerging unobtrusive capture technologies (such as wearable and mobile devices) to capture a rich representation of our everyday experiences. The casual review of such experiences will allow then users to refresh and reinforce existing memories, a process known in psychology as *cued-recall*.

Maybe obviously, a system that captures everywhere we go, and everything we say, see, and do, is greatly increasing the danger to our privacy. Any data breach of a such memory repository, whether accidental or malicious, could have significant repercussions – starting with negatively impacting our (professional and private) reputation, up to risking physical harm (e.g., targeted assaults). Even more than prior pervasive systems, the design of such architecture requires one to thoroughly include privacy consideration at design-time [4].

The threat of infringing users' privacy is not the only worrisome implication of such systems. Research from the field of psychology has shown that the aforementioned cued-recall process does not only allow us to reinforce our past memories but can also attenuate them [5, 6]. In practice, this means that the cued-recall process can be misused to reinforce a particular set of memories and decrease the ability to recall other memories, hence manipulate our overall memory of prior experiences. In fact, there is strong evidence that our memories can be manipulated almost at will. In a recently published book [7], UCL researcher Julia Shaw, finds that "even the precious memories of our childhood can be be actually shaped and reshaped like a ball of clay". Through a series of experiments, Shaw was able to implant "full false memories" in 70% of her study participants,



making them remember details of prior experiences that never took place [7]. Obviously, such memory manipulation attacks would be greatly amplified in the context of our envisioned technology-based memory augmentation.

This thesis thus investigates how we design *secure* memory augmentation systems. Particularly, we focus on the two above-mentioned challenges of user privacy and manipulation of user memory. In the course of this research, we develop tools and prototypes that can be applied by researchers and system engineers to develop pervasive applications that help users capture and later recall episodic memories in a secure fashion. We built trusted sensors and protocols to securely capture and store experience data, and secure software for the secure and privacy-aware exchange of experience data with peers. Regarding the user privacy challenges, we focus in only one aspect of privacy, that is, the support for modifications of visual experience data (images). The other aspects of “classical-privacy” such as confidentiality, anonymity, or data usage policies are out of scope of this research. We furthermore explore the suitability of various access control models to put users in control of the plethora of data that the system captures on their behalf. We explore the possibility of abstracting most of the control logic via simple-to-use and easy-to-remember physical gestures. Ultimately, this thesis contributes to the design and development of secure systems for memory augmentation.

## 1.1 Vision: Pervasive Memory Augmentation

The proliferation of sensors-rich mobile devices such as smartphones, smartwatches, wristbands, wearable cameras has made technology more intertwined with our life. This technological “invasion” gives us the possibility to digitally capture our daily experiences in increasing numbers and quality, a process known as *lifelogging* [8]. Its early adopters have created the quantified-self movement – the quantification of one’s daily activities (e.g., steps taken, calories burned, sleep patterns, heart rate, etc.) in order to better manage one’s life following the old adage “You can’t manage what you can’t measure”.

Beyond such use, it is not hard to imagine that captured experiences can also support and augment human memory. In this vein, lifelogging offers a powerful new set of tools that can radically change the way how we can put technology to support our overall memories. In the following, we map out the vision of pervasive memory augmentation systems and how they can be used to facilitate the recollection of our prior memories.

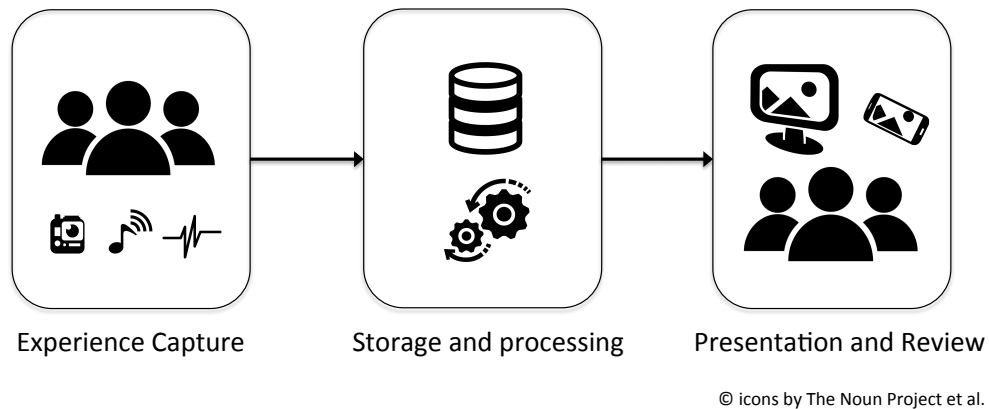


Figure 1.1. Three steps memory augmentation process.

We envision a memory augmentation system as a three step process [9] (as shown in Figure 1.1). At the outset, one captures different aspects of daily activities (step 1) using unobtrusive capture technology, such as wearable cameras, smartglasses, or smartphones. Activity data is then processed (step 2) in order to build carefully selected memory triggers (memory cues) such as a set of photos of the activity, the type of the activity, the environment where it happened, the main conversation themes, the weather conditions, etc. The extracted memory cues are then repeatedly and unobtrusively presented back to users in suitable moments and through ambient displays (step 3), for instance on a mobile phone’s lock screen, as a laptop’s screen saver, or on a picture frame in the living-room.

The casual review of memory cues will allow then users to refresh and reinforce existing memories. This process is known in psychology as *cued recall*. In this respect, a memory cue is simply a piece of information that, when reviewed, can help one to retrieve the memories associated with it. Consider how a photograph can bring back distant memories from childhood, how a song can help one reminisce about a past vacation, or how a set of keywords can help one remember details from a previous work meeting.

Many prior memory supporting approaches (e.g., Memex or MyLifeBits) are designed as look-up data stores which one can then query in order to retrieve the missing memories. Unlike this approach, we envision a system that would automatically deliver relevant memory cues based on user-defined goals regarding what one would like to remember, e.g., remembering faces of new encounters, or preparing for an upcoming exam. By constantly reviewing the presented memory cues, one can then *train* their memory, so that ultimately, those memories can be recalled without the help of any tool.

Users will predominantly will capture their daily activities using their own capture devices. This means that the quality and the type of the cues will be limited to what one's own device has managed to capture. From our experience working with wearable cameras, we have observed that images captured by them do not lend well as memory cues. Very often, the camera lens gets obscured by the wearer's hair or clothes. Even when having a clear view, the camera's narrow field of view may fail to capture *key moments* of the activity. However, the wearable cameras of other users that are in close physical proximity, as well as any infrastructure camera, could have captured those key moments. For instance, while one's own camera will fail to capture a person sitting right next to the user, the camera of the person sitting in front can. Therefore, we envision a memory augmentation system that will also exchange data it captures between co-located peers and infrastructure sensors, so that ultimately the produced data stream will capture a great amount of details.

In short, a memory augmentation system seamlessly captures a user's daily activities, extracts memory cues and other background contextual information (e.g. type of event, or other users nearby), and considers the user's memory preferences (e.g., which memories would one like to reinforce or to attenuate), before finally deriving a review schedule for such cues. It also automatically exchanges captured data with co-located users in order to deliver any information that the user's camera may have failed to capture. The system further provides mechanisms that allow users to customize the capturing and sharing preferences and support users in-situ according to their current situation.

### 1.1.1 Memory Cue Sources

In principle, almost anything can work as a memory cue that can help us remember: a written note can help us remember what was discussed in a meeting, a tied knot on the finger can remind us to pickup our friends from airport, a glass of sand can help us reminisce about the vacations at the beach, the smell of a cake can bring back memories of our last birthday celebration, etc.

According to psychology research on human memory, visual information, such pictures or videos, are of a particular interest for memory recall [10]. Thanks to their rich-level of information, visual data offer the most effective memory cues. Given today's capture technology (in particular body-worn cameras), visual memory cues live at the sweet spot of both ease of capture and recall power. The work in this thesis, therefore, focuses on visual information as a central data source for generating what we call *primary memory cues*.



(a) As a computer screensaver



(b) As a smartwatch flashcard



(c) As a slideshow on the living room TV



(d) As picture frame content

Figure 1.2. Conceptual prototypes of ambient displays showing visual and/or textual memory cues.

In certain applications, such as work-related meetings, seminar talks or lectures, audio data can also provide valuable memory cues. Their memory-recall power stems from the fact that they can provide topic-based memory cues (derived through topic modeling techniques). These cues can then be combined together with any visual memory cues captured during the same time, with a goal of delivering more effective cues.

Beyond audio-visual data, there is a number of other data sources which can generate what we call *secondary memory cues*. Common secondary cues can be derived from context information that is available in most of today's smartphones and smartwatches, namely, time, location, acceleration, light levels, temperature, blood pressure, or galvanic skin response. These secondary cues can either complement the primary audio-visual cues (for instance to assess the significance of a selected primary cue), or instead used as independent memory cues on their own.

## 1.1.2 Cue Presentation

Once memories are captured and processed, they must be delivered back to users at the right moment in useful and attractive visualizations. The proliferation of ubiquitous displays via personal devices (e.g., smartphones, smartwatches, or smartglasses), as well as in the environment (e.g., computer screens, photo frames, or large-wall-mounted displays) provides new opportunities for displaying and reviewing memory cues. Figure 1.2 illustrates several conceptual designs of such peripheral displays that can be used to show relevant memory cues according to the users' given context (i.e., time and location). By using such an ambient review of experience data over a range of timescales, users should be able to enhance memories of various past activities when needed, an effect that should ultimately persist without the support of any visualization.

## 1.1.3 Sample Scenarios

In the following we provide two scenarios that attempt to illustrate how pervasive memory augmentation systems can be used in real life. The scenarios also discuss the motivation, use cases, benefits of sharing experience data with others, stakeholders, as well as privacy concerns inherent in pervasive memory augmentation.

### Reliving Past Experiences Through Digital Memories

*Craig is an IT consultant working for an international company. During his job he has to travel to different places in order to meet and talk to his clients. On most of his trips, Craig takes his capturing gear with him which is composed of a wearable camera that can both take geo-tagged pictures and record short video snippets, as well as a wristband to record his bio-physiological responses. In addition to his work activities, he enjoys exploring the places he visits and engages himself in sightseeing activities. He has recently experimented with a technique to automatically compile some highlights (out of the vast amount of captured data) to help him review his memories of recent periods (e.g., last days, weeks, months and even years). In order to further optimize the highlights selection, Craig would like to let the system know his memorization goals (i.e., the things he wants to remember most). For instance he may like to see more images captured while he was highly engaged on his job. Moreover, he also likes to share some selected highlights with his co-workers and friends. In this case, even though sharing some sight-seeing moments would not be a problem for him, clearly he wants the generated data share to be related to the*

*work activities. He also wants to share the social facet of his travel experience with his family and his online friends. In both cases Craig would not like to share moments that he finds embarrassing (e.g., he ended up reading a knitting magazine during lunch time, as he was very bored but could not find any other magazine) or sensitive (e.g., data showing his computer screen or images showing bystander faces).*

This scenario describes how Craig uses captured memories to re-experience previous activities and re-live positive feelings linked with those activities. Data that Craig records is sourced from his personal wearables that capture images, audio, video, location traces, and physiological responses. Finally, this scenario also highlights the privacy issues that can emerge due to sharing potentially sensitive data with the wrong audience (e.g., sharing data of a business pitch with his friends), or sharing data of embarrassing moments (e.g., reading a knitting magazine during lunch time).

### **Sharing Digital Memories with Co-located Others**

*Dorothee is an accountant working for a local company. She is an enthusiastic lifelogger, and she always has her wearable camera to capture her daily activities. However, often she finds out that her camera has missed to capture various important situations, due to the camera lenses being covered by her hair or obscured by clothes, or simply pointing out to the wrong direction. Moreover, she never sees herself in those pictures. During the morning walk to her working place, she stops at a local restaurant for a coffee. In order to not miss much from these moments, she would like to access any infrastructure camera (e.g., in the street and in the restaurant) that captures her walk in a third-person perspective, yet preventing their owners from easily tracking her location. While entering the department building she meets a colleague in the hallway. Since they both want to have a more comprehensive capture of this personal encounter (compared to what their wearable cameras offer) they want to exchange captured pictures with each other. In the afternoon Dorothee attends a work meeting. In order to better remember this meeting, she wants access to the data captured from the room's built-in high-quality sensors (camera, microphone, board contents, etc.), as well as data captured from other colleagues. However, people who simply pass by the meeting room should not have access to this data. During the meeting break, Dorothee writes emails from her laptop. She would not like her colleagues to get access to any data that shows content from the laptop screen (both captured by the room's sensors and her own wearable camera). After the meeting, while packing her bag, she has a chat with*

*a colleague. Even though high-quality capture of the meeting room is stopped, they still want to exchange data from their wearable cameras. Other colleagues who have already left the room should not have access to this data.*

Beyond data captured by her wearable camera, in this scenario, Dorothee is also interested in accessing data sourced from infrastructure sensors capturing her activities, e.g., pictures from the in-street fixed cameras or audio from the meeting room’s microphone. Furthermore, she wants to seamlessly exchange pictures captured from the wearable cameras of other co-located people. Data coming from such “additional” sources can complement the data captured solely by user’s personal devices in creating a more comprehensible representation of the captured experience. In light of this, this scenario highlights some privacy issues: oversharing event-related data with people who were not part of the event and were simply passing by or sharing highly-sensitive data, e.g., photos showing laptop screen while writing a confidential email during the meeting break.

## 1.2 Research Questions

To investigate how we can design secure systems for human memory augmentation, we considered two main aspects, namely, (i) the security of experience data that constitutes users’ memories, and (ii) access and privacy controls of experience data when exchanging this data with others. In Table 1.1 we list the corresponding research questions which have driven the research presented in this thesis.

A system that captures and stores different aspects of our everyday activities can give us immediate access to a stream of information regarding our previous experienced activities. Whilst reviewing any memory cue can help us in imparting and “validating” our memories of past events, such process can be put to wrong use in order to distort and manipulate our overall memories. This raises significant security implications for any memory augmentation system. Imagine that an attacker compromises our memory augmentation system and gets access to our complete collection of captured data. Besides accessing personal and sensitive information depicted in this data, the adversary can try to modify the stored experiences, inject new fabricated data, delete existing data, control the data selection process to ultimately attenuate some of our memories while reinforcing others, or even fabricate memory of events that we never experienced. With the goal of addressing these risks of memory manipulation attacks, here we focus on a technological solution that can ensure security aspects, namely, authenticity, integrity and provenance of captured experience data (**RQ1**).

No.	Research Question
RQ1	How can we guarantee digital memory integrity and provenance to prevent memory manipulation attacks?
RQ2	How can we seamlessly and securely share captured experiences with co-located others, avoiding the risk of accidental oversharing, i.e., sharing with the wrong audience, or sharing parts of a capture that we would otherwise have kept to ourselves?
RQ3	How can we verify the integrity and provenance of experience data which we obtain from others to detect the sharing of falsified experience captures?
RQ4	What interfaces and policy-based access control models can we use to exercise control over data capture as well as to prevent the disclosure of private and sensitive information when sharing experience data?

Table 1.1. Overview of research questions in this thesis.

While captured experience data is chiefly used for capturing personal memories, there are at least two benefits of *sharing* this data. Perhaps it comes to no surprise that sharing experience data allows us to reminisce together with others about an event, or to show others which were not there what they missed. This is very similar to the practice of sharing pictures and videos on social media. We refer to this process as *explicit memory sharing*. Another possibility is to *implicitly* exchange captured memory experiences with co-located peers. Combining streams from co-located peers allows us to create more comprehensive data streams that go beyond the physical capture limitations of our own sensors. Our research thus focuses on an approach to first detect co-located experiences, and then seamlessly and securely exchange data captured during that time among co-located users (**RQ2**).

The possibility of receiving and reviewing any experience data shared from others requires us to revisit the risks of memory manipulation. Memory sharing may allow malicious peers to send us falsified data streams that do not present an accurate reflection of the experienced event during which the shared data was allegedly captured. Hence, we focus on a two-party protocol to verify shared but modified experiences (**RQ3**). Such protocol would account for any modification carried on by the data sharer for privacy-preserving reasons, and subsequently allow the recipient of such data to verify the claimed modifications without accessing the original unmodified data.



Regardless of how we share such data, either explicitly or implicitly, sharing any sensitive information that may be depicted on them can infringe our privacy. Therefore, we investigate the opportunities of controlling different aspects regarding the capturing and sharing of experiences through *in situ* physical gestures. We furthermore investigate the use of policy-based control mechanism in order to have finer-grained control on the practices of capturing and sharing experience data (RQ4).

### 1.3 Research Context: The RECALL Project

The work presented in this thesis was carried out within the EU research project RECALL<sup>2</sup>. The RECALL project was funded through the Future and Emerging Technologies (FET) programme within the 7th Framework Programme for Research of the European Commission, under FET grant number: 612933. Over the course of three years (November 2013–October 2016), four consortium partners (Lancaster University, University of Stuttgart, Essex University, and USI) collaborated closely with the goal of rethinking and redefining the notion of technology-driven human memory augmentation. The vision of pervasive memory augmentation systems reported in this thesis is a fundamental part of the RECALL project and is the result of joint effort from all partners. This collaboration resulted in several joint publications [11, 12, 13, 14], as well as a workshop on the topic of “Mobile Cognition” at the 2015 International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI’15) [15].

### 1.4 Thesis Goals

Pervasive technologies allow us to already build systems that support and augment our memories of everyday experiences. In the course of this research, we set out with the goal of investigating how we can design *secure systems* for human memory augmentation. While challenges such as experience data confidentiality and privacy are most likely the largest user-concerns of this kind of technology, in this thesis we, however, focus on a potentially far more serious attack space: risks of memory manipulation that stem from security related issues. A second challenge that we aim to address is the design of efficient mechanisms for controlling the practices of capturing and sharing of experience data with others.

---

<sup>2</sup><http://recall-fet.eu>

We tackle these two challenges by utilizing and combining together several concepts, namely, primitives from computer security and cryptography, pervasive sensing, short-range communication technologies, and the concept of tangible interfaces. In designing our solutions we conducted formative and summative experiments, as well as security and performance evaluations. We also incorporated key design principles stemming from research efforts on secure pervasive systems, namely 1) the socio-technical view of pervasive systems, 2) context-awareness of these systems, and 3) resource-constrained operation environments [16, 17, 18]. Table 1.2 provides a summary of the contributions delivered in the course of this research.

<b>Contribution</b>	<b>Publication</b>
A trusted camera sensor coupled with storage protocol to securely capture and store experience data	[19]
A mobile system to seamlessly and securely exchange experience data with co-located peers	[14]
A two party protocol to verify integrity of shared but modified visual experience data	[19]
A tangible interface for controlling capture and sharing of experience data with in-situ gestures	[20]
A critical review of the suitability of policy-based access control models with regard to requirements of pervasive memory augmentation systems	[21]

Table 1.2. Overview of contributions of this thesis.

## 1.5 Methodology

In this section we describe the methodology we followed for addressing the aforementioned thesis goals (see section 1.4), that is, building and evaluating secure systems for human memory augmentation. In principle we design and develop security protocols and architectural components which we deploy on low-power mobile devices. We then evaluate their security and performance, as well as usability and user experience aspects. Therefore, our research methodology is grounded in cross-disciplinary research methods from *computer security*, *performance evaluation of computer systems* and *user-centered research approach*.

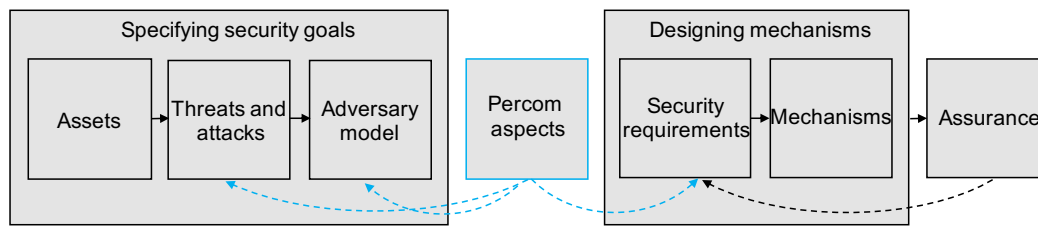


Figure 1.3. The process that we followed for designing the security mechanisms as described in [23].

### 1.5.1 Design of Security Mechanisms

There are a plethora of definitions of what computer security is about, often attaching different meanings to it in different contexts. In the context of this work, we borrow Ross Anderson’s view on security engineering:

*“Security engineering is about building systems to remain dependable in the face of malice, error, or mischance. As a discipline, it focuses on the tools, processes, and methods needed to design, implement, and test complete systems, and to adapt existing systems as their environment evolves.”*

Ross Anderson [22]

In this work we adopted a common viewpoint where security is perceived as *risk management* [16]. To realize this viewpoint we followed a well-established process for providing computer security [22, 23, 24]. Generally, this process requires three things to come together: 1) defining security goals, 2) designing protection mechanisms for mitigating such threats, and 3) evaluating the assurance of the proposed solution (i.e., how much reliance you place on designed mechanisms). However, many of the underlying security concepts to realize the aforementioned viewpoint may not directly apply to pervasive memory augmentation systems. Therefore, we also looked at security aspects applied to emerging applications from pervasive computing domain [16, 17, 18].

Figure 1.3 describes the process that we followed in designing the security mechanisms. The first step is to develop the security policy, i.e., identifying assets that need to be protected and understanding the threats and risks. Thus, we identified memory cues as the most critical *assets* of a memory augmentation system that need to be protected. Next, we analyzed each of the three stages of the memory augmentation process individually (which we described in section 1.1): 1) experience capture; 2) data storage and processing, and 3) memory

cue presentation and review. For each stage we identified the following:

1. *Threats*: represent possible danger that might exploit certain flows or weaknesses of a memory augmentation system in order to violate the security of memory cues. For instance, capture experience data can be modified to not necessarily reflect an actual experienced event.
2. *Attacks*: deliberate attempts that derive from threats in order to gain unauthorized access or to make unauthorized use of a memory cue asset. For instance, a malicious person can compromise one's capture gear or memory repository in order to intentionally modify experience data streams.
3. *Adversary model*: assumptions of what resources an attacker has access to and what attacks they can perform. For example, one envisioned adversary could be the service provider where experience data is stored. Such an adversary could have access to one's complete data stream.
4. *Risks*: the envisioned consequences should an attack happen. Typically, we foresee two chief risks, i.e., manipulating users' memories and infringing their privacy.

For each threat and corresponding attack we then delineated a set of *security requirements* in order to ensure three fundamental security properties of memory cues:

1. *Provenance*: ensuring that memory cues reflect an actual representation of an original experience;
2. *Confidentiality*: protecting the disclosure of memory cues to unauthorized others; and
3. *Integrity*: ensuring that memory cues are not maliciously altered by others.

As part of our research process, we also looked at research efforts related to secure pervasive systems. During this step, we identified three key aspects (but also opportunities) that are not easily captured through classical security research frameworks, but that are relevant to our work [17]. Such aspects are then used as additional input to the specification of threats, attack model, and security requirements (see Figure 1.3). These aspects are the following:

1. *Socio-technical systems*: the secure design of pervasive systems requires one to consider issues such as usability and trust in the proposed solutions. Unlike traditional systems which imply static trust relationships, here trust is dynamic and changes from context to context.

2. *Context-awareness*: context is a piece of information that can be attached to users, devices, and the environment. Therefore, context can be considered as the interface connecting both the social and technical aspects of such systems. It provides additional information that can be used to address a specific security action.
3. *Resource-constrained environments*: while pervasive systems are characterized as sensor-rich ecosystems, they are often limited in terms of computational resources. This limits the selection of security mechanisms (i.e., cryptographic protocols) that can be used.

Once we had defined the attack model and elicited a set of security requirements, we then proceeded with the design of the necessary security mechanisms. Throughout this phase we incorporated a combination of knowledge, best practices, and tools from computer security and cryptography, including tamper resistance (trusted platform modules), public-key cryptography, digital signatures, encryption, hash functions, and message authentication codes. We will introduce these concepts and tools in more detail in chapters 3 and 4.

Last, but not least, we evaluated the proposed security mechanisms with respect to the security requirements. We analyze the proposed protocols and present informal proofs and arguments why our solutions offer an acceptable level of security.

### 1.5.2 Benchmarking and Performance Evaluation

To validate the practical feasibility of the proposed protocols, we follow common methods and approaches for evaluating the performance of computer systems. Many such methods consider three key factors that influence the performance of a system: the system's *design*, the system's *implementation*, and the system's *workload* [25]. Out of these three factors performance is dramatically affected by the workload to which a system is exposed to. Furthermore, a good performance evaluation methodology requires three things to come together, i.e., defining *the workload*, *the metric*, and *the goals* of the test [26].

The workload describes the type of the request submitted to a system (either by a user or by a scheduled process). Consider for instance that we want to benchmark a wearable camera. The camera's load can be characterized by the number of pictures it takes per second. If a camera also computes a cryptographic hash of every image it captures, then these computations could probably take a significant fraction of its overall load. In our analysis we generally employ

similar workload patterns from other camera devices (e.g., the Narrative Clip or Microsoft SenseCam wearable cameras).

Once we have a workload defined, we need to precisely specify the metrics, that is, what quantities do we want to measure. For our evaluation, we typically use four kinds of metrics: power consumption (electricity consumed by the system per time unit), system battery runtime (the amount of time the system would run with a single battery charge), response time (time it takes to complete the execution of a specific operation), and throughput (the number of operations performed per time unit).

The next step requires the specification of the evaluation method. Generally there are three different approaches: 1) directly measuring the real system, 2) measuring a software-implemented simulation of the system, or 3) analyzing a mathematical representation of a system. Since in our work any validation should entail external validity, we follow the first approach, i.e., we deploy our prototypes directly on low-power mobile devices. Therefore, we are able to collect more accurate measurements that correspond to real-world deployments.

Note that our goal is to show that our solutions can run fast enough and that it is feasible and acceptable to build them on resource constrained mobile devices – we are not aiming to design (and implement) the most power efficient and/or the fastest executable solutions.

### 1.5.3 Design of a Memory Control Interface

As part of this research, we also investigate empowering users with more control over their captured experiences. To this end we design and build *MemStone*, a TUI operated by five physical gestures that allows one to in-situ control different aspects of the data capture and sharing practices.

We design, build, and evaluate this interface following a user-centered design (USD) process [27]. The UCD is a four stage iterative process guiding researchers and designers in creating interactive products with a focus on *usability*. At the outset, one envisions the context and the intended use of a system (stage 1), which then leads to a set of user requirements (stage 2). Based on these requirements, one develops a solution prototype (stage 3), and finally evaluates it against the initial context and user requirements (stage 4).

We design our MemStone prototype based on challenges, requirements, and design principles that we extract from prior research in the fields of lifelogging, memory augmentation, and tangible interfaces. We administer a lab study with a goal to evaluate the usability and perceived usefulness of the MemStone interface, but also to investigate how our envisioned gesture interactions compare

to a more traditional interaction alternative (i.e., a smartphone). We recruit 20 participants, and employ a within-subject design. Each participant uses both our MemStone interface and a smartphone app in a meeting capture scenario. Participants thus try to control different aspects related to the capturing and sharing of memories of the meeting. Besides quantitative data including task efficiency, effectiveness, perceived interface usability, and learnability, we also collect qualitative feedback by inviting participants to an open-ended discussion session in order to unfold further concerns, factors, and opportunities that can improve the design of the interface.

We also administer a follow-up study with the goal of evaluating participants' long-term gesture memorability. Good and easily remembered gestures can reduce mistakes, frustration, and the time to learn them. At the same time such gestures can increase enjoyment, usability, and hence lead to better device adoption rates. We conduct this second study four months after the first primary study, contacting all participants from the first study by email and inviting them to participate in a short online survey.

## 1.6 Thesis Outline

This thesis consists of seven chapters and the bibliography, and is divided into four parts. In Part 1 we describe the motivation and the vision behind pervasive memory augmentation systems. Part 2 describes our system for capturing and sharing of experience data with the goal of preventing human memory manipulation attacks. In Part 3 we focus on access control of captured data in order to address some of the privacy issues inherent in the practice of sharing experience data. In Part 4 we summarize the work, describe our contributions, and present directions for future research in the context of secure systems for memory augmentation. Next we provide a brief description of each chapter.

### Part 1 – Introduction and Background

- **Chapter 1 – Introduction**

In the first chapter we describe the motivation and vision for pervasive memory augmentation systems. We then describe the research context in which this thesis was conducted, list the research questions, state the contributions made, provide an overview of the threat model, and describe the research methodology that we followed.

- **Chapter 2 – Background**

This chapter introduces the key concepts and technologies that this thesis is structured in. We then briefly describe the structure of human memory and highlighting how our memories can be manipulated with falsified images.

## **Part 2 – Manipulation Resistant Memory Capture and Sharing**

- **Chapter 3 – Secure Memory Capture and Storage**

In this chapter we address the challenge of preventing human memory manipulation in pervasive memory augmentation systems. We explore the different ways how such attacks can be executed in practice. We then design and build a systematic and practical solution based on a trusted wearable camera that addresses the elicited threats.

- **Chapter 4 – Secure Memory Sharing**

Building on top of the results from the previous chapter, in this chapter we investigate the possibility of securely sharing captured experiences with others. Consequently, we propose a system that will implicitly and securely exchange images among co-located peers, as well as a protocol to verify any such shared but modified images obtained from others.

## **Part 3 – Memory Capture and Access Control**

- **Chapter 5 – A Tangible Interface for Controlling Memory Capture and Sharing**

In this chapter we investigate the feasibility of controlling the capture and sharing of experience data through in-situ physical gestures. We present MemStone, a prototype of a tangible user interface (TUI) that allows users to control access to (and the sharing of) captured memories in-situ. We report analysis of our user study with 20 participants with the goal of investigating the suitability of MemStone, as well as comparing the usability and efficiency of MemStone with a mobile app user interface.

- **Chapter 6 – Access Control for Memory Augmentation Systems**

This chapter presents our analysis of evaluating the suitability of existing context-aware access control models towards our security requirements for memory augmentation systems. Our goal is to inform and motivate further research on the design and development of access control solutions suitable for this kind of systems.




## Part 4 – Conclusion and Future Work

- **Chapter 7 – Conclusion and Future Work**

In this chapter we summarize findings from previous chapters, state the contributions of this thesis, and provide directions for future research.

## 1.7 Publication Overview

Parts of this thesis have been published in peer-reviewed conferences, magazines and workshops. For a publication overview, see Table 1.3.

No.	Publication
1	A. Bexheti, M. Langheinrich, I. Elhart, and N. Davies, “Securely Storing and Sharing Memory Cues in Memory Augmentation Systems: A Practical Approach,” in <i>Proceedings of the 17th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom’19)</i> , 2019, p. 10  <i>The Mark Weiser Best Paper Award</i>
2	A. Fedosov, A. Bexheti, E. Ermolaev, and M. Langheinrich, “Sharing Physical Objects Using Smart Contracts,” in <i>Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct</i> , ser. MobileHCI ’18. New York, NY, USA: ACM, 2018, pp. 346–352
3	A. Bexheti, A. Fedosov, I. Elhart, and M. Langheinrich, “Memstone: A Tangible Interface for Controlling Capture and Sharing of Personal Memories,” in <i>Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services</i> , ser. MobileHCI ’18. New York, NY, USA: ACM, 2018, pp. 20:1–20:13
4	E. Niforatos, M. Laporte, A. Bexheti, and M. Langheinrich, “Augmenting Memory Recall in Work Meetings: Establishing a Quantifiable Baseline,” in <i>Proceedings of the 9th Augmented Human International Conference</i> , ser. AH ’18. New York, NY, USA: ACM, 2018, pp. 4:1–4:7
5	A. Bexheti, M. Langheinrich, and S. Clinch, “Secure Personal Memory-Sharing with Co-located People and Places,” in <i>Proceedings of the 6th International Conference on the Internet of Things</i> , ser. IoT’16. New York, NY, USA: ACM, 2016, pp. 73–81

- 
- 6 **A. Bexheti**, E. Niforatos, S. A. Bahrainian, M. Langheinrich, and F. Crestani, “Measuring the Effect of Cued Recall on Work Meetings,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. New York, NY, USA: ACM, 2016, pp. 1020–1026

---

  - 7 T. Dingler, P. E. Agroudy, H. V. Le, A. Schmidt, E. Niforatos, **A. Bexheti**, and M. Langheinrich, “Multimedia Memory Cues for Augmenting Human Memory,” *IEEE MultiMedia*, vol. 23, no. 2, pp. 4–11, Apr. 2016

---

  - 8 **A. Bexheti** and M. Langheinrich, “Understanding Usage Control Requirements in Pervasive Memory Augmentation Systems,” in *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM ’15. New York, NY, USA: ACM, 2015, pp. 400–404

---

  - 9 T. Dingler, **A. Bexheti**, E. Niforatos, and F. Alt, “Workshop on Mobile Cognition: Using Mobile Devices to Enhance Human Cognition,” in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, ser. MobileHCI ’15. New York, NY, USA: ACM, 2015, pp. 970–973

---

  - 10 E. Niforatos, V. Lim, C. Vuerich, M. Langheinrich, and **A. Bexheti**, “Pulse-Cam: Biophysically Driven Life Logging,” in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, ser. MobileHCI ’15. New York, NY, USA: ACM, 2015, pp. 1002–1009

---

  - 11 **A. Bexheti**, A. Fedosov, J. Findahl, M. Langheinrich, and E. Niforatos, “Re-Live the Moment: Visualizing Run Experiences to Motivate Future Exercises,” in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, ser. MobileHCI ’15. New York, NY, USA: ACM, 2015, pp. 986–993

---

  - 12 E. Niforatos, M. Langheinrich, and **A. Bexheti**, “My Good Old Kodak: Understanding the Impact of Having Only 24 Pictures to Take,” in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, ser. UbiComp ’14 Adjunct. New York, NY, USA: ACM, 2014, pp. 1355–1360
-

- 
- 13 K. Wolf, A. Schmidt, **A. Bexheti**, and M. Langheinrich, “Lifelogging: You’re Wearing a Camera?” *IEEE Pervasive Computing*, vol. 13, no. 3, pp. 8–12, Jul. 2014
- 

*Table 1.3.* Publication overview.



# Chapter 2

## Background

Work presented in this thesis is structured in the field of **ubiquitous and pervasive computing**. The vision of a technological memory surrogate presented in the previous chapter, furthermore benefits from the technical achievements in the field of **lifelogging**. Beyond these fields of the computer science discipline, we also look at research on human memory and memory manipulation attacks stemming from the field of **cognitive psychology**. In this chapter, we briefly describe such foundation blocks before concluding with a description of the research methodology of this thesis.

### 2.1 Pervasive and Context-aware Computing

*“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.”*

Mark Weiser [33]

In 1991, Mark Weiser published an article describing his vision on the computer of the 21<sup>st</sup> century. This vision has led to the creation of a new computing paradigm that we know today as “ubiquitous” or “pervasive computing”.

Going back in the late 1950s, a single *mainframe* would provide computing services to multiple users. In the 1980s, the computing norm moved closer to a one-to-one relationship where typically one person owned a personal computer.

Weiser envisioned that technology will be pervasive and will “take into account the natural human environment and will allow the computers themselves to vanish into the background” [33]. According to Weiser, technology will go

beyond desktop computing, and will take many different forms, from laptops and tablets, to a pair of glasses or a fridge.

Nowadays, thanks to advances in hardware (smaller form-factor with higher transistor density), communication (fast wireless networks), as well as sensing (different mobile and wearable sensors), we can realize most aspects of Weiser's vision. Computing of today has transformed into a many-to-one model, where a single person has multiple devices. This includes laptops, phones, watches, eyeglasses, or any other gadget with a computer in it.

The aim of pervasive computing is not only to add computing power to everyday objects, but also to make them "smart" and "context-aware". According to Abowd et al. [34] "*when humans talk with humans, they are able to use implicit situational information, or context, to increase the conversational bandwidth*". Consequently, by rendering computers more context-aware we do not only enrich the human-to-machine dialogue, but we also help the realization of more useful computational services.

Abowd et al. note that a context-aware application aims to encapsulate information about the *who's*, *where's*, *when's*, and *what's* about an entity (whether this is a person, place, or a relevant object of the environment). In general, context can be categorized into *location*, *identity*, *activity*, and *time*. Dey [35] enumerates further context categories such as the user's *physical*, *social*, and *emotional* state.

Advances in sensing and inferring context have introduced a number of systems that make use of context. One example of such systems is the emerging class of personal assistants such as Amazon Alexa, Apple Siri, or Google Now. Based on a weather forecast fetched from the Internet, a digital assistant can remind one to take their umbrella before leaving from home, or adjust the home lighting considering the time of day and light conditions. Furthermore, by looking at one's online availability, a digital assistant can, for instance, schedule an appointment to the dentist [36]. The personal assistant presented above is just one of the many examples of pervasive and context-aware systems that are now part of our daily life. In fact, with the ubiquity of context-aware, mobile, and wearable systems we can now build systems that can augment many aspects of our life and activities, including human memory.

Pervasive computing has also changed the way how we interact with computer systems. Novel interaction techniques are now expanding the traditional approach, i.e., interaction with a desktop PC through a mouse and a keyboard used to be the norm, this is slowly fading away. One can now interact with their devices by *speaking* or *touching* them, as well as through *physical gestures*. As part of this research, we investigated the use of physical gestures for controlling how a system captures and shares data of one's daily activities.

## 2.2 Lifelogging

*“Lifelogging is a form of pervasive computing consisting of a unified digital record of the totality of an individual’s experiences, captured multimodally through digital sensors and stored permanently as a personal multimedia archive.”*

Dodge and Kitchin [37]

Lifelogging refers to the practice of indiscriminately recording the totality of one’s life experiences. This leads to the creation of comprehensive lifelog archives that document everything one has said, seen or heard, every action one has performed, every place one has visited, to name but a few. The realization of such concept is made possible thanks to achievements in the field pervasive computing. In fact, Dodge and Kitchin consider lifelogging a form of pervasive computing [37]. This concept is fueled by achievements in several technological strands [8, 37]: firstly, smaller, cheaper, and more autonomous digital sensors allow near-continuous recording of mundane activities; secondly, while capacity of digital storage is constantly growing, storage disks are becoming physically smaller, more power efficient, and cheaper, enabling almost infinite storage capabilities; thirdly, advances in data processing (i.e., big data) account for better interpretation and more efficient retrieval of stored information, thus increasing the utility of stored lifelogs

While the vision behind the Memex device could be considered a first concept of a lifelogging system [2], practical examples had to wait technology to catch-up. Early prototypes of lifelogging capture devices were developed by pioneers of this field such as Steve Mann and Gordon Bell. In 2005, Mann developed the EyeTap digital glasses that had a built-in camera to continuously record experience data and an integrated display to visualize such data [38, 39]. His approach was relived later by the appearing of commercial digital glasses such as Google Glass and Epson Moverio. On the other hand, the Microsoft SenseCam [40, 41], developed in 2006, was arguably the first “lifelogging camera”: a neck-worn front-facing device that captured images periodically (e.g., every 30 seconds) or when the scene changed significantly (such as when moving to a different room).

These early systems intended to primarily record images and short videos, however, the type of information that can be recored is almost limitless. The lifelogging concept can be applied to all kinds of data sources, including audio data, GPS location traces, fitness data, information on food intake, amount of

liquid consumption, physiological data such as heart rate, level of excitement or arousal, etc [12]. Of course, this also includes “digital traces” such as Internet browsing history, bookmarks, emails, calendar entries, social media activity, which serve as additional candidates for potential lifelogging data sources.

Lifelogging benefits a range of different applications and Gurrin [8] divides them into two categories: 1) *personal applications* where lifelogging is performed by single entities for personal benefits, and 2) *population-based applications* where lifelogs from several users are combined and processed together for some greater organizational or societal good. A vast majority of personal applications come from the quantified-self domain: the quantification of one’s life (e.g., calories burned, steps taken, sleep patterns, smoking habits, emails sent) in order to better manage one’s health, work, or private life [31, 42, 43]. On the other hand, examples from population-based lifelogging applications include the work from Hughes et al. [44] which aims to analyze large collections of visual lifelogs with the goal of informing market research. In this vein, authors propose a system which can measure audience exposure to advertising campaigns, using object recognition algorithms to detect the presence of specific brands and logos. Another example of this kind of lifelogging is the work from Byrne et al. [45]. Here, authors used SenseCam as a tool to collect observational data to better understand the information needs of clinicians in a hospital setting.

### 2.2.1 Lifelogs as a Surrogate Memory

Beyond applications from the quantified-self domain, prior research has shown that lifelogging can serve as human memory surrogate [46, 47, 41]. It can offer sufficient information about prior memories, thus supporting users in situations of everyday memory failures. The user could review captured data and encode from scratch information that otherwise has been missed from the experienced event, but also re-encode existing aspects that are fading away. As a result, this can support the recollection of those memories that had been completely inaccessible or partially accessible until that time.

From psychology research in human memory, it is well established that our episodic memories can be supported by reviewing external stimuli or cues [48], and that visual information (e.g., pictures, videos) account for the strongest such memory cues which can maximize the elicitation of prior memories [49]. Prior researchers have showcased such premise and have observed that SenseCam-like pictures and videos can support individuals diagnosed with specific memory problems (e.g., patients suffering from amnesia [41], limbic encephalitis [50], Alzheimer’s disease [51], and other episodic memory difficulties [52]).



For a detailed survey of studies that use SenseCam images to help people with memory impairments the reader is referred to the work by Harvey et al. [53]. On the other hand, little research attention has been devoted in understanding how lifelogging data might be used to address memory difficulties of healthy individuals in everyday settings. One such study involving 19 participants by Abigail et al. [46], offers strong evidence that SenseCam images provide efficient links to experiences from people's past. Their study not only shows that reviewing SenseCam images can help healthy individuals to truly remember and relive past experiences, but also to know and recognize what has occurred during a past event. Abigail et al. believe that the ability to recognize what has happened in an event from the past is due to their study participants utilizing schematic knowledge about familiar places, people in their life, or general knowledge about their daily routines. This means that lifelogging images should be able to assist people in finding their lost car keys or to recall the name of the person they have met the other day.

The availability of contextual information such as location traces, audio data, or even bio-physiological responses can increase the efficiency of visual memory cues. Kalnikaite and her colleagues found that location traces combined with visual cues offer further improvements in retrieving memories of past events [54]. Audio is yet another interesting data source candidate of such additional information. Niforatos et al. [29] investigated the possibility of augmenting memories of work meetings using a combination of both lifelog images and keywords that were extracted automatically from recorded audio conversations. From a study conducted over a period of five weeks involving 12 participants, they observed that such combination of data sources is capable of achieving memory improvements of up to 15% on average.

Another parallel avenue of research in lifelogging investigates how this practice can be made more efficient in order to improve one's memory recall. Work in this field has shown promising evidence that bio-physiological responses play a crucial role in identifying those moments that are of true significance to users. In fact, a number of studies from psychology and neuroscience have found that memory recall is correlated with arousal. Events for which a person manifests an increased emotional arousal are more likely to be recalled than other neutral events [55, 56]. Based on such encouraging results, Sas et al. [57] proposed AffectCam, a wearable system integrating SenseCam and a wristband measuring wearer's Galvanic Skin Response (GSR) in order to filter the most relevant pictures taken. Their initial results indicate that photos captured during high arousal moments support a 50% improvement in memory recall over low arousal photos. In this vein, Niforatos et al. [30] proposed the design of a similar

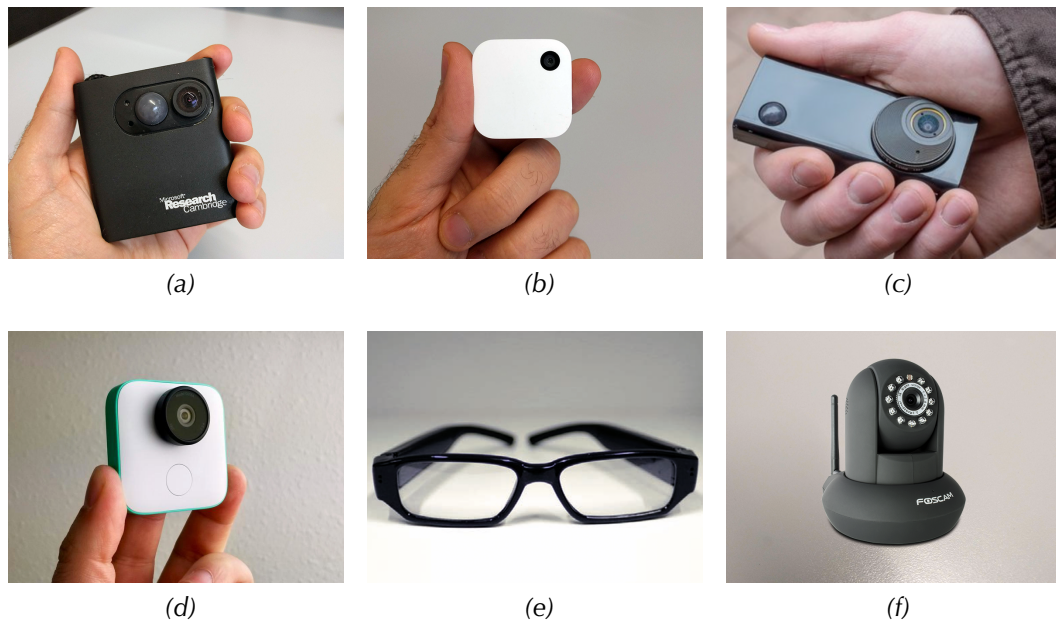


Figure 2.1. Commercial lifelogging devices: (a) SenseCam, (b) Narrative Clip, (c) Autographer, (d) Google Clips<sup>1</sup>, (e) Camera Glasses, and (f) Foscam<sup>2</sup>.

<sup>1</sup>Image by Jason Cipriani, ZDNET: <https://tinyurl.com/ybmrkzw9>

<sup>2</sup>Image source: <https://tinyurl.com/y54k4t38>

system but that relies on a user's heart rate as opposed to GSR, given that a user's heart rate can reflect almost instantaneous variations in their excitement levels. Their initial PulseCam prototype consists of an LG G watch R smartwatch that continuously measures the wearer's heart rate, and a Nexus S phone for picture taking, attached to the user's body using an armband.

## 2.2.2 Lifelogging Devices

In the following we will present an outline of commercial lifelogging devices. We will first look at visual recording devices, such as wearable cameras, camera glasses, and infrastructure cameras. We will then present devices for capturing audio data and other contextual information, such as smartwatches and bracelets.

### Visual Recording Devices

A typical lifelogging camera is small, compact and light, allowing it to be carried for long periods of time. Being small and compact can make the wearer forget about its presence and also forget that her actions are being recorded.

A lifelogging camera is commonly worn around the neck (e.g., on a lanyard), clipped on the users's body, or worn on the head (e.g., as glasses). It could also be mounted on an object in the environment (e.g., the TV or on a shelf). Regarding capture modality, most devices feature near continuous-capture mode, ranging from cameras that are recording all the time, to cameras that capture images on a fixed interval (e.g., every 30 seconds), to cameras that are triggered by sensors or user actions. In previous work [11] we reviewed the most common commercial lifelogging cameras, focusing on device form factors, picture quality and ethical and privacy issues stemming from the use of such devices. In the following we will briefly summarize our findings.

### Camera Hardware

Figure 2.1 shows a number of commercial lifelogging cameras that have become available in recent years. Modern lifelogging cameras have extended the functionality of the pioneering SenseCam (shown in Figure 2.1a) while offering an increasingly compact format. For example, the 2016 Narrative Clip 2 wearable camera (depicted in Figure 2.1b) weighed less than 20g (the SenseCam weighed several hundred), included WiFi connectivity (SenseCam needed a cable), featured 30h of battery life (the SenseCam lasted 12h), recorded 8MP images with  $3264 \times 2448$  pixels (SenseCam did  $640 \times 480$ ), and could record video (which the SenseCam did not). The Narrative Clip sensed additional data such as time and location. In fact, it offered partial access to location traces: to save battery, it only captured raw GPS signal strength, which needed to be uploaded to the company's website should one be interested in obtaining the actual location coordinates.

The Autographer (Figure 2.1c) was a similar camera with a slightly larger form factor than the Narrative Clip. It featured 5MP camera with a 136-degree wide angle, allowing it to capture more details of the wearers surrounding. However, this came with a cost: contrary to Narrative Clip, it produced distorted images (the Clip instead captured more "normal" looking pictures which were much more "shareable" with others). Unlike the Narrative Clip, the Autographer had a built-in GPS sensor providing direct support for GPS data. Both the Autographer and the Narrative Clip are no longer available or produced.

The recently announced Google Clips (shown in Figure 2.1d) is a hands-free camera with a similar size as the Narrative Clip. Unlike the Narrative Clip and the Autographer (which are both body worn cameras), the Google Clips is not intended to be clipped on the user's body. Instead, it should be mounted to a fixed object in the environment. However, the most notable difference with the other cameras is in the way how the Google Clips takes a pictures. Google Clips has

a built-in machine learning algorithm to recognize familiar people and to decide which moments are worth capturing. It delivers short video clips (without audio) of spontaneous moments that otherwise might have been hard to photograph. It features a high resolution camera lens with a 130-degree field of view. The battery allows for about 3 hours of smart capture.

Furthermore, a number of commercial camera glasses have become available in recent years. They are usually labeled as “spy camera glasses” and can be purchased for less than US\$50. They do not differ much from normal glasses, as shown in Figure 2.1e. Most camera glasses that we investigated can both take pictures and record high-definition video (including audio), which are then stored on internal storage, typically between 32–64GB, and downloaded from the device via USB.

Besides such mobile cameras, fixed infrastructure cameras can also be used to source experience data. In a previous data capturing experiment conducted in the context of the RECALL research project, we investigated one such camera, the Foscam IP camera shown in Figure 2.1f. The experiment’s objective was to produce our own lifelogging dataset by placing a group of researchers from the RECALL project (including the author of this thesis) in a heavily instrumented house [58]. The experiment ran for 2.75 days and we used a range of wearable and infrastructure camera sensors. The Foscam camera that we tested had a 0.3MP image sensor capable of taking pictures with  $640 \times 480$  pixels at a maximum rate of 15 frames per second. It featured a 300-degree horizontal pan and 120-degree vertical tilt to capture a great amount of environmental details. Thanks to its integrated night-vision capabilities it captured decent quality images even under poor light conditions. Unlike most wearable cameras, the Foscam also sourced audio data. Furthermore, it could wirelessly stream its video stream directly to any of the user’s devices. However, our prior experiment taught us an important lesson: contrary to mobile cameras, infrastructure cameras require a greater amount of time to install and calibrate (e.g., to ensure a good enough coverage of the environment). Furthermore, despite our efforts to properly setup the WiFi network, we still experienced network issues when streaming Foscam data. Consequently, we could only stream at a much lower rate than expected (4.04 frames per second instead of the advertised rate of 15fps).

### **Where to Position the Camera**

When using visual lifelogs as memory cues, the quality of the captured images is of great importance. It is essential that the camera captures high-resolution images, but also that it has a fast shutter speed in order to capture clear images

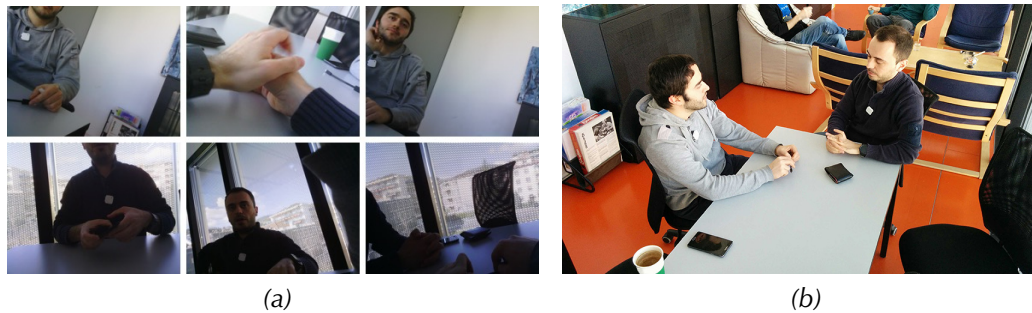


Figure 2.2. Comparison of experience capturing using (a) Narrative Clip devices clipped on users' chest, and (b) a fixed infrastructure camera.

while in motion. But perhaps the most important aspect regarding image quality is where the camera is positioned or worn (in case of wearable cameras): a choice that is usually driven by product design. For wearable cameras, different camera positions produce different characteristics. For example, a camera clipped to the chest (e.g., Narrative Clip) or worn around the neck (e.g., Sense-Cam) will very often produce partially or totally occluded images (e.g., by participants hands, arms, hair, or clothing, see Figure 2.2a). In our experiment [58], we observed that many shots from Narrative Clip cameras contained irrelevant background objects (e.g., ceiling, floor, walls). We furthermore observed an inevitable amount of duplication among such images. Being close to the user's eyes, cameras embedded in glasses deliver the most natural images. However, such devices usually produce unstable images due to users' frequent head movements.

Fixed infrastructure cameras on the other hand can offer radically different images from those of wearable cameras. The high vantage point of infrastructure cameras allows them to capture more comprehensive scenes compared to the low vantage scene of wearable cameras (see Figure 2.2b). During our experiment [58], we observed that, unlike images produced by wearable cameras, none of the images from the fixed cameras were blurred or occluded. However, as we previously argued, such cameras require higher infrastructure support.

Triggered by the capture constraints of wearable cameras identified above, in Chapter 4 we present our work of a system for exchanging captured images among peers that are experiencing the same event together. Wearable cameras of others with whom we are socially interacting with (e.g., having a chat) can often offer a more comprehensive view than our own camera. For instance, our chest-worn camera may never capture the person sitting next to us, while the camera of the person in front of us will. During such image exchange process,

the system will also seamlessly obtain any images captured by a nearby infrastructure camera, as long as we are in close proximity with it.

### Capturing Contextual Data

Beyond images, there is a number of additional data sources that can either assist in the selection of the best image-based cues, or which can be used as memory cues on their own. The spectrum of such additional data sources ranges from audio data, to location traces, to fitness-related data such as number of steps taken or stairs climbed, to physiological data such as heart rate, skin conductance, etc.

Among these, audio is probably the most interesting but at the same time the most contested such data source. There are a number of devices that can record audio information: starting from classical voice recorders such as dictaphones, to digital voice recorders, to smartphones. However, recording audio in most legislations is considered illegal unless all of the involved parties express their consent to do so [59]. Furthermore, in many circumstances this action can stir issues that can hamper the social acceptability of such practice. A crowdfunding effort raised in 2013<sup>1</sup> attempted to tackle this problem by recording short audio clips of only 60 seconds. Their wristband Kapture<sup>2</sup> constantly listened in the background but it only saved the last 60 seconds of audio when the user tapped it twice, allowing the wearer to retroactively record the most interesting parts of a discussion. It is not completely clear whether this kind of recording is legally compliant or not, but for one thing, the act of visibly tapping the device to trigger the recording may positively contribute to overcoming social issues around it.

Commercial products such as Fitbit<sup>3</sup> can capture different aspects of a user's daily activities including step counting, stair climbing, sleep tracking. Similar measurements can also be found in one of the the latest "consumer trends" in this field, i.e., the smartwatch. Furthermore, most smartwatch products are equipped with sensors to capture user bio-physiological responses. For instance, an optical sensor that touches the inner part of the user's wrist can estimate wearer's the heart rate. Other sensors such as the Empatica E4 wristband<sup>4</sup> can deliver the same measurement but with a clinical quality level. The E4 can furthermore measure Galvanic Skin Response (GSR), which can be used to derive levels of stress, arousal, and excitement.

---

<sup>1</sup><https://www.kickstarter.com/projects/1483824574/kapture-the-audio-recording-wristband>

<sup>2</sup><https://www.bizjournals.com/cincinnati/blog/2015/08/i-tried-it-kaptures-audio-recording-wristband.html>

<sup>3</sup><https://www.fitbit.com>

<sup>4</sup><https://www.empatica.com/research/e4>

### 2.2.3 Extracting Useful Information from Lifelogs

Human memory is driven by various triggers which are also known as memory cues. As we previously described in Section 1.1, memory cues can be images, sounds, location traces, smells, emotions, mood, etc. Therefore, extracting effective and useful memory cues from the sheer volume of captured lifelogs is an essential operation of memory augmentation systems. By drawing a parallel with how the human memory system functions, prior work [8, 60] highlights three major principles that need to be addressed in order to turn lifelogs into a surrogate memory: 1) segmenting lifelogs into smaller episodes, 2) annotating episodes with high-level information, and 3) retrieving memories. We will now look at state-of-the-art with respect to each of those principles.

#### From Lifelogs to Segmented Episodes

The continuous stream of memories of one's daily experiences are segmented into discrete semantic events, often referred to as *episodes* [61, 62]. Episodes form the basis for later recall of a particular event. Harvey et al. highlight [53] that sometimes episodes can be quite lengthy (for instance driving to work), but other times they can be quite short featuring a single moment (for example greeting a colleague in the hallway).

While it is not exactly clear how the mind creates such episodes, it seems that it depends on several factors [63, 64]. These include the type of event being recorded, physical changes in the environment (e.g., the movements of objects or people), personal goals, and prior knowledge about the structure of the event.

Analogous to such memory segmentation into episodes, lifelogs should also be segmented into semantic events for later retrieval. Event segmentation of visual data is not a new concept. In general, it is based on the analysis of the content. This has been applied in various scenarios: organizing photos in a semantically meaningful way based on the context and purpose of taking the photos [65], event detection in sports videos [66, 67], and automatically indexing surveillance data [68, 69]. Another early approach by Wang et al. [70] segments videos into shorter clips of fixed duration. However, this may not be that useful, since real-live events are not always of a fix length. Furthermore, Zhang et al. [71] propose a system that detects boundaries of events in video data. Their approach is based on a set of difference metrics, and boundary positions are estimated based on the amount of difference between successive video frames. All these approaches work reasonably well with video data, however, they cannot be directly applied to lifelogs. A lifelog data stream usually features a much lower frame rate, that goes even below one frame per second [53].

The approach by Doherty and Smeaton [72] can be probably considered as the most related and at the same time the most deployed lifelog event segmentation solution that was using SenseCam images. This solution incorporates the Heart's and Plaunt's Text Tiling algorithm [73]: instead of comparing a single pair of adjacent images for similarity, they do so with a block of 5 images. Furthermore, this system employs a sliding window approach across the complete image stream. The authors also incorporate data from other sensors available in the SenseCam, including accelerometer, temperature, and infrared sensor readings. The code of this work has been also been released as open source [74]. Irrespective of the wide adoption of this model, and de-facto being *the* baseline standard, Gurrin [8] expresses his concerns that Doherty's and Smeaton's segmentation model can be generalized to suit all (or even many) use cases. Harvey et al. [53] echo such concerns, suggesting that "these sophisticated methods may still not be sufficiently powerful as even a small number of errors may be problematic". They call for more sophisticated, dynamic and more powerful methods to analyze lifelog data.

### Annotating Segments

To make any memory augmentation system useful, it is necessary to tag the segments with higher-level concepts. Such tags can be useful in retrieving the right lifelog segment, and hence prompting the user to recall the corresponding memory episode. Prior research exploring query patterns of email search has shown that names of people can assist one in finding the email she is looking for [75]. For instance, while you may have difficulties in remembering when or where an event took place, but nevertheless you may remember who else was there present.

Manually tagging each single image (or even episode) can be a cumbersome and time-consuming process. For instance, wearing a Narrative Clip camera during the day and capturing two images every minute would result in about 1,500 images captured in only a single day. Clearly, this creates the need for automatic annotation techniques.

There has been a considerable interest in trying to build semi-automatic or even fully automatic image annotation solutions. For instance, Zhou et al. [76] have experimented with a web-based annotation system optimized for large data archives. Given a set of user defined labels, their MemLog system will then run a machine learning algorithm to propose the most likely labels for each derived segment. Advances in the field of machine learning allow for fully automatic image annotation solutions [77, 78]. There are even commercial image annotation



services that provide fully integrated image recognition and annotation APIs, such as the Google Vision API<sup>5</sup>, or Imagga API<sup>6</sup>.

Automatic annotation techniques have also been proposed for lifelogging images. For instance, Korayem et al. [79] investigate the feasibility of automatically detecting computer screens appearing in lifelog images. This work offers promising results showing that it is possible to reliably detect computer screens even from low quality, blurry and possibly occluded images captured by wearable cameras. Using a similar approach applied to lifelog images, Templeman et al. [80] developed a system that can reasonably well infer the location where an image is taken. Iwamura et al. [81] present a wearable system that, after recognizing the person in front of a user, shows videos of any previous encounters with the same person.

In yet other work, researchers have been able to infer high-level information from visual data captured by wearable cameras [82, 83, 84]. For instance, Castro et al. [83] propose a machine-learning approach to learn and predict everyday activities from lifelog images. Their technique achieves a high overall accuracy in predicting among 19 different activity classes (such as working, watching TV, reading, having a work meeting, cooking, eating, driving, etc.). Fathi et al. [84] propose an approach for recognizing social interactions such as discussion, dialogue and monologue from first-person perspective day-long videos. Their method relies on two kinds of data sources: detected faces and attention patterns. After detecting all faces appearing in one's video and then calculating their location and orientation, the system infers attention patterns such as who looks at who or whether all users look at a common place. This was shown to provide enough insights regarding the type of the social interaction (e.g., whether is is a monologue or a dialogue).

### Retrieving Memories

For any captured and annotated lifelog archive to be useful, it is essential to consider how users can explore and ultimately retrieve the very small elements that can trigger the recollection of a previous event from the whole archive [85]. This creates the necessity for an intuitive interface for both managing and cataloging lifelog archives. However, the development of any such interface can be particularly challenging. For instance, manually reviewing captured data can be very slow, and often impossible, due to the large volume of captured data.

---

<sup>5</sup><https://cloud.google.com/vision/>

<sup>6</sup><https://imagga.com>

Therefore, to facilitate efficient browsing of the collected data, many systems exploit temporal and spatial information – allowing a particular image to be found based on when and where it was taken [86, 87, 88]. It was found that when using an interface that incorporates time and location information participants needed significantly less amount of time to complete photo finding tasks [88, 89]. Furthermore, the interface received higher user satisfaction scores. In another study, Le and his colleagues investigate the possibilities for automatically summarizing large sets of image streams [90]. Their study insights echoe those benefits and lead to the following design guidelines that one can consider for the automatic creation of video summaries:

- Produced summaries should be brief and concise, and their duration should not exceed three minutes.
- Selected images should feature people, places, objects or actions. This improves users’ understanding of the social context.
- Summaries should preserve the chronological order of images, which provides additional support for the memory recollection process.

The interface developed by Lee et al. [91] can be considered as the first interface for tailored to lifelog archives captured by SenseCam cameras. It includes an algorithm for segmenting an entire day of lifelogging data into approximately 20 events. Events are then grouped into a morning, afternoon, evenings, or night cluster. Each such event is labeled with a uniqueness score and is represented by a “representative keyframe”. The higher the uniqueness score, the larger the keyframe of an event is. In a subsequent study using a lifelog collection captured over the course of 2.5 years, Doherty et al. [60] have found out that allowing users to refine the search strategy using “who”, “what”, “when”, and “where” queries reduces the average time to find a particular event to 127 seconds, as opposed to an average of 774 seconds when searching based on time and location information only. However, 127 seconds is a significant amount of time to find a particular event, and as authors stress “this still represents the single greatest challenge of our (i.e., lifelogging) community”.

All these techniques and interfaces of displaying the sheer volume of data require explicit input from the user. Browsing for about 2 minutes might be acceptable, very often also appreciated, if one wants to simply *retrieve* a specific piece of information such as a document or an email. As Sellen and Whittaker highlight in their constructive critique of lifelogging [85], the retrieval practice might require us to remember something about the item we want to retrieve,

for example when retrieving a document we might want to know when or where we wrote it, or remember any keywords of the document. But this does not involve remembering the complete experience about it. However, if the intended use of a lifelogging system is to assist the *recollection* (thinking in detail of past experiences) or *reminiscing* (recalling past experiences for emotional reasons) [85] than a pro-active data presentation approach may be more suitable. Instead of using such system as a search engine to find the index of a particular memory cue, the system would automatically deliver any relevant cues to a user in an unobtrusive fashion. As we explained previously in Section 1.1, this would still require that users specify high-level memorization goals to inform the system on what kind of memory cues they would like to see. But we envision that user input should be minimal in this case, and there should be less effort in specifying such goals compared to the practice of manually browsing for memory cues. The selected cues will then be delivered to users through ambient-fashion displays, e.g., presented as screen-saver images, displayed on a living room TV, or visualized in digital picture frames hanging on an apartment entrance hall. By regularly reviewing key memory cues one can train her memories and consequently may better recall prior experiences even when there is no support from a memory aiding tool.

The overview presented in this section shows that much of the lifelogging technology to develop human memory augmentation systems exists. Furthermore, we also highlight the necessity for novel presentation techniques tailored to the practices of recollection and reminiscing of past memories. But beyond such technical issues of how to best use such system and how to design visualization interfaces, the practice of lifelogging raises a number of privacy and security implications. In the following section we discuss what impacts might this have on users' privacy.

#### 2.2.4 Legal and Ethical Issues

The practice of lifelogging inevitably “looks” outwards and captures other people in the lifelogger’s vicinity including bystanders, family members, or friends. This may disclose to the lifelogger many aspects of their persona such as appearances, activities and whereabouts [8, 92]. Therefore, traditional research into this field has predominantly focused on privacy concerns of bystanders captured in camera footage. In this section we provide an overview of such problems, looking at both the legal and ethical aspects, prior to reviewing state-of-the art research solutions to those problems. We conclude this section with an overview of the privacy issues that lifelogging can also create for the lifeloggers themselves.

## Bystander Privacy

In many countries it is legal to take pictures and videos of identifiable people in public spaces for personal use. However, the legal landscape varies widely across countries. For instance, capturing others in public space in Italy and Sweden does not require consent as long as the pictures are used for personal consumption – for sharing with family or friends, but also for posting online for non-commercial purposes. Laws in other countries such as Spain or Japan consider this as an illegal activity, unless consent has been given before. Switzerland also prevents taking a picture of a person in a public space without the person’s consent, but contrary to Spain and Japan this activity is permitted if the captured person appears incidentally and has nothing to do with the purpose of the image – which is the case for lifelogging. A comprehensive overview of consent requirements worldwide for photography can be found at [https://commons.wikimedia.org/wiki/Commons:Country\\_specific\\_consent\\_requirements](https://commons.wikimedia.org/wiki/Commons:Country_specific_consent_requirements).

It is unlikely that a user can obtain usage and sharing permissions from everyone appearing on their lifelogs. Therefore, researchers have instead investigated other alternatives to protect the privacy of bystanders. For instance, Gurrin et al. [93] propose a privacy-aware lifelogging framework that utilizes a Google-street like technique to blur any identified faces appearing on captured pictures. Their solution embodies two design principles: *privacy by design* and *privacy by default*. Captured images are stored in their original form (i.e., unblurred), however, when visualizing stored data the system’s default policy prevents the display of any recognizable faces of others. Known contacts of the lifelogger can provide their explicit consent to appear on her data. This is achieved by providing a set of images featuring face models that can safely appear on the lifelogger’s images. This system has, however, certain limitations. Firstly, the face blurring technique is yet not completely reliable, and thus it might miss to “block” certain faces. Secondly, to support a retroactive access of images, images are stored unblurred. However, this may allow malicious others to break in and get unauthorized access to the unblurred images.

Other researchers have looked at techniques to prevent a camera from capturing privacy-sensitive situations in the first place. For example, Jung and Philippe propose Courteous Glass [94], the design of a wearable system combining an RGB camera sensor with a far-infrared (FIR) sensor. The infrared sensor is used to monitor the lifelogger’s social environment when the camera is not recording, and determines when it is acceptable to turn on the RGB sensor. On the other hand, when the camera is recording, its feed is piggybacked to a computer vision algorithm that detects when a new person enters the field of view or

when someone performs a pre-defined off-recording gesture, and turns off the RGB sensor accordingly. Schiff et al. [95] propose the Respectful Camera system, which blurs faces of people that wear specific markers such as orange vests. In yet another work, Truong et al. [96] investigate the possibility of preventing covert recordings by creating “capture-resistant” environments. Their system first detects the presence of mobile cameras in the environment (by tracking light-reflections produced by CCD and CMOS camera sensors), and then directs a localized pulsing light to each camera to distort its view.

All these solutions show that technology can help in overcoming legal barriers of visual lifelogging. The fact that something is legal, however, does not make it necessarily socially acceptable. The presence of a person with a wearable camera can easily create significant social frictions. A camera that is always-on is perceived as a threat to privacy, even if most lifeloggers do not try to hide their cameras, and sometimes even employ self-censorship to avoid situations of unethical recordings [97]. In fact, several reports show that lifeloggers have often come under social scrutiny. For instance, Steve Mann was assaulted in a restaurant in Paris for wearing his Digital Eye Glass [39, 98], while in another incident, a blogger was taunted for wearing Google Glass in a bar in San Francisco [99]. Because of such social backlash, Google Glass was banned in several places [100]. In an attempt to avoid any further situation of such kind, Google published a set of guidelines about the *do's* and *don'ts*, suggesting not to be a “Glasshole” and to respect the privacy of others when wearing the Glass in public [101]. Google Glass has been since discontinued; these incidents might suggest that Glass failed in part due to a strong culture of anti-surveillance [102].

Beyond such media attention, privacy issues with lifelogging devices were also investigated in research experiments. Denning et al. [103] investigated bystanders' privacy perspectives regarding a co-located peer wearing augmented reality (AR) glasses with a built-in camera sensor. After interviewing 31 bystanders in 8 different cafés in Seattle, the authors found out that participants were “predominantly split between having indifferent and negative reactions to the device”. Furthermore, participants expressed an interest in being asked for recording permissions and in having at hand an easy-to-use mechanism for blocking recording devices. In yet other studies [104, 105], researchers observed somewhat higher levels of acceptability for SenseCam-like wearable cameras, possibly because they are worn around the neck and look less obtrusive than glass-like cameras.

Gurrin et al. [8, 93] believe that increasing bystanders' awareness of wearable cameras, and allowing bystanders to have their say on whether they would like to be recorded or not, can do much to ameliorate social frictions regarding

covert recordings. Koelle et al. [106] echo such observations, and explore design strategies for privacy notices for wearable cameras that announce themselves and their actions to bystanders. Throughout two co-design sessions involving both design and UX experts, they delineated eight low-fidelity artifacts for status indicators that go beyond having a simple LED light on the camera. The proposed camera artifacts embody different concepts including (1) physical occlusion of camera lenses, (2) indicating area of capture, (3) visualizing device actions and usage intentions, (4) displaying the camera image (or a derived abstraction of it), and (5) transfer control over the image to the bystander. For instance, the idea of physically covering the lenses would prevent image capturing even if the camera's software would still be recording. This was rated by experts as highly secure and trustworthy, as it can reassure the bystander what it is impossible to be recorded through. Displaying to bystanders a preview of what the camera is capturing was another explored design strategy. While this strategy obtained high initial ratings regarding understandability of camera status and application purpose, analysis revealed some potential issues with it. Experts were worried that displaying the camera's preview in remote locations such as the wearer's chest may break the "connection" of the camera and its image – a connection that is otherwise obvious when operating with conventional digital cameras.

Results from this study show that all of the investigated design strategies can significantly improve *noticeability* and *understandability* of body-worn cameras. However, the authors' analysis also revealed that the lack of *security* and *trustworthiness* still remains an open issue, which can hinder the social acceptability of lifelogging devices. In all but the physical occlusion strategy, experts were reluctant to trust the camera's operation – i.e., whether its software is executing what otherwise would be communicated through the camera's feedback mechanism. For example, despite that the camera's preview display could be paused, there is no guarantee that its software is not recording. This resonates with Knowles's observation that "trustworthy data is data that isn't simply accurate but is *verifiably accurate*" [107]. In this context, the trust landscape can be shifted from trust in lifeloggers to trust in their cameras, seeing the concept of trust as a verifiable technical property of a camera. Trust in systems can be established by means of hardware-based solutions for secure firmware attestation. This would still require the design of interactive protocols that would allow a bystander to verify that a camera is not really recording. Obviously any such protocol needs to be efficient and practical in order to be used every time one encounters others with cameras.

### Privacy of Lifeloggers

Beyond privacy concerns of third parties appearing in someone's we also find questions related to the privacy of the lifeloggers themselves. Unlike traditional cameras, a lifelogging device is always on and automatically capturing images. This can ultimately alter a lifelogger's perception about the privacy of captured images. From our own experiment on lifelogging [58] (as reported in section 2.2.2 above), we observed that many images captured through this practice included information that subjects consider private, which otherwise is very unlikely to have been captured by users when using a legacy camera. Specifically this included images of the lifelogger's own laptop or smartphone screen (showing contents of a private email), or images captured in private locations such as bathrooms or bedrooms. Even though study administrators took care to remind participants not to capture in private spaces by placing "do-not-capture" signs at the entrance of such private spaces, very often lifeloggers were forgetting to stop capturing in these areas.

The capture of such sensitive information would pose less issues if this kind of data was never meant to be shared with others or uploaded to cloud services (one would still need to protect their own repository from the prying eyes of a hacker). By drawing a parallel with the already established practice of sharing photos online [108], any privacy issues stemming from sharing lifelogging data would seem irrelevant at first. However, sharing online is predominantly done manually and deliberately by the user. Instead, lifelogging photos will be often shared seamlessly and automatically in order to capture a greater amount of details (as we described in section 2.2.2 above).

In this vein, Templeman et al. [109] have shown how such opportunistically collected images pose new risks to users privacy and physical security. In their work, the authors conceptualize a new attack vector where image data is used to construct rich, three-dimensional models of a person's environment, hence enabling a "virtual theft". In another work, Hoyle et al. [105] seek to understand privacy attitudes and perceptions of lifeloggers and shed some light on how will users manage the capturing and sharing of their lifelogs. In their study, Hoyle et al. asked their study participants (N=36) to wear a lifelogging camera and capture their mundane activities for a week. At the end of the study, participants were asked for a subset of the captured images about why they would or would not share them. The results show that most participants expressed their willingness to share captured images, but factors such as the presence of certain objects, location, appearance of others, would most likely affect sharing decisions.

To better understand what makes a lifelog photo sensitive, Hoyle et al. [110] conducted a follow-up study. Five researchers annotated and analyzed images captured during the first study and combined their results with the reasons participants gave for sharing or not sharing those images. Their analysis confirms that computer screens pose a major concern for users. Beyond screens, many images had text clearly visible, disclosing various private information including credit card numbers, academic transcripts, exam answer sheets, etc. Impression management was found to be yet another reason that affected the sharing decisions of many participants. Authors observed that lifeloggers avoided sharing images that showed a negative trait (e.g., smoking a cigarette or drinking alcohol). On the other hand, photos of positive traits (e.g., showing one working or studying) were deemed as shareable in order to promote positive impressions.

That said, there are several approaches that can assist users in curating lifelogging pictures prior to sharing them. In principle these approaches can be grouped into three categories: (1) manual post review control, (2) in-situ control and (3) automatic control. Clearly, given the amount of captured images in a day, any manual and after-the-fact approach will be cumbersome and labour intensive for the user. Furthermore, this may result in mistakenly disclosing sensitive information or “misclosures” [111].

Automated approaches on the other hand can mitigate user’s burden and offer practical solutions in controlling the sharing of lifelogging images. Initial work in this domain has investigated computer vision algorithms to scan for sensitive places [80], particular objects [112], computer screens [79], or high-level activities [113]. In one such work, Jana et al. [114] present a privacy-protection system that transforms visual data using computer vision algorithms before making it available to other applications. However, the executed transformations produce image data that can only be understood by system applications and are not viewable by humans. This clearly defeats the purpose of augmenting one’s memory through visual cues. In this vein, Thomaz et al. [82] seek to assess and quantify the balance between privacy-sensitive information appearing in lifelogging images versus salient information that can assist users in achieving a particular task. Using their proposed framework, which they refer to as *privacy-saliency matrix*, the authors evaluate the performance of four automated techniques for protecting user’s privacy: face detection, image cropping, location filtering, and motion filtering. Their analysis was conducted using a total of 14’422 images collected by five participants over a course of three days. Their results show that all of the tested techniques performed poorly, suggesting that more work is needed for creating automatic solutions which can efficiently balance both privacy and utility of lifelogging images. This resonates with Adams’ [115] observation that



privacy issues related to captured experiences often rely on users' implicit assumptions of its usage and the intended receiver, and as such they can vary with person and context [58]. For instance, an image that can infringe a user's privacy because it contains a computer screen can be *the* strongest memory cue. Therefore, one way to improve the efficiency of automatic privacy-mitigating mechanisms is to couple them with context-dependent user supplied privacy policies. In Chapter 6 we provide a comprehensive review of state-of-the art approaches on this topic.

In situ or momentary control mechanisms can provide an efficient and immediate solution to the challenge of curating lifelogs. As an event is being experienced, a user would control different aspects of how the experience is captured and how such data is shared with others. For instance, the user could momentarily pause the photo collection, indicate what moments of the experience can be shared and with whom such sharing can take place, or delete the last captured moments in case they were captured by mistake. This allows users to reason about the capturing and sharing decisions using the experience context while such context is still fresh in their mind. It also alleviates the burden of later trying to manually find and remove problematic images without having a clear representation of the context while such data was captured. A similar finding was observed in the study of Hoyle and his colleagues [105]. Their study participants used in situ pause and delete controls as a primary mechanism for managing their privacy throughout photo collection study tasks.

In principle, such in situ controls can be embedded in the camera itself, can run on a smartphone, or even be implemented in a dedicated physical device. However, using the camera itself as a control interface might not be the most optimal solution. Performing any interaction with the camera may well interfere with the photo collection process and hence can miss capturing important moments. For instance, one may need to unclip the camera for a moment and interact with its touch screen interface in order to change the camera's settings regarding the capture and sharing process. Even if the camera would be operated by gestures, the user can block the sensor with their hand while performing the gesture. Instead, using a separate control interface can better decouple these two processes and offer a better user experience. In Chapter 5 we investigate how a dedicated device can be designed, how it can be used to control the practice of capturing and sharing of lifelog traces, and how does such an interface compares to a smartphone application designed for the same purpose.

## 2.3 Human Memory

Human memory has been widely studied in the fields of Neuroscience and Psychology. This has resulted in the development of many models that describe what human memory is and how it functions. In this section we will briefly present memory and its workings, focusing on the most common and widespread models.

In a seminal work from 1968 [116], Atkinson and Shiffrin describe a memory model that was a turning point in further developments and consolidation of human memory theories. Their model consists of three structural components:

1. *Sensory memory* is constructed by what we see, hear, taste, or feel throughout everyday experiences. As it is impractical (and useless) to remember each and every detail of such moments, the brain will typically only retain this information for less than half a second (but sometimes even up to 3 seconds) [117] – which is long enough to allow us to remember the most relevant details about our surroundings. This information is processed in a very quick fashion, following a pre-attentive process where changes are detected because they violate predictions of a pre-built neural representation of the environment [118, 119]. Research has found that only visual stimuli is processed this way [116], allowing us to make sense of, e.g., the shape and color of objects that we see around us.
2. *Short-term memory* is often regarded as one's *working memory*. At any given time it can hold between 5 and 9 items [120]. Its capacity can be increased by first clustering similar information into bigger chunks, and then storing only the chunks [121]. Information in this store is kept for a brief period of time, i.e., between 15 and 30 seconds. However, rehearsed items can be instead retained there for several hours [120]. A characteristic of short-term memory is that it does not necessarily store sensory information in its original form. For instance, a word that was presented visually will be registered by visual sensory input, however, the same information may well be encoded in a short-term memory store as auditory information.
3. *Long-term memory* is the last component of the Atkinson's and Shiffrin's model. Information stored in the two preceding components will eventually decay, whereas information in the long-term store is believed to remain there for a lifetime. Experimental results show that we can recall names and faces of classmates with an accuracy of 90% even 15 years after graduation [122]. This number declined by 60% when tested 48 years after graduation. Transfer of information to the long-term store happens while

such information is still available in the short-term store. Prior research has found out that information on topics that we already have some knowledge about is more more likely to be copied to this memory store. This is mainly due to the fact that such information can be easily connected to other related information that is already in the long-term store. There is also a reverse flow of information, i.e., from the long-term store to the short-term store, which is manifested when we usually think about something [116].

Long-term memory is divided into *implicit (or procedural) memory* and *explicit (or declarative) memory* [123]. Implicit memories relate to skills and the unconscious act of “remembering” how to do things (e.g., knowing how to drive a car or how to open a door). Explicit memory refers to knowledge of facts and events that can be remembered later on. Tulving proposes a further distinction of explicit memory into *semantic* and *episodic* memory [124]:

1. *Semantic memory* is the memory of facts and knowledge and is independent of the context when it was acquired. Tulving describes it as the precondition of using a language, and further notes that semantic memory ... “*is a mental thesaurus, organized knowledge a person possesses about words and other verbal symbols, their meaning and referents, about relations among them, and about rules, formulas, and algorithms for the manipulation of these symbols, concepts, and relations.*”. The act of retrieving a semantic memory leaves its contents intact in the memory store. As a result, this kind of memory is more robust to modifications than episodic memory is.
2. *Episodic memory* holds information about one’s personal experiences, which are structured into smaller chunks of episodes or events. In addition to information about individual episodes, this memory store also contains contextual information such as the temporal-spatial relationship between these episodes [125]. Information retrieved from this store is also used as a special input back into the store itself. Thus, unlike the case with semantic memories, retrieval of episodic memories can modify the memory system.

From the moment a memory is conceived, to the moment it can be retrieved, the memory goes through five stages namely *encoding*, *consolidation*, *storage*, *recall* and *forgetting* [126]. Two of these processes are most relevant to our work:

1. *Recall*: refers to the mental process of retrieving a particular event from the past. To make this happen, our memory predominantly relies on contextual information that was captured during the memory encoding (i.e., creation)

phase but also on contextual information that is available at the time of retrieval – a process based on the theory of *encoding specificity* [63]. This contextual information acts as a stimulus and triggers the recollection of associated memories. Such information can be constructed implicitly (e.g., hearing a song playing in a bar) or explicitly (e.g., browsing a photo album). In principle, anything can act as a memory trigger, including photographs, music, smell, location, etc. These stimuli is what is known in the field of Psychology as *memory cues*, and the process of replaying and reviewing such memory cues is called *cued recall*. Evidence has shown that cued recall can help both subjects with memory impairments [127] and healthy individuals [46] in improving their recall of episodic memories.

2. *Forgetting*: Contrary to recall, forgetting is the inability to partially or completely retrieve memories related to past episodes from the long-term memory store. This can result in very unpleasant and embarrassing situations. For instance you may forget where you left your car keys, or you may struggle to remember the name of a person that you met some time ago. Commonly, forgetting is considered as a negative feature of our memory process. However, Bannan highlights that “forgetting is a feature and not a bug” [128]. The ability to forget is equally important as our ability to remember: it helps us to soften information overload, to unburden from negative experiences [129, 130], but also to reconcile storage for acquiring new knowledge [131, 132]. There are many factors that influence the act of forgetting. This can happen due to the lack of necessary memory cues, a mismatch between cues and encoded information, or simply because of somewhat infrequent access to particular memories. Schacter [133] has studied the act of forgetting and he delineated seven different causes why we fail to recall: 1) information gets less accessible over time, 2) lack of attention during the encoding of memories, 3) memory blocking, 4) misattribution of memory context, 5) tendency to include information from others, 6) bias from preexisting memories and 7) inability to forget unpleasant episodes. One interesting observation that arises from reasons mentioned in points 4 and 5 is that our memory system can become vulnerable to false episodic memories. This would allow one to implant in our heads fabricated memories of events that never occurred or that occurred differently compared to how we would recall them after the implant. There is significant evidence to support the notion that such threats are real, which we describe in the following section.

### 2.3.1 Human Memory Manipulation

Apart from the challenge of keeping our memories safe from the prying eyes of others, the threat of memory manipulation might be the most worrisome aspect of memory augmentation systems: if an attacker is able to remove, add, or change our captured information, the resulting memory cues may implant memories in our heads that never took place, or, in turn, accelerate the loss of other moments by ensuring that no memory cue will ever remind us of them.

Human memory manipulation has been subject to extensive experimental research in psychology, with studies repeatedly demonstrating that human memory is easily manipulated. In a recent set of studies, Shaw [7] shows how our memories can be manipulated to make us pleasantly believe “that we had tea with Prince Charles” or worse “that we committed crimes that never happened”. Shaw explains how, through a social psychology process of six steps, carried out in a lab-study over the course of few weeks with ordinary university students (no hypnosis or torture involved), she could implant what she calls “full false memories”<sup>7</sup> to 70% of the study participants.

In another experiment, Morgan et al. [134] show how misinformation can have an effect on the memory of a recently experienced and stressful event. By introducing misinformation through a mugshot photo to military personnel, participating in a mock prisoner-of-war camp, they could decrease the accuracy of them remembering the interrogator of one of the (mock) prison sessions. Moreover, they could trigger participants to believe and hence report the presence of items that didn’t exist in the interrogation room such as glasses or a telephone.

Many sources of misinformation have the potential to modify and manipulate our memories of an event [7], e.g., viewing a set of photos, discussing with others, reading news articles or reading what others are “tweeting” for an event. In this thesis we particularly focus on *photographs*, since they are easy to capture and make for rich memory cues (see Section 2.2.2). Several studies have demonstrated the role of photographs “overshadowing” our memories by creating other competing ones. Henkel et al. [135] showed that even generic photos (i.e., from a stock catalog) showing a particular task have the potential to trick participants into thinking that they performed such a task (while they did not). Brown and Marsh [136] were able to use photos of different places to manipulate participants’ autobiographical experiences, making them believe that they had visited them (while they have not). Other studies have experimented with

---

<sup>7</sup>Shaw defines full false memories of an event as memories in which (among others) 1) the number of reported details by a participant is no less than ten and 2) participants explicitly saying that they believe that the event really happened.

lifelog-like photos (both unmodified and modified) of previous real events (e.g., family vacation or birthday celebration). In one such study, Wade et al. [137] were able to make participants recall details of a previous hot air balloon ride experience (which never happened). To achieve this, they created a fake image by photoshopping the participant in a balloon air ride. In another study, Lindsay et al. [138] could incite participants to reminisce about a previous school-related event (which they did not attend) by showing them pictures of the event obtained from their class-mates and claiming participants had taken them.

The aforementioned studies on human memory manipulation highlight the fact that our memories can be manipulated with fake image data. The security implications this gives raise to are significant. Imagine that an attacker compromises our memory augmentation system and gets access to our complete digital lifelog. By modifying the stored data or controlling the memory cue selection, an adversary can attenuate some of our memories while reinforcing others, or even fabricate memories of events that we never experienced. Data that we obtain from others is yet another source of memory manipulation attack. We do not know whether such lifelogs represent a true reflection of what really happened. Our “trusted friends” in this case can be the evil adversaries or maybe their systems have been compromised in the first place by a different attacker. The peculiarity of these sort of attacks is that the misinformation data need no longer to be generic but can be sourced from the lifelog of each individual, potentially making the attack more effective.

In Chapter 4 we present a systematic and practical solution for addressing the threat of memory manipulations by ensuring the integrity and provenance of digital memories. Our proposed solution is comprised of a secure and trusted wearable camera, a storage protocol to link captured images in a secure chain, and a zero-knowledge protocol for verifying shared but modified images obtained from others.

## 2.4 The Envisioned Threat Model

The threat model that we consider in this work shares many aspects with threat models of other applications in the field of lifelogging. Nevertheless, it differs in at least four aspects with other lifelogging threat models.

Firstly, many lifelogging applications traditionally consider only the data confidentiality aspect with the goal of protecting the privacy of the lifelogger (a person capturing data for herself) when sharing captured data with others. If the lifelogging application also captures visual data then such works extend their

threat models to also account for privacy issues of bystanders (others that can appear in one's lifelogs). While data confidentiality is very important, in this thesis we focus on the aspects of data integrity and provenance. We show that lack of integrity and provenance opens the door to another less obvious but equally important problem as privacy, that is, the risk of human memory manipulation.

Secondly, when sharing data with others, we focus on threats and challenges for both the sharer and the recipient of such data. Sharing sensitive data can risk sharer's privacy, while receiving modified data can endanger the recipient's memory, since such modified data can generate falsified memory cues.

Thirdly, many works from the field of lifelogging do not consider scenarios where the lifelogging gear is compromised. This would permit the attacker to get a copy of the recorded data but also to modify data on the fly directly in the device itself. In this thesis, we account for such attacks with the goal of ensuring integrity and provenance of experience data from the moment such data is captured.

Lastly, we provide a comprehensive threat model by identifying security challenges for each of the three stages of memory augmentation, from experience capture, to data storage and sharing, to data processing and presentation. However, note that our solutions address only threats related to the capture, storage and sharing practices. Threats and attacks related to the other stages of cue processing and presentation are presented as future work (see Section 7.2).





## **Part II**

# **Manipulation Resistant Memory Capture and Sharing**



## Chapter 3

# Secure Memory Capture and Storage

An all-embracing memory augmentation system can bring tangible benefits both for us as individuals but also for our society. For one thing, through the act of remembering we are able to “learn from history” and advance knowledge. However, as Bannon [128] points out, any such technology that aids us in *remembering* should also support us in *forgetting*. Mayer-Schönberger [129], in his book on the virtue of forgetting in the digital age, observes that our transition to the information society has shifted a very old norm about memories: “while forgetting used to be the norm and remembering the exception, today forgetting has become the exception, and remembering the default”. The point that these authors make is clear: any memory augmentation system has to consider the *duality of our memory*, i.e., “the role of remembering and the importance of forgetting”.

In fact, there exists a way how we can decrease our ability to recall certain memories. In the field of Psychology, this process is known as “retrieval induced forgetting” (RIF) and, most importantly, it is based on the cued-recall process that we previously introduced in Section 1.1. The basic idea behind RIF is as follows: by selectively reviewing memory cues of a category, one can significantly increase the recall of memories associated with those cues, while at the same time decrease the recall of memories from the same category but for which no cues were presented [5]. In a recent study by Cinel et al. [6], a series of six experiments provides strong evidence that RIF can also be carried out using images captured by lifelogging cameras.

These findings suggest that we can address Bannon’s critical view on the duality of memory, and that we can use the same technology to willingly fade out some memories from our past. While this is a completely legitimate thing to ask from our memory augmentation system, however, how do we make sure that it is *only us* demanding our system to do so? As we pointed out in section 1.2,

by controlling what cues we review, an attacker can manipulate our memories of past experiences.

Therefore, in this and the next chapter we set out with the goal of addressing the challenge of preventing human memory manipulation in pervasive memory augmentation systems. At the outset, we explore the different ways how such attacks can be executed in practice. We then design and build a systematic and practical solution that addresses the elicited threats. Thus, we propose:

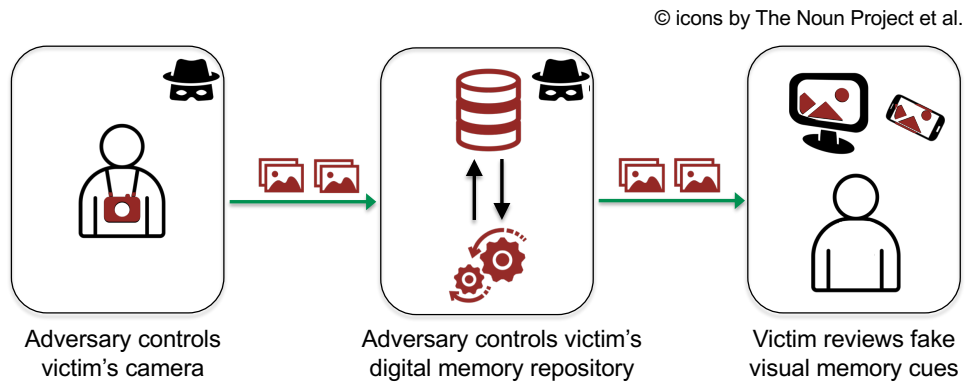
- a secure and trusted wearable camera coupled with a data storage protocol to capture and store digital memories in a secure fashion (presented in this chapter);
- an approach to securely exchange memory cues among co-located users, coupled with a zero-knowledge protocol for verifying shared but modified memory cues (Chapter 4).

By using off-the-shelf cryptographic primitives, our protocols can efficiently run on low-power wearable cameras and thus can protect memories from the moment they are captured by a user's device to the time that they are stored in repositories for later processing (e.g., into memory cues). We assess the protocols' security and demonstrate their practical feasibility using an implementation based on a prototype memory-capture camera. In this chapter we target the following research question:

- **RQ1:** How can we guarantee digital memory integrity and provenance to prevent memory manipulation attacks?

*Parts of this chapter are based on the following publication:*

- **A. Bexheti**, M. Langheinrich, I. Elhart, and N. Davies, "Securely Storing and Sharing Memory Cues in Memory Augmentation Systems: A Practical Approach," in *Proceedings of the 17th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'19)*, 2019, p. 10



*Figure 3.1.* Threat model for captured experience data that constitute users' memories. Victim's items colored in red, namely, the wearable camera and the memory repository can be controlled by various adversaries and represent potential points of attack. Attackers can change our memories by inserting fictitious experiences, modifying existing experiences, or deleting recorded experiences.

### 3.1 Threat Model and Requirements

As digital memory cues are elevated to become an essential source of our recollection of past events, we expose our overall memories to significant manipulation threats.

Without the ability to ensure the security of digital memory cues, we would endanger our memories in a number of ways. Figure 3.1 depicts the envisioned threats intertwined with the memory augmentation process. We look at each possible threat individually and highlight the envisioned adversaries, before deriving a set of security requirements towards the creation of secure pervasive memory augmentation systems. In this vein, we see the following threats:

#### **T1: Capturing fake experience data.**

Attackers can try to distort our memories as early as the experience capture phase. They can compromise our cameras and install malicious software which would permit them to modify captured images. This may sound more far-fetched than it actually is. Today's smartphones are powerful enough to, for instance, detect faces and make them younger, change their gender, or even replace them with different faces, all in real-time<sup>1 2</sup>.

<sup>1</sup><https://www.insider.com/iphone-and-android-apps-photoshop-your-pictures-instagram-2017-7>

<sup>2</sup><https://www.abc.net.au/news/2017-04-27/should-you-worry-about-privacy-when-using-faceapp/8476666>

**T2: Repository manipulation.**

An attacker that has compromised our memory repositories can (1) **inject fake memory cues**, (2) **modify existing ones**, or (3) **delete a carefully picked set of memory cues**. They will be able to implant memories of events that we never experienced using attack (1) or (2), or accelerate the vanishing of existing memories (i.e., induce RIF) using either attack (2) or (3). Moreover, by influencing the memory cue creation process they may also induce RIF, which is outside the scope of this thesis.

These attacks can be carried on by (i) any malicious person that is able to hack into our wearable cameras, (ii) the repository service provider, or (iii) any third-party that can compromise a memory repository.

**Security Requirements**

Based on the elicited threats, we delineate a set of requirements that should be considered in designing manipulation-resistant memory augmentation systems:

**R1: Ensure verifiable memory cue integrity and provenance.**

Captured experience data should feature reliable and verifiable provenance data, i.e., information on the origin and context of capture, as well as information about their integrity, i.e., what changes (if any) have been made to them. This ensures that the captured data is an accurate reflection of what occurred during that experience.

**R2: Ensure equipment integrity.**

Any such cue provenance and integrity information is accurate and valid as long as the devices producing this information are trusted and are not compromised. Thus, we need guarantees that experience capture sensors are running a valid and a certified software stack.

**R3: Secure personal repositories.**

Digital memory cues should be stored in secure and user-controlled repositories. However, not everyone of us will want (or will be able to afford) to host their own memory repository, but will instead subscribe to third-party services able to host our captured memories (for instance just like how we subscribe to an email service or commercial cloud server and not host our own serves). Thus, we need to make sure that the integrity and provenance of our memory cues is intact even while they are kept in repositories.

## 3.2 Related Work

The work presented in this chapter intersects two principal research strands: capture sensors based on trusted computing and protocols for securely linking data.

### Capture Sensors Based on Trusted Computing

Knowles [107] explores the implications of *trust* in emerging data-rich systems, with the goal of understanding how users generate and sustain trust in these systems. In her view, trust is a holistic concept composed of three interconnected aspects: trust in data, trust in systems, and trust in people. Findings from this work are in line with the design requirements that we draw as part of secure memory augmentation systems. The trust in data highlights the importance of data accuracy when collecting user-related content. However, as Knowles points out, trustworthy data is not simply accurate, but is *verifiably* accurate. This is in line with our design requirements R1 (i.e., accounting for verifiable integrity and provenance of memory cues), and R2 (i.e., ensuring equipment integrity).

A second observation of the work from Knowles is that trust should be propagated from the data-level to the system-level. System trust results from the level of security of each individual component of the system (if the system sits within a larger system-of-systems architecture), and how the system combines and interprets the collected data. This observation resonates with our design requirements R3 (i.e., continuing to keep data secure even on memory repositories), and R4 (i.e., providing transparency for the memory cue selection process).

Prior research has proposed using trusted computing capabilities for camera sensors. In an attempt to restore credibility to digital photography, Friedman proposes the design of his Trustworthy Digital Camera [139], which relies on public-key cryptography to digitally sign captured images. The private key is stored in a “secure microprocessor”, but the author does not provide any additional information about such a microprocessor. Winkler and Rinner [140, 141] take this concept further, with TrustCAM, a prototype of a smart embedded camera equipped with a trusted platform module (TPM). All images captured by TrustCAM are digitally signed and encrypted to guarantee image authenticity and integrity. In later work, Winkler et al. [142] propose TrustEYE.M4, which further improves the security of their first TrustCAM sensor. Improvements come from the fact that they move security features as close as possible to the sensor, thus reducing the number of trusted components in their camera design. Our proposed secure camera goes beyond TrustCAM’s capabilities: it securely links captured images in a chain structure in order to prevent unnoticed image deletion

from user repositories. However, we acknowledge that our camera can benefit from the improved security design of TrustEYE.M4 in an attempt to prevent any hardware-based attacks.

Saroiu and Wolman [143] propose two design alternatives for trusted sensor devices. One design is similar to our trusted camera, i.e., embedding TPM-like capabilities directly into the sensor. Their second approach relies on a virtualized environment coupled with a TPM. The virtual machine will be in charge to read data coming from each individual sensor and then use the TPM to sign such readings before making them accessible to others. This design based on virtual machine can be further explored with the ultimate goal of improving the functionality and security of our camera. Furthermore, the authors show that trusted computing can be used in mobile sensing applications beyond trustworthy picture taking. They highlight a number of different application scenarios that can benefit from the deployment of trusted sensors, such as location proofs, participatory sensing, vehicular sensing networks, or energy consumption monitoring. Findings from this work testify for the security and efficiency of a TPM-enabled wearable camera, making for a stronger case for using an approach based on trusted sensors for the the protection of experience data.

## Protocols for Securely Linking Captured Data

One way to detect data unsolicited deletion attacks is to securely link data elements together in a data-chain structure. Traditionally, such schemes are realized using cryptographic *hash chains* [144, 145] where the hash of an item  $i$  is calculated using the item itself and the keyed hash of the previous item  $i-1$ . Any subsequent data deletion or modification attempt would invalidate the structure. Prior research has employed this concept for creating a data authentication protocol suitable for low-power devices [146, 147] to prevent fake data injection in broadcast networks.

Other works [148, 149, 150] have used hash-chain structures to securely link elements together with strong guarantees of their temporal order. They make use of trusted authorities to produce signed timestamp tokens that also depend on tokens issued for previous items. Once an item is timestamped and added to the chain it is impossible to modify its token or remove the item itself from the chain, even by the item owner or the timestamping authority.

While such immutability is a desired property in some contexts, it is a limitation for our envisioned application. We instead propose a lightweight protocol for securely linking captured data that allows *authorized* users to legitimately remove a particular item from their memory repository without invalidating the



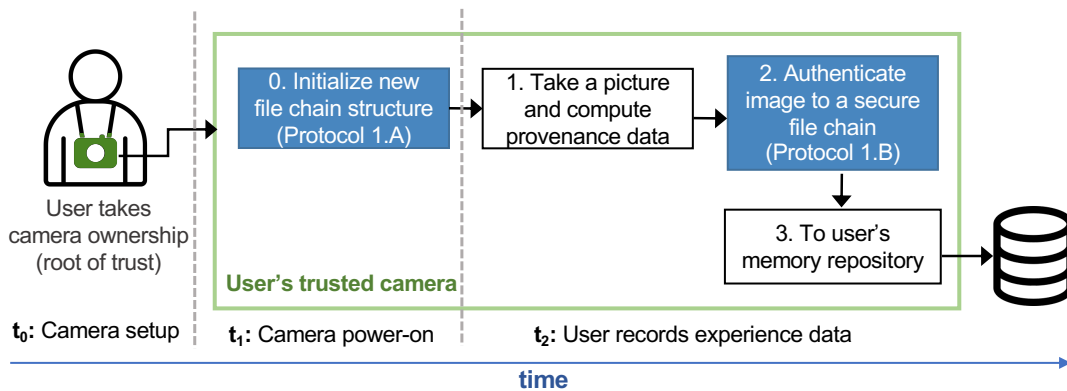


Figure 3.2. Overview of the event flow of using our system for securely capturing and storing experience data.

whole chain structure. A user can delete accidental capture of, for instance, a bathroom break or a museum visit where photography is forbidden. However, an attacker cannot do so undetected, since any instance of *unauthorized* data deletion will invalidate the structure and thus will be detected.

### 3.3 Secure Memory Capture Using Trusted Sensors

The root of trust in our system is a secure and trusted camera. Trusted cameras are a well-known concept in security [140, 142, 143]. In the course of this research, we have built a prototype wearable camera that uses a Trusted Platform Module (TPM)<sup>3</sup>. The purpose of the TPM is threefold. Firstly, it enables the camera to guarantee the integrity and provenance of all captured images, and secondly, it ensures that the camera’s firmware is tamper-resistant and intact. Furthermore, the TPM, described in this chapter, lays the groundwork to securely bootstrap the subsequent data storage protocol as well as the data sharing protocol explained in the next chapter.

Figure 3.2 depicts the event flow of using our system for securely capturing and storing experience data. At the outset, a user will take “ownership” of a camera (i.e., issue the `TPM_TakeOwnership` command, which triggers the TPM to create its set of root keys). For each captured image  $I$  of a user’s experience, the camera will then log “provenance” data  $P$ , such as time and location of capture, and a fingerprint (cryptographic hash) of the captured image. It will then cryptographically “seal” this provenance information by digitally signing a hashed

<sup>3</sup>[https://trustedcomputinggroup.org/resources/tpm\\_main\\_specification](https://trustedcomputinggroup.org/resources/tpm_main_specification)

representation of  $P$  using its root private key. The combined raw provenance data and its signature are called the *provenance certificate*  $\Pi_I = \{P, \text{Sign}_{PK_{priv}}(H(P))\}$ . Later on, when reviewing an experience, the camera owner can verify that those images have indeed been captured by their trusted camera (addressing threat T1) and that they have not been maliciously modified in the meantime. Any attempt to add external images into users' memory repositories threat (T2.1) or modify existing images (T2.2) can be easily detected since the suspicious image will lack a valid signature. Furthermore, using a data storage protocol described in section 3.4, they can also check if any images have been deliberately deleted (threat T2.3), e.g., to reduce the memories of a particular experience.

The rest of this section describes the camera's implementation, followed by a description of the underlying trust bootstrapping process [151].

### 3.3.1 Camera Implementation

We built a prototype camera (depicted in Figure 3.3) to evaluate the feasibility of the proposed protocols. Our camera is powered by the “Raspberry Pi 3 Model B+”IoT platform to which we added a CryptoShield<sup>4</sup> with an “Atmel Trusted Platform Module”<sup>5</sup>. For picture taking we use a serial camera module<sup>6</sup> that provides pre-compressed JPEG images with a maximum resolution of  $640 \times 480$  pixels. We implemented all of our protocols in Java. Our implementation makes use of jTSS [152], a Java library that implements the software stack proposed by the Trusted Computing Group (TCG)<sup>7</sup> for managing the communication with the TPM. Ultimately, our solution may be ported to the smaller “Raspberry Pi Zero W” in order to make it more portable and manageable for real-life applications. The Pi Zero W board measures  $6.5\text{cm} \times 3.0\text{cm} \times 0.5\text{cm}$ , making it about 2.5 times smaller than the Pi 3 B+. While the slower Pi Zero will run our solution less efficiently, our results will most likely be matched by the performance of any next-generation Pi Zero boards.

### 3.3.2 Trusting the Camera

To establish trustworthiness in our system, we rely on a hardware-based approach for secure platform attestation (i.e., firmware integrity) and secure storage of key material (i.e., private keys are not disclosed to unauthorized parties).

<sup>4</sup><http://learn.sparkfun.com/tutorials/crypto-shield-hookup-guide>

<sup>5</sup><https://www.microchip.com/wwwproducts/en/AT97SC3204>

<sup>6</sup><https://www.adafruit.com/product/1386>

<sup>7</sup><https://trustedcomputinggroup.org>

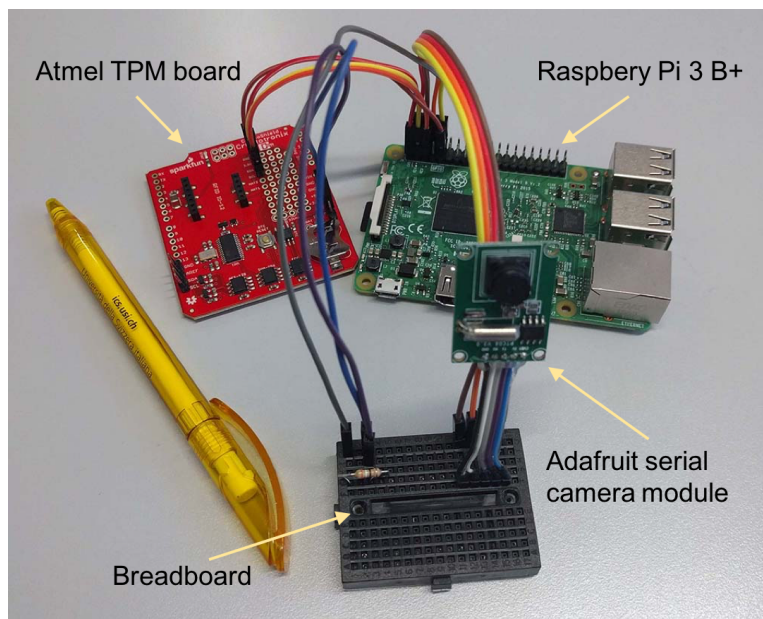


Figure 3.3. Our prototypical camera setup used for performance evaluation.

With the help of the included TPM module, our camera can establish an implicit chain of trust. In this context, the camera proves to its owner – but also to other users, in case images are being shared with others – that it is in a *known-good* state. Typically, the state of a system is a function of both the hardware and the software it has executed. As Parno et al. [151] suggest, the state of the hardware can be verified by a signed certificate by the system’s manufacturer. However, software’s state is more dynamic and ephemeral, and needs a different attestation procedures. In this vein, traditional attestation schemes start from a trusted component (Root of Trust for Measurement) to perform system measurements as the system’s software is being loaded. A common such measurement is to compute a hash over the software’s binary, libraries, configuration files, and any inputs used [151]. The computed measurements are then stored inside the TPM, specifically in a set of dedicated registers also known as Platform Component Registers (PCR). A PCR has a special security property such that it cannot be freely overwritten, but can only be *extended* by hashing the concatenation of data that is already stored in it with a hash of the new data.

To verify what state a system has booted in (i.e., knowing what software component is being executed on a platform), a verifier engages in an interactive challenge-response protocol with the system’s TPM (Root of Trust for Reporting) [151]. The verifier will send a random nonce  $n$  and request from the TPM

to generate a *quote*. The quote contains the verifier's nonce and the measurement aggregates that are stored in the PCRs within the system's TPM. The TPM will then respond with such a quote packaged in an attestation report  $R_{Att}$ , signed using its private attestation key  $AK_{priv}$  (how such key is generated is explained in the next section 3.3.3), i.e.,  $R_{Att} \leftarrow \{n', PCR', \text{Sign}_{AK_{priv}}(n', PCR')\}$ . The verifier will recognize the system as trustworthy if: 1) the signature can be verified with the public part of the TPM's attestation key  $AK_{pub}$ , 2) the received nonce is the same as the one that the verifier provided (i.e.,  $n' = n$ ), and 3) the measurements contained in the PCRs are in an expected good state (i.e.,  $PCR' = PCR$ ). This scheme allows a user to remotely verify the current state of a device at arbitrary times.

However, in our application, it is crucial for a user to know that a camera was in a known-good state at the moment when it captured a picture. Moreover, using the above scheme to engage in a platform attestation protocol each time a new image is captured might introduce significant overhead. As a result, we modify the traditional interactive platform attestation protocol to a *non-interactive* one as follows. Instead of waiting for a verifier to initiate the protocol, the camera automatically provides an attestation report (obtained from its TPM) every time it captures a picture. In order to ensure that the report is fresh (and not one from a previous time) the camera uses a *fingerprint* of the currently captured image as a nonce parameter when requesting a report from the TPM. With a view towards supporting a secure and trustworthy sharing of images with other co-located peers, the fingerprint is computed right after the image is captured and immediately shared by means of a short-range wireless broadcast with co-located others, using a protocol that we designed for such purpose (the protocol is described in detail in Chapter 4). Finally, the produced platform attestation report for an image is embedded in the image's provenance certificate  $\Pi_I$ .

To check the reported platform state of a camera that captured an image  $I$ , in addition to verifying the reports signature and its PCRs values, a verifier should also check the report's nonce for freshness. If the image in question was captured by the verifier's own camera, they can then compute a new fingerprint nonce  $n'$  from the associated image  $I$ , and match it with the nonce of the report (i.e.,  $n' = n$ ). When receiving a modified image from a co-located peer, one instead uses the fingerprint that was collected wirelessly during co-location and compare it with the report's nonce.

### 3.3.3 Provisioning and Protecting Camera Key Material

The TPM provides means for ensuring the confidentiality and integrity of its key material. Every TPM module is provisioned with an *Endorsement Key-pair EK*, which is stored in tamper-resistant non-volatile memory inside the TPM. The *EK* is embedded in the TPM as part of its manufacturing, a process which is assumed to be carried out in a secure and trusted environment. During the first-time activation of the camera and its TPM, the *EK* is used to generate the *Attestation Report Signing Key AK* and *Storage Root Key SRK*. The *SRK* is another key which is also stored inside the TPM. We follow the specifications from the Trusted Computing Group (TCG) and use *SRK* to derive other application-specific keys. This also happens during the activation of the camera. Specifically, we use *SRK* to derive the *Provenance Signing Key PK* and all other keys that are mentioned later in this work. Such derived keys are stored in the camera's storage (outside the TPM) but encrypted with the private parent key (i.e., *SRK*). To control access to these keys, we employ a PCR state constraint, i.e., the TPM will refuse to use *EK* and *SRK* – and hence any key derived from them – if the platform is not running in a trusted state. We explained the platform attestation procedure based on PCR measurements in section 3.3.2 above.

## 3.4 A Storage Protocol for Securely Linking Data

Our secure camera prevents an adversary from injecting or manipulating images in a user's repository (i.e., threats T2.1 and T2.2). Each image requires a valid signature that is unique to the user's camera, something an attacker should be unable to produce. The TPM equally prevents an attacker from hacking the camera's firmware in order to modify images directly on the camera, and before even uploading them to the user's repository (i.e., threat T1). However, threat T2.3 – image deletion – can not be prevented this way: an attacker who gains unauthorized access to a user's repository could easily remove important images. Given the large number of captured images and the fact that such data is usually reviewed a long time after it was captured, it is challenging for a user to determine if any particular image is missing. While we may not be able to *prevent* someone who already has unauthorized access from performing any deletion, we want to be able to *detect* such instances, i.e., we want to know if an image in a series of captured images was removed, and whether it was done by the users themselves or by an unauthorized party.

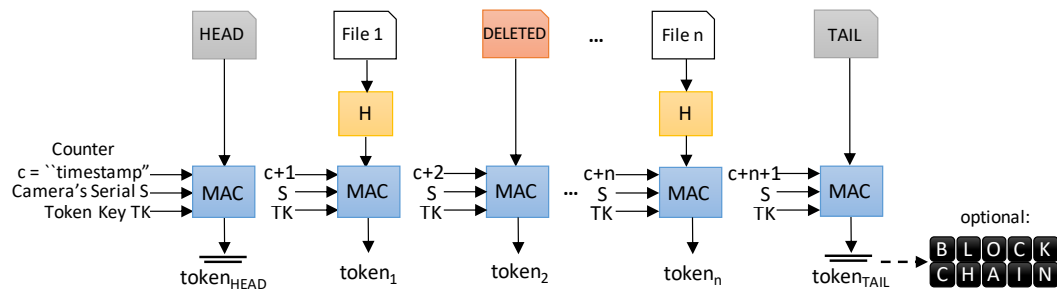


Figure 3.4. Overview of the file chain scheme. Each file is linked by a unique counter. The order is then authenticated using a MAC token computed over the file’s content and its counter.

### 3.4.1 Protocol Description

We propose an efficient scheme to securely link images, captured by users’ cameras, in their memory repositories. Figure 3.5 describes the underlying steps, Figure 3.4 provides an overview of the scheme, while Table 3.1 itemizes the used identifiers. In our scheme, data is ordered and linked using an incremental counter “*c*”, represented as a hexadecimal number. Two special (empty) files, “*HEAD*” and “*TAIL*”, mark the beginning and end of the list, respectively. Removing an element from within the linked list will require either a replacement file to be inserted, or all subsequent elements to be renumbered. Adding a file to the end requires updating the *TAIL* file. Both operations will require secret key material (as described further below) that is only available to authorized users.

To first get a feeling of how large the counter value can get, consider the following estimation. Constantly recording one’s life in pictures (with a frequency of 2 photos per minute) will result in 2,880 pictures produced in a day, 1,051,200 pictures in a year and about 73.5 million pictures during a lifetime of 70 years. In order to ensure that, after a camera reset, the list never overlaps a prior one, we always initialize the counter with a current timestamp (e.g., Unix epoch, seconds since 1970), which is currently at  $\approx 1.5$  billion.

A hexadecimal counter value of 8 characters ( $16^8 \approx 4$  billion) would thus be more than sufficient, with another 8-character serial number to differentiate between cameras. The counter can be part of the file’s header (e.g., EXIF tag in case of JPEG images) or simply be part of the filename. In case of the latter, a counter of length 16 should easily fit within the maximum length of 255 characters that most standard file systems support, such as Microsoft’s NTFS, Apple’s HFS, or Linux’s EXT.

Variable	Description
$\Pi_I$	Signed provenance certificate for image $I$ .
$EK_{pub/priv}$	TPM endorsement key.
$SRK_{pub/priv}$	TPM storage root key derived from $EK$ .
$PK_{pub/priv}$	Key for signing/verifying $\Pi_I$ .
<b>Storage Protocol</b>	
$TK$	Symmetric key for computing authentication tokens.
$PTK_{pub/priv}$	Derived from $SRK$ , used to encrypt $TK$ .
$S$	Camera's unique serial number.
$c$	Counter initialized to a current Unix time.

Table 3.1. List of variables used for the storage protocol.

In order to prevent an adversary from replacing or renaming files, any operation on them must be authenticated by an authorized user. For this we rely on the established cryptographic primitive of a message authentication code (MAC) [153], which allows one to authenticate a message  $M$  by computing an authentication token using a secret key  $TK$ . The token can be used later to verify the authenticity of the initial message.

We thus build our linked file structure as follows (also see Protocol 1 in Figure 3.5). At power-on, the camera initializes a counter  $c$  (step 1) and creates two empty anchor files for the *HEAD* and the *TAIL*. The anchor files are assigned with a counter value of  $c$  and  $c + 1$ , respectively, and with the camera's unique serial number  $S$ , by simply writing these information to the filename (steps 2, 5). This operation is then authenticated by computing a MAC token over their filenames (steps 3, 6).

For each newly captured image  $I$ , the camera proceeds as follows. At the outset, the next counter value  $c + 1$  (step 7) and the camera's serial number  $S$  are assigned to the image by writing them to the image's filename (step 8). This operation is authenticated by first hashing the image's content<sup>8</sup> and then concatenating it with  $c$  and  $S$  before computing the MAC token over this (step 9). The produced tokens are embedded in the image's provenance certificate  $\Pi_I$  (the provisioning of these certificates was explained previously in section 3.3) which in turn is stored in a separate auxiliary file as explained in step 9 of Protocol 1.

<sup>8</sup>We use a one-way collision resistant hash function (such as SHA3-256).

---

**Protocol 1** Securely Linking Captured Data
 

---

**A. Initialize a secure file chain (on camera power-on)**

1.  $c = UNIX\_timestamp$  //initialize counter
2.  $newFile("", f_{HEAD} = "HEAD" || S || c)$   
//create *HEAD* empty anchor file with filename  $f_{HEAD}$
3.  $token_{HEAD} = MAC(TK, f_{HEAD});$   
 $newFile(token_{HEAD}, f_{CERT\_H} = f_{HEAD} || ".cert")$  //authenticate *HEAD* and  
//store its token in a separate certificate file with filename  $f_{CERT\_H}$
4.  $c = c + 1$  //increment counter
5.  $newFile("", f_{TAIL} = "TAIL" || S || c)$   
//create *TAIL* empty anchor file with filename  $f_{TAIL}$
6.  $token_{TAIL} = MAC(TK, f_{TAIL});$   
 $newFile(token_{TAIL}, f_{CERT\_T} = f_{TAIL} || ".cert")$   
//authenticate *TAIL* and store its token in a separate certificate file  
//with filename  $f_{CERT\_T}$

**B. Add an element to the secure chain**

for each newly captured image  $I$

7.  $c = c + 1$  //increment counter
  8.  $fileRename(I, f_I = "IMAGE" || S || c || ".jpeg")$   
//add  $c$  and  $S$  to the image's  $I$  filename
  9.  $token_I = MAC(TK, H(I) || S || c)$   
 $newFile(token_I, f_{\Pi_I} = f_I || ".cert")$   
//authenticate image and store its token in a separate  
//provenance certificate file with filename  $f_{\Pi_I}$
  10. Update *TAIL* by executing steps 4-6
  11.  $commitToBlockchain(f_{TAIL}, f_{CERT\_T})$   
//upload *TAIL* and store its token certificate file to  
//an immutable blockchain ledger (optional)
- 

Figure 3.5. Pseudocode of the storage protocol.

Computing the token as a function of the contents of the file and its counter value binds these two together, meaning that an adversary cannot delete a data file and then overwrite the counters of the subsequent files without being noticed. Finally, the structure's tail is updated (step 10).



In case that a legitimate user wants to delete a particular image  $I_D$ , and without renumbering all subsequent files from the structure, she can simply use an empty replacement file. After removing both the image file and the corresponding auxiliary certificate file that contains the MAC token, the user creates an empty file and assigns it the same counter as that of the deleted file, by writing it in the filename:  $f_{I_D} = [“DELETED”||S||c]$ . She authenticates this by computing a MAC token in a similar way as for the empty anchor files:  $token_{I_D} = MAC(TK, f_{I_D})$ , and again stores it in a separate file with the same name:  $f_{CERT\_DELETED} = [f_{I_D}||“cert”]$ . If the user wants to hide this deletion from a third party, she alternatively can recompute the entire list by replacing all counter values of subsequent images in the list and recomputing the corresponding authentication files.

### 3.4.2 Checking for Missing Images

To check a stream of images for potentially unauthorized deletions, one proceeds as follows. For each image  $I$  that was captured between time  $t_i$  until time  $t_j$  (e.g., all images from the last work meeting):

1. read the filename and corresponding MAC token;
2. using the obtained counter from step 1, re-compute a new MAC token and compare it with the token obtained from step 1;
3. 3) check if this file is linked properly with the subsequent file in the given range by verifying that the counter of the previous and subsequent files equal  $c - 1$  and  $c + 1$ , respectively, and that all serial numbers match the serial number of the desired camera. Furthermore, re-compute new MAC tokens for both the previous and subsequent files and compare if the computed tokens match with the tokens stored in the filenames.

If, for every image in the given range, the computed token matches the token associated to it, one can conclude that the tested data link is valid and intact. A broken link or an unauthenticated one is an indication of a deliberate data deletion attack.

### 3.4.3 Generating the MAC token key TK

$TK$  is randomly generated by the TPM during the camera’s first-time activation. It is then encrypted with an asymmetric parent key  $PTK$ , which in turn is derived from the TPM’s storage-root-key  $SRK$ . Both  $TK$  and  $PTK$  are securely kept

inside the camera's storage encrypted with each other's parent keys, i.e., *PTK* and *SRK*, respectively.

Checking for missing images requires knowledge of the key material *TK*, which is used to verify MAC tokens. Since this process will be performed outside the camera, (e.g., on a user's personal computer) *TK* has to be shared with other potentially untrusted computer devices. Managing the storage of *TK* is outside the scope of this work. However, given a computer with a TPM, a secure key-migration protocol as specified by TCG [154] allows for the secure transfer of *TK* from the camera to another device.

### 3.4.4 Security Analysis

An adversary that wants to delete an image file without the victim noticing has three options:

1. renumber all subsequent files and then create new updated MAC tokens for each of them;
2. create a replacement file ([*“DELETED”*||*S*||*c*]) and compute a valid MAC token for it; or
3. reuse the MAC token of another file that the victim deleted herself (similarly as in 2, but in this case the replacement file *“DELETED”* would have been created by the victim herself).

The first two options are prevented by virtue of the secret key *TK* used to compute MAC tokens. Reusing tokens of other files is not going to help due to the mismatching serial number *S* and counter *c*.

An attacker with unauthorized access to the victim's repository can, however, overwrite the whole directory with a previously made backup. In practice, the attacker would “roll back the time” to a much earlier, but valid state, thus making the last *n* images disappear. To prevent this, the MAC token of the current TAIL can be occasionally committed to an immutable public ledger, i.e., a blockchain (Protocol 1, step 11). Since no image is uploaded to a blockchain (but only tails' MAC tokens are), it is still possible to perform authorized image deletions as explained before (i.e., using an empty replacement file). However, if the user wants to hide the fact that she deleted an image herself, after recomputing the entire list following the image deletion (as before), the user also needs to invalidate the previously committed token of the tail from the blockchain and commit the token of the new tail.

As we described above, the security of the proposed scheme depends on the secrecy of the symmetric key  $TK$ , which is used to compute MAC tokens. Unauthorized possession of the secret key would make the system vulnerable to the all threats presented in this chapter, as it would allow an attacker to re-compute a valid MAC token for every injection or deletion operation. To retrieve they key an attacker can try the following advanced attacks:

1. chosen-plaintext attack with the final goal of recovering  $TK$ ; or
2. obtain  $TK$  through so called timing attacks.

By definition, one of the key security requirements of a MAC algorithm is its ability to resist to chosen-plaintext attacks. This means that even if an attacker can use one's camera as an oracle and request it to compute a MAC token for different plaintext images, the attacker cannot get any information about the secret key or guess a valid token for an image that was not sent to the oracle. In our work we use a particular type of a MAC which involves a hash function (also known as HMAC). In this context, Bellare [155] proves that an HMAC is secure and is a pseudorandom function family (PRF) if the underlying compression function is also a PRF. According to Bellare, a HMAC is still collision-resistant even when implemented with a hash function that is not resistant to a second pre-image attack. One has still to use a long-enough key in order to prevent brute-force attacks.

However, this does not prevent an attacker who can get physical access to one's camera (e.g., stealing it for a short time and returning it before anyone can notice) to use it to capture and hence sign images that the attacker would chose on purpose. This type of attack can be thwarted by having the camera authenticate the user prior to being activated. This can be achieved using a fingerprint scanner and encrypting the fingerprint information using a TPM-issued key.

A timing attack is another vector for disclosing the secret key. Such an attack relies on information gained by observing the behavior of a MAC verification oracle. Similarly as in the chosen-plaintext attack, an adversary can submit an image and MAC token, and the oracle will tell if the token is valid or not. A typical pitfall here is that the system would take slightly longer to respond for a valid token than for an invalid one. An attacker that can detect the difference can then try to construct the correct secret key. This kind of attack can be prevented by using a proper and a valid implementation of a MAC function.

### 3.4.5 Evaluation

We measured both runtime overhead and energy consumptions of the proposed camera in order to validate the camera's practical feasibility. In this chapter we focus only on runtime overhead figures of the camera's operations including Protocol 1. In Section 4.7.1 of the next chapter we additionally report on the runtime overhead of our second protocol for securely exchanging images with co-located others. Furthermore, in Section 4.7.2 we report on the total power consumption of the camera when running both protocols.

Latency tests were conducted using two different images sizes: (1) a low-resolution image with  $640 \times 480$  pixels (the maximum resolution of the installed camera module) and (2) a high-resolution image with  $4096 \times 3072$  pixels (a resolution offered by most of today's wearable cameras). Images are interpreted using an 8-bit RGB color model, where each pixel is represented by three bytes. We benchmarked three processes:

1. capturing and storing a picture in the camera's internal storage,
2. obtaining a TPM-signed platform attestation report, and
3. appending an image to a secure link structure (following the Protocol 1 process described in Figure 3.5).

Tests were conducted with a TPM complying to version 1.2 of the standard. All cryptographic operations were carried out with RSA keys of 2048 bits, while all hash operations were computed using SHA3-256. Digital signatures were calculated over a SHA1 hash representation of the data to be signed.<sup>9</sup>

Figure 3.6.a summarizes the camera runtime results. Our proposed protocol works reasonably well on low-power devices. A low-resolution image is captured and processed in less than 13 seconds (3.5 s for taking the photo, 1.7 s for generating a platform attestation and 7 s for executing Protocol 1). Less than 45 seconds are needed for a high resolution image (35 s for taking the photo, 1.7 s for the platform attestation and 7 s for Protocol 1). The tested camera module did not support such high resolution capture, however, we estimated that it would take about 35 seconds to download a pre-compressed JPEG image of that size with the module's maximum supported transfer-rate of 115,200 bps. The runtime overhead for both generating a platform attestation and appending an image to a secure data structure is almost constant irrespective of the image size.

---

<sup>9</sup>TPM provides support only for signing SHA1 hashes.

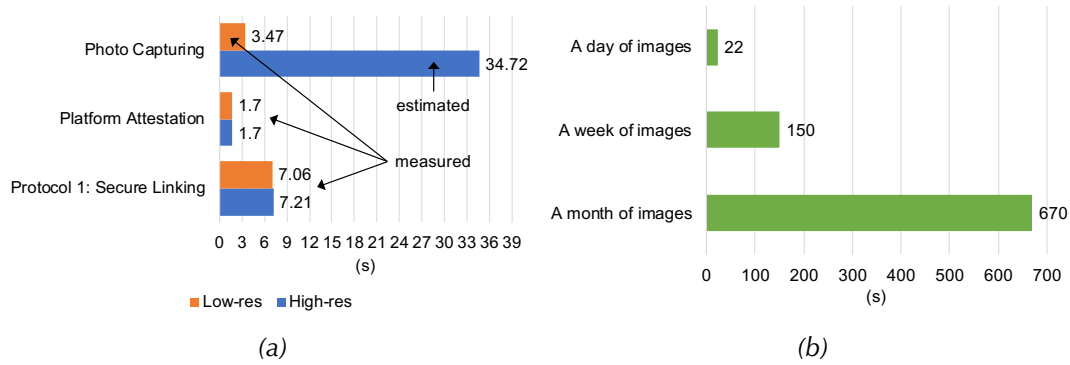


Figure 3.6. **Left:** Implementation runtime overhead (in s). Measurements were conducted a with low-resolution images ( $640 \times 480$  pixels) and high-resolution images ( $4096 \times 3072$  pixels). **Right:** Image storage verification times (in s) using: one day worth of images, one week of images and one month of images.

In additional tests, we benchmarked the verification times of our system. We ran the benchmarks using high-resolution images (with  $4096 \times 3072$  pixels) on a machine with a 2.3 GHz quad-core processor. Specifically, we evaluated only the storage protocol for securely linking captured images (Protocol 1) since the runtime of the other parts, i.e., verifying image authenticity and verifying the platform attestation is negligible (i.e.,  $< 3$  seconds overall for the longest image chain that we tested) compared to the link protocol. We ran the verification procedure from section 3.4.2 and observed how long it took to verify file-chains of different lengths (see Figure 3.6.b). We could verify a stream of 2,880 images (one day worth of images captured in 30 second intervals) in about 22 seconds; 20,000 images (approx. one week of images) in about 150 seconds; and 86,000 images (approx. one month) in about 670 seconds.

### 3.5 Limitations

Using a TPM-based camera as a root of trust has its own security caveats, as discussed in [156]. With physical access to the camera, an attacker may still modify its picture-taking sensor, thus feeding the camera’s software with already modified image data (i.e., threat T1). Possible remedies include modifying the sensor to provide encrypted image data to the rest of the camera components (proposed also by [156]), or authenticating the sensor using pattern noise data that may serve as a unique identifier [157].

All this would still be unable to prevent *staging attacks*. Consider the following examples of such situations:

- **Recording in a Malicious Way:** When obtaining images shared from other malicious users, one cannot say whether they were facing their cameras in a particular way in order to hide some information that otherwise would have been visible in the captured images. Moreover, such malicious users may try to highlight some other information in order to reinforce a particular memory.
- **Re-enact a Fictitious Experience:** A group of malicious users could enact a fake version of an experience, more or less recording a “movie” with actors (which could include, e.g., a look-alike actor to fake the presence of the victim in the recordings). Then, they could share those memories with a victim user and claim that those are the memories of a real experience.

The proposed camera is addressing the risk of memory manipulation by ensuring the integrity and provenance of captured images that will potentially be reviewed by users. However, in no way it intends to provide the *ultimate proof* whether an image that a user may review is a fake memory cue or not. In practice, this issue cannot be solved with only a technical solution, it needs much more than that. Thus, even though one can *verify* the integrity and provenance of images captured from a trusted camera, in the above examples this may not be enough to ascertain that the images one reviews are not part of a targeted memory manipulation attack.

The above examples are not that easy to be successfully realized in practice. For one thing, they would produce *conflicting evidence*: the difference between the shared image stream and the personal one, captured from the victim’s camera, could be enough to alert the victim for suspicious data; or during the time that the fictitious experience was happening, the victim user could remember being to a different location. Nevertheless, such examples show what the proposed camera *cannot* directly protect against. Instead, in this thesis we target a use case that can more likely happen in practice: deliberately modifying captured experience memory cues.

## 3.6 Chapter Summary

Memory augmentation systems allow us to improve the recollection of episodic memories, and hence bring obvious benefits. However, fueled by their ability

to also attenuate our memories, any such technology also entails great risks, as it makes our memories vulnerable to targeted manipulation attacks. In this chapter we investigated how such attacks can happen in practice, and consequently proposed a systematic approach to address some of those attacks.

An attacker can compromise our system in several ways. They can hack our capturing gear, for instance by means of malware injection through a (regular) firmware update, and by extension cause the camera to produce modified images. Such intentionally modified images would then be considered as valid representation of past situations. Furthermore, an attacker can break into our digital memory repositories, which yields several attack vectors. An attacker can thus change our memories by inserting fictitious experience data, modifying existing ones, or silently deleting recorded experiences.

In this chapter we presented a secure wearable camera based on a trusted computing platform (TPM) which addresses these attacks (**RQ1**). Unlike other similar trusted cameras which ensure the credibility and integrity of captured photographs, our solution additionally addresses two of the most important memory manipulation attacks.

Firstly, the camera runs a custom protocol which joins captured images in a secure data structure, thus thwarting surreptitious data deletion attempts from compromised memory repositories. We showed that the proposed scheme can efficiently run in low-power embedded cameras: a high-resolution image ( $4096 \times 3072$  pixels) is captured and added to a secure chain in less than 45 seconds. We further showed that verifying the integrity of a chain of images captured in a day can be done in about 22 seconds, or about 670 seconds for verifying longer chains of approximately one month worth of images.

Secondly, the proposed camera allows the seamless exchanging of captured images with other co-located users. The benefits of sharing memories and the description of such secure sharing is the topic of the next chapter, which will present an approach that enables secure image sharing, and hence, prevents malicious others from manipulating our memories by sharing fabricated images.





# Chapter 4

## Secure Memory Sharing

In Chapter 3 we presented an approach to securely capture and store experience data, which can be later used to construct cues for reinforcing one’s own memories. In this chapter we go beyond and investigate the possibility of securely sharing captured experiences with others.

It is perhaps not surprising that the principle objective of a memory augmentation system is to support and augment one’s own memories. However, driven by today’s “social networking culture” where users continuously generate and share information with others, O’Hara et al. [158] argue that lifelogging is *not* only capturing data about oneself for one’s own purposes. Seen this way, captured memory cues could be shared with others in a same way as we already share an abundance of information in social network sites. This unfolds pivotal benefits in the recollection of past memories [9]. For example, it allows users to reminisce together about a past experience, or show one who was not there what they missed.

Yet another benefit from sharing memories with others is the possibility to enhance one’s own memory cues by offering a richer capture perspective for a given experience. The motivation for this stems from a technical limitation of wearable cameras. We previously found [11] that, regardless where you wear your camera, their images may not lend themselves well as memory cues [52]. For example, a camera worn on the neck or chest will often have its lenses covered by clothes or hair, might capture user’s hands and arms, or simply face the wrong way. Even when unobstructed, the first-person-view typically shows only a small part of the scene, potentially never capturing a person sitting right next to the person recording. In a similar vein, cameras embedded in glasses may produce distorted and out of focus images due to users’ frequent head movements.

In light of this, we acknowledge two complementary data sources that can enhance the overall quality of memory cues. Firstly, the views from the wearable cameras of other co-located users (i.e., people that are experiencing a particular moment together) may offer a richer view than one's own. For instance, our first-person camera will not show who is next to us, while the view captured from the person opposite from us would. Secondly, the view of co-located infrastructure cameras allows them to capture comprehensive scenes, completely unobstructed. Furthermore, as Clinch et al. point it out [159], environmental cameras do not have the same size restrictions as wearable cameras, hence their bigger and heavier sensors can provide higher-quality pictures than wearables. Consequently, we will now look into this type of sharing among co-located peers, with the ultimate goal of enhancing the quality of produced memory cues. We refer to such sharing as “implicit memory sharing”.

Sharing memory cues with others raises significant privacy and security implications. Captured experience data will inevitably feature sensitive and personal information. For instance, cameras might record users while they work in front of computer screens, or even continue to record images when one enters a bathroom.

Receiving experience data that were shared from others opens up additional threats that go beyond privacy risks. Co-located others could share fabricated images, allegedly featuring an accurate reflection of the shared experience. As we previously saw in Chapter 3, our episodic memories can be easily affected by falsified image cues.

In this chapter, we set out with the goal of enabling an implicit but secure sharing of memory cues among peers that are part of the same experience. Therefore, starting from the trusted wearable camera from Chapter 3, we propose a system that will implicitly and securely exchange images among co-located peers, as well as a protocol to verify any such shared but modified images obtained from others. Continuing from RQ1 defined in section 1.2, in this chapter we address the following research questions:

- **RQ2:** How can we seamlessly and securely share captured experiences with co-located others, avoiding the risk of accidental oversharing, i.e., sharing with the wrong audience, or sharing parts of a capture that we would otherwise have kept to ourselves?
- **RQ3:** How can we verify the integrity and provenance of experience data which we obtain from others to detect the sharing of falsified experience captures?

*Parts of this chapter are based on the following publications:*

- **A. Bexheti**, M. Langheinrich, and S. Clinch, “Secure Personal Memory-Sharing with Co-located People and Places,” in *Proceedings of the 6th International Conference on the Internet of Things*, ser. IoT’16. New York, NY, USA: ACM, 2016, pp. 73–81
- **A. Bexheti**, M. Langheinrich, I. Elhart, and N. Davies, “Securely Storing and Sharing Memory Cues in Memory Augmentation Systems: A Practical Approach,” in *Proceedings of the 17th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom’19)*, 2019, p. 10

## 4.1 Threat Model and Requirements

Sharing images, also quite private photographs, is a common activity in today’s social media platforms. However, having a system that does such sharing automatically and implicitly (i.e., without active user involvement) can radically change the nature and scale of disclosed information. Continuing from our list of threats started on page 57, listing T1 and T2, we foresee the following additional threats which are also depicted in Figure 4.1:

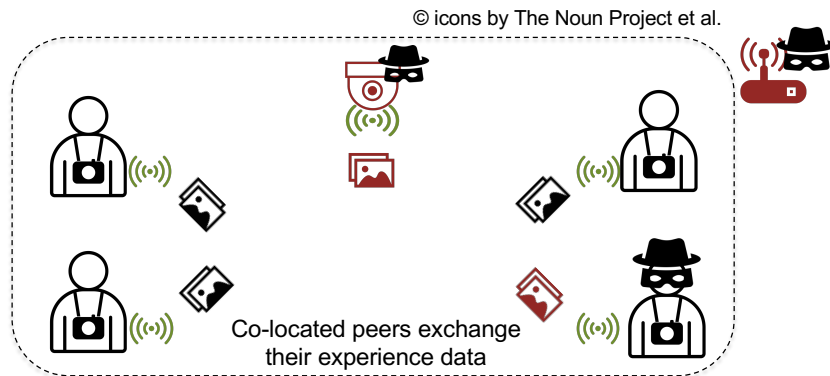
### **T3: Memory Oversharing.**

When implicitly exchanging memories among co-located users, there are two key risks of accidental data oversharing. Firstly, one can share their memories **with the wrong audience** (e.g., sharing meeting data with a person sitting next door to the meeting room). Secondly, one can risk sharing **data of the wrong experience** (e.g., mistakenly disseminating data of a meeting from two weeks ago), or **sharing more data** than one intends to (e.g., continuing to share data even after a meeting is over).

The envisioned attacker is will try to access others’ data by passively sniffing the traffic of generated metadata that allows co-located users to legitimately exchange data with each other, actively claim co-location, or impersonate a peer.

### **T4: Receiving fake memories.**

Maybe obviously, during an automated data sharing practice, one will often exchange experience data with co-located others who one knows well



*Figure 4.1.* A visualization of envisioned threats when seamlessly exchanging data among co-located peers and infrastructure cameras. Malicious peers or infrastructure providers can intentionally share modified images of a joint experience with the victim. Other adversaries that are not part of the event can also try to passively sniff the traffic in order to get access to the exchanged data.

and trusts, such as friends and family. However, there will be certain situations when one will share, but also will receive, experience data from others that one does not necessarily know well, and hence may have less trust on them or no trust at all. In this context, a second peril when exchanging memories with strangers is that dishonest data sharers (either peers with wearable sensors or an infrastructure provider sharing data from a fixed sensor) can behave maliciously by providing intentionally altered data. As a consequence, in addition to the previously described risks of human memory manipulation in Section 3.1 (i.e., T1: hacking one's camera in order to deliver already modified images; and T2: compromising the integrity of one's digital memory repository), this opens the door to yet another way of manipulating one's overall memories. This form of attack can be carried out by a dishonest user or infrastructure service provider that shares falsified images with their co-located peers.

#### **T5: Tracking users.**

Tracking is a prominent risk in our envisioned system for memory sharing. An attacker could simply listen passively for the generated network information between peers and thereby track their location. Attack vector space is increased if we assume that an attacker can also get access to image download requests from users' repositories. In this case the attacker could infer additional information, such as who was with whom (knowing that usually sharing happens between co-located peers).

## Security Requirements

Starting from such threats, we draw up a set of requirements for the seamless but secure exchange of experience data with co-located others:

**R4: Secure experience data exchange between co-located entities.**

Recorded data should be solely exchanged with people that are in close physical proximity with each other. When possible, data will also be sourced from a nearby infrastructure camera. People that simply pass by should not receive any part of the shared data. Moreover, any such sharing should be fine grained and account only for the data captured during the period of co-location and interaction. For instance, sharing with a person should immediately stop should that person leave the event and walk away.

**R5: Support for verifiable cue modifications.**

From the viewpoint of recipients, it is pivotal to have a way to verify the integrity and provenance of obtained data, i.e., whether the received image accurately reflects the experience and has not been modified in the meanwhile. However, as it may be perfectly legitimate to share only a modified version of one's captured image, e.g., to block a certain region of the image showing sensitive information, the system should allow recipients to reliably identify the modified parts of the image, obviously without revealing the hidden information. Recipients can then decide whether they should keep or discard the obtained image.

**R6: Prevent user tracking.**

Beyond the previous requirements, it should be infeasible to infer anything about the users (e.g., their identity or their traversed path) or the disseminated data by simply observing the generated network traffic. This can be achieved, for instance, adding "noise" to the generated traffic, or dynamically changing the packet contents and transmitting schedule, and thereby reducing chances of making any connection between the different packets and users.

## 4.2 Related Work

The work described in this chapter is grounded in the following fields: (i) systems for proximity detection, and (ii) zero-knowledge protocols for proving knowledge of a particular piece of information without revealing it.

## Proximity Detection Systems

Detecting the presence of nearby individuals is a popular feature of location-based services (LSB) [160, 161, 162]. To achieve this, users' mobile devices need to share their GPS location traces with centralized location servers. Then, whenever two users move to each others' vicinity, the server will, for instance, send them notification messages. However, GPS-based approaches do not work indoors and they are not accurate enough for close proximity estimation.

Other approaches rely on WiFi signals to estimate a device's position [163, 164]. In these systems, mobile devices can compute their position by scanning for nearby WiFi access points and then cross-referencing the access point positions in pre-computed databases. Unlike these approaches, Krum and Hinckley [165] propose a proximity detection system that does not compute a device's absolute position, allowing it to work "out of box" without any a priori configuration and setup. Their NearMe system compares clients' WiFi signatures (e.g., access point names and signal strengths) to compute devices proximity to one another. The detection range is lower-bounded to the coverage area of one access point, thus, resulting in a minimum range of 30–100 meters. In the same vein, Li et al. [166] present a similar approach but using cellular tower readings instead.

Our approach is closely related to the class of peer-to-peer proximity protocols based on low-range communication technologies such as Bluetooth and RFID [167, 168, 169]. For example, Liu et al. [170] present a fine grained proximity detection system that can provide adequate accuracy for inferring face-to-face user proximity. In addition to Bluetooth RSSI values, their solution also incorporates readings from light-sensors for greater accuracy. Findings from this work show that Bluetooth offers an accurate and power-efficient mechanism for user proximity estimation.

Cattuto et al. [171] have investigated the use of RFID technology for measuring device proximity. The authors have designed conference-like badges equipped with RFID tags. Such badges exchange low-power radio packets in order to sense the neighbourhood and proximity with each other. Since these badges are equipped with a single RFID radio, they alternate between transmitting and scanning cycles, both for advertising one's presence to others and listening for presence information sent by others. Lastly, the designed badges can reliably exchange radio packets within 1–1.5 m of one another.

Many of the aforementioned proximity detection systems require one to reveal their location to other entities. However, sharing one's location can greatly endanger users' privacy. As a result, prior work has also investigated the possibilities of designing privacy-aware proximity detection systems. For instance,

Zhong et al. [172] present a system that enables a user to learn information about another person's location only if the two are actually nearby (i.e., if their distance is below a specified threshold). Depending on the risk appetite, authors have designed three protocols to solve this problem. Their first protocol, "Louis", relies on a trusted third-party entity that does not learn any location information. "Lester", their second protocol, works without any third-party, but it allows one, with a bit of extra effort, to obtain information on a user's location even when the two of them are not nearby. Authors' third protocol, "Pierre", corrects the limitation of Lester on the cost of not providing a precise distance measure to nearby friends. In another work, Siknys et al. [173] introduce the concept of *vicinity regions*, which compared to the standard circular shape regions employed by other works, can be of any shape. Their system is modeled using the client-server approach, and it employs spatial cloaking and encryption to ensure location-privacy. In yet another work, Mascetti et al. [174] provide a rigorous and formal definition of location privacy preferences and adversary model. One particular feature of their model is that it considers an a priori probabilistic knowledge of a user's location that an adversary can have (e.g., a user is likely to be located in her hometown and not in another one). Furthermore, authors present two privacy-aware proximity detection protocols, which they formally analyze using the proposed adversary model and prove that location privacy is guaranteed.

Finally, many more interesting examples of co-presence detection systems can be found in the survey work by Conti and Lal [175].

## Protocols for Verifying Shared but Modified Images

Designing protocols for the verification of modified images is an active field of research. In recent work, Naveh and Tromer propose PhotoProof [156], a protocol for verifying modifications performed on an original image. Their solution is based on digital signatures and the *proof-carrying-data* (PCD) concept: a cryptographic primitive for secure execution of distributed computation [176]. After capturing a signed image with a trusted camera, a user can modify the image according to a set of permissible transformations, and then compute an integrity proof following the PCD algorithm. Using these proofs, anyone can then verify if the performed transformations is permissible or not. The verification procedure takes less than half a second, however, generating a PCD proof for a small image of  $128 \times 128$  pixels takes about 300 seconds on a powerful<sup>1</sup> machine.

---

<sup>1</sup>PhotoProof [156] used a machine with a quad-core CPU at 3.4 GHz and with 32 GB of RAM for their proof creation task.

Chabanne et al. [177] propose a similar scheme to verify redacted (obfuscated) pixels of scanned documents. Their solution relies on the *extracted signature* scheme [178], which allows one to remove parts of a previously signed document and re-sign it without the knowledge of the signer’s secret key. The extracted signature can be still verified with the signer’s public key and without having the removed parts from the original document. While this solution is more efficient than PhotoProof, partially because it supports only one type of image modification (i.e., pixel obfuscation), it is still too complex for low-power wearable cameras. Generating a redaction proof of a gray-scale image with  $1200 \times 800$  pixels takes 124.5 seconds and 39.7 seconds on a single-core and octa-core system, respectively<sup>2</sup>.

We propose a less flexible but more efficient scheme that uses cryptographic hash functions. Similar to the work of Chabanne et al. [177], our protocol only supports the simple “blinding” (i.e., blocking) of certain parts of the image, rather than operations that apply on the whole image (e.g., cropping, color adjustment). Both our own experience, as well as the survey paper from Bettini and Riboni [18], have shown that area blinding is crucial for addressing privacy issues in pervasive systems that capture and store visual data streams.

## 4.3 A Systematic Approach for Memory Sharing

In this section we investigate a technique for using our trusted camera for the seamless exchange of self-captured images with co-located others, and for acquiring data sourced from any co-located infrastructure devices. We start by eliciting initial system requirements, and then describe our system for sharing personal lifelog images with others, with a view to preserving the privacy of all involved users. With a focus on the technical side of this work we report initial security analysis results and performance measurements.

### 4.3.1 System Requirements

We delineate the system requirements through a scenario involving various daily activities. For each activity, we describe the envisioned data sharing practice, highlighting how captured data should be shared and with whom.

---

<sup>2</sup>Chabanne et al. [177] used two different computers for generating redaction proofs: 1) one computer with a 3.6 GHz single-core CPU and with 4 GB of RAM; and 2) a second computer with a 2.9 GHz octa-core CPU and with 16 GB of RAM.



- **Morning walk to the office.** We want access to any fixed infrastructure camera capturing our walk, yet must also prevent their owners from easily tracking our location.
- **Encounter a work colleague.** While entering our department building, we meet a colleague in the hallway for a chat. We want access to our colleague’s data captured during the encounter, and data from any hallway cameras that captured the meeting.
- **Attend a work meeting.** After lunch we attend a work meeting. During the meeting, we want in-room sensors to provide access to high-quality captured data (camera, microphone, board contents, etc.), as well as captured data from co-located colleagues. People who simply pass in front of the meeting room should not have access to this data.
- **After the meeting, chat with a colleague.** While packing our bag in the meeting room, we have a quick chat with a colleague. Albeit high-quality capture of the meeting room is stopped, we still want to capture data from our colleague’s wearable camera. Colleagues who have already left the room should not have access to this data.
- **Verify integrity of some dubious images.** Later on, our system informs us that we have obtained some modified images from our colleagues at the meeting. After a quick inspection, we see that one of them has blurred all image regions that showed laptop screen. We decide to keep those images since their overall integrity is intact and they do not pose any manipulation risk of the memories of the event.

### 4.3.2 System Description

This section describes the full life cycle of a shared capture session – from peer discovery to data exchange and access control. Table 4.1 itemizes the identifiers used in this section, while Figure 4.2 gives an overview of data flows. In a nutshell, a user’s camera advertises its willingness to both share its self-captured data and to acquire data of other co-located peer devices<sup>3</sup> by broadcasting periodically updated access tokens and a periodically changing public key (see section “*Advertise Sharing Availability Using Memory Beacons*”). Tokens represent a way

---

<sup>3</sup>Peer devices may include both the personal capture cameras of other users and fixed capture installations, e.g., room cameras.

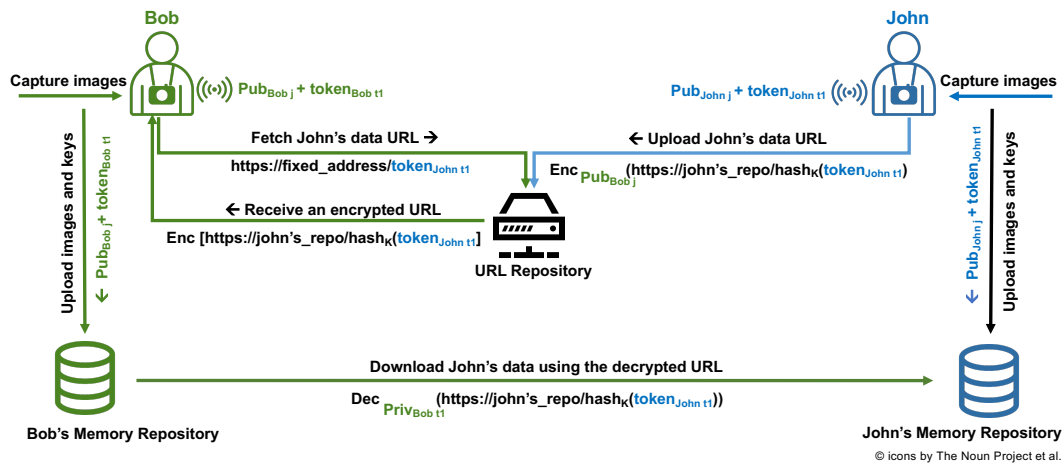


Figure 4.2. Data flow for capturing and sharing lifelog data between two co-located people. Bob's repository downloads John's shared capture data, using decrypted URLs available from a URL repository.

of letting others know where to access data one is willing to share. Instead of sharing captured data directly, only a reference (i.e., the token) is shared, which not only improves the security of the scheme but also lowers bandwidth requirements. Both tokens and public keys are sent out only when the device senses the presence of a co-located peer in order to prevent a malicious person from tracking its location by passively sniffing announced data (see section “*Smart Memory Beacons Based on Improved Peer Detection*” for details). When the camera of another peer observes such announcements, it will trigger a key-exchange with the peer (see section “*Key Exchange Protocol*”). To perform such a P2P key exchange in a multi-user environment, all peers broadcast both access tokens and their (periodically changing) public keys. A user's camera grants access to its captured data that it is willing to share only to co-located peers. It does so by encrypting both the shared data, as well as the corresponding repository access information, with all peers' public keys (using “broadcast encryption”). As a consequence, only the peers who possess the valid private key will be able to both get and decrypt the shared data (see section “*Access Control to Shared Data*”). We will now describe each of these steps in more detail.

### Advertise Sharing Availability Using Memory Beacons

A device advertises its capability and willingness to let others access its captured data by simply announcing an online location (URL) of where the data will eventually be located (real-time upload of captured data is not a key requirement).

Variable	Description
$Pub_{ij} / Priv_{ij}$	Temporary session key-pair; “ $i$ ” identifies device, “ $j$ ” identifies session.
$K_{hash}$	Key for hashing tokens to storage location.
$token_{it}$	Access token, where “ $i$ ” is the device that issued it and “ $t$ ” is the time period it was issued in.
$t_{dwell}$	Minimum dwell time before adding or removing a peer $Pub_{ij}$ key to the list of registered peers.

Table 4.1. Variables used in the system for sharing memory cues.

This also means that a device does not exchange the actual data itself over the wireless channel, which minimizes both bandwidth and energy consumption.

The actual announcements (which we call *memory beacons*) are sent using the Bluetooth Low Energy (BLE) short-range wireless protocol. The advertised online location is not fixed, but is based on an implied fixed system-wide base URL (not sent). This approach has several benefits. First, it has low bandwidth requirements as we can only broadcast the actual image identifier, i.e., the token in this case, and omit the other parts of a URL link as specified by the URL protocol. Second, given the already small BLE packet size of 25 bytes (see Figure 4.7 on page 102 for the actual BLE data package), it allows us to use all this space for the token. This improves the security of the scheme as it lowers the chances of an attacker (who might already know the fixed URL of a user’s repository) to guess a valid token and subsequently use it to obtain data from the user’s repository. Third, this approach gives more flexibility to users in choosing the service provider for their memory repository. Switching repositories requires one only to update the URL address of the new repository in the URL resolver. All issued tokens would then resolve correctly to the updated repository address. Similar such schemes are already in use. For instance, the Digital Object Identifier (DOI) scheme usually includes a DOI as a URL which uses a resolver through an HTTP proxy at a fixed base URL (<http://dx.doi.org>). Finally, we note that this approach may have some privacy implications similar to those in the case of the Domain Name System (DNS) [179]. By inspecting token queries going to the URL resolver one can infer, e.g., who was co-located with whom.

The beacon therefore only provides a continuously changing *access token*, under which peers can find captured data for the short period over which this token was used. Our current prototype updates tokens every few seconds, though longer or shorter intervals may be equally possible. Access tokens are simply

large numbers – enough so that accidental overlap of tokens becomes unlikely. Due to the limited size available in BLE announcements (see section “*Implementation*” below) we use 200 bits in our prototype, which does not scale well in the long term<sup>4</sup> but is probably fine for immediate deployments. With a view towards allowing recipients to verify the integrity of obtained images, access tokens are computed as a function of the actual image content that the camera just captured. This process is part of our proposed protocol for verifying modifications of shared images and is described in more detail in the next section (i.e., Section 4.4). Peers use captured tokens to find the shared data at the known system-wide URL. In the following we denote tokens as  $token_{it}$ , where “ $i$ ” is the issuing device and “ $t$ ” is the time period for which the token was issued. Their short lifetime means that as soon as a peer leaves the range of the device’s beacons, they are unable to access any of the data captured at a later time, as this data will use different access tokens.

While tokens thus provide some access control, as clients need the token to know the address, one might accidentally “stumble” upon uploaded data simply by trying arbitrary tokens. To prevent this, peers also need an access key to decrypt the data present at the token address. As part of the announcements, devices thus also periodically send out a public key  $Pub_{ij}$ , where “ $i$ ” again identifies the device and “ $j$ ” identifies a “sharing session”. Whenever a device stops sharing captured data, i.e., when all peers leave, it starts a new session upon first discovering a new peer. Hence, over the course of a day, several public keys will be generated and used. In our prototype, every 10<sup>th</sup> announcement carries a public key instead of an access token. We describe the use of these public keys to access shared data in section “*Access Control to Shared Data*”.

### Smart Memory Beacons Based on Improved Peer Detection

Broadcasting a continuous stream of tokens could potentially allow a passive attacker to track a device. While tokens change frequently, public keys (which are interleaved with tokens) may not change that often. With few devices around, there may simply not be enough “noise” for a device to “blend-in” with others. Infrastructure sensors on the other hand are basically peers who do not move. No privacy issues prevent them from simply always broadcasting beacon announcements.

<sup>4</sup>One billion people, each sending 86’400 individual tokens per day (one new token per second), would create  $8.6410^{13} \approx 10^{14}$  tokens per day. If we want these to be around for some 3 years (1000 days), we have  $10^{17}$  tokens “in use” at any time. Given the birthday paradox, this would leave a  $10^{17}/2^{200/2} \approx 10^{17}/10^{30} \approx 10^{-13}$  chance of overlap.

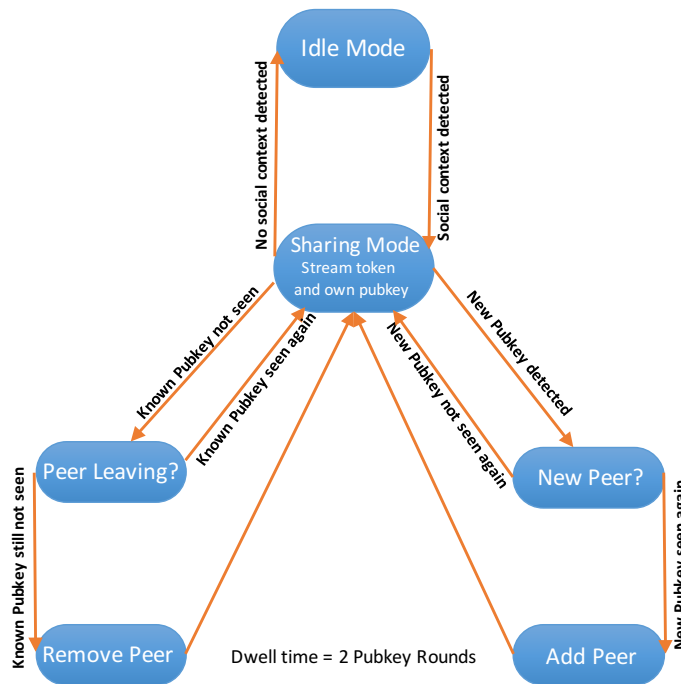


Figure 4.3. System state diagram showing the transition between Idle Mode and Sharing Mode and the peer presence detection, using an example dwell time  $t_{dwell}$  of two rounds.

For this reason, users' devices periodically stop beaconing, moving from *sharing mode* into an *idle mode*. Mode transition is triggered by the absence of other proximate peers; the device will switch from idle mode to sharing mode if it detects the presence of a peer in range, and vice versa (see Figure 4.3).

Presence detection cannot therefore rely on the reception of tokens from other peers, as these may be in idle mode. Instead, we envision the use of a *social detector sensor* – a sensor designed to detect social engagement with other peers. Prior works propose different approaches for designing such a sensor. For example, Nakkura et al. [180] present an audio based sensor which can detect if a device owner is engaged in conversation. In another work Xu et al. [181] use a similar audio-based approach to count the number of different speakers that participate in an experience. In yet another work, Fathi et al. [84] use image recognition to detect faces in captured photographs – a face lingering in front of the camera may well indicate social interaction. Note that, however, in this thesis we focus only on the actual data sharing process using a proximity detection based on BLE distance threshold. However, the design of a social presence detection sensor as described above is out of the scope of this thesis.

### Key Exchange Protocol

Upon encountering a memory beacon from another device, or upon detecting social engagement, a device will trigger beacon announcement. As part of this process, it will run a “broadcast-based” key exchange protocol with other co-located devices to exchange public keys for the session ( $Pub_{ij}$ ). The exchange is broadcast-based in the sense that there is no direct communication or a handshake between the devices whatsoever – all data is sent via broadcast. Once a peer’s public key is received, a device subsequently encrypts uploaded data using that public key. If multiple peers are detected, a broadcast encryption scheme [182] is used to encrypt the data using each peer’s public key (see section “Access Control to Shared Data” below).

In order to prevent passers-by from receiving a set of access tokens “by accident” and thus having access to captured data, we enforce a *dwelling time* ( $t_{dwell}$ ) during which the peer’s public key must be seen repeatedly. Only if a peer has been present for long enough, will we retroactively mark all captured data starting from its first detection as being also shared with this particular peer. As long as the peer stays in range (i.e., as long as its public key is periodically received), the device will keep the peer’s  $Pub_{ij}$  key within the list of authorized clients that can access its captured data (see Figure 4.3).

### Access Control to Shared Data

A device controls access to its captured data using both the access tokens sent during “sharing mode”, and the public keys received from other peers. Co-located peers can query that device’s database at a *known base address* and exchange tokens they have collected for actual data:

$$\longrightarrow \text{https} : // \text{fixed\_address} / \text{token}_{it}$$

We previously discussed the benefits of having a fixed URL resolver. For one thing, it facilitates the use of one’s own repository to host actual captured data, the known base address does not directly provide data but instead offers a “redirect” to the data host (similarly to today’s digital object identifier, dx.doi.org). In order to prevent trivial tracking through resolution of collected tokens (see threat T5 in page 80), the URL redirect is encrypted with the public keys of all authorized peers:

$$\longleftarrow E(\text{https} : // \text{user\_repo} / \text{data\_address})$$

A peer accessing the token URL thus receives an encrypted value that, when decrypted using the peer's private session key at the time (which the peer will need to keep track of), will result in another URL for access to the actual data.

To achieve a multi-user encryption setup, we use *broadcast encryption* [182]: instead of re-encrypting the data URL for each peer, we use a (random) *symmetric key* to encrypt the URL with a symmetric cipher, and then encrypt that symmetric key with each recipients  $Pub_{ij}$  key. While URLs are not long, encrypting only a 128-bit (16 byte) symmetric key instead of a 30-40 byte URL is clearly more economical, especially as this will need to be performed multiple times (once for each peer).

Given the above, captured data can then be made available at a repository of a user's choice. Note that the actual shared data (e.g., images) does not need to be encrypted, as only those who both know the token  $token_{it}$  and have had their session public key  $Pub_{ij}$  captured by the device (and subsequently used to encrypt the used symmetric key) will be able to find URLs of shared images. However, if desired, the same broadcast encryption scheme can of course be applied to the shared data (e.g., to prevent accidental disclosure to an attacker guessing the URL) [183].

In order to come up with the final URL on the user's repository, the user can simply use a keyed hash, together with a long-term secret key  $K_{hash}$ , to convert each access token into a storage URL on their own repository. A non-keyed hash function would not be sufficient, as it would allow an attacker who knows the base address of the user's repository (e.g., from a previous peer exchange) to simply "fish" for images using only captured access tokens.

A device can revoke access to (unretrieved) data at any time. To invalidate access for a peer, the device simply removes the public key  $Pub_{ij}$  of the peer from the set of authorized devices and re-runs the broadcast encryption scheme to re-encrypt URLs with a new key. This, of course, only makes sense if the peer has not already downloaded the shared data.

## 4.4 Verifying Shared but Modified Image Cues

As we previously argued, being able to share captured experiences with co-located peers can yield tangible benefits for memory cue creation. For a recipient of such shared images it is thus crucial to have some guarantee that the image has not been maliciously (i.e., invisibly) altered.

Variable	Description
$\kappa$	Tile size for the image splitting procedure.
$T_I$	Collection of tile fingerprints for image $I$ .
$salt_I$	Random number for computing tile fingerprints.
$\tau_I$	Hash-based fingerprint of $T_I$ .
$SK_{pub/priv}$	Derived from SRK, used to sign $\tau_I$ .
$\sigma_{\tau_I}$	Signature over $\tau_I$ .
$R_{ATT}$	TPM-signed platform attestation report.
$\Sigma_I$	Sharing certificate for image $I$ that encodes $T_I, salt_I, \sigma_{\tau_I}$ and $R_{ATT}$ .

Table 4.2. Variables used in the protocol for verifying shared images.

While it is trivial to verify that the image has not been altered (using the secure signature of the peer’s camera, which the receiving user can verify), there are perfectly legitimate reasons for a peer to share only a modified version of their captured image. Oftentimes, our own captured images will contain private information (e.g., a document, a phone or laptop screen) that should not be shared with others.

The goal is hence to allow an image recipient to reliably identify all modified parts of an image. The identification of the modified and unmodified parts should obviously be possible without revealing the original, unmodified image – the verifier should only have access to the modified image.

To this end, we will present two protocol variants for addressing this problem. The first variant is based on a practical scheme that allows only one type of image modification. It can be efficiently implemented on today’s low-power wearable cameras. The second variant is a more general scheme that supports more image modification operations. However, due to the complexity of the underlying cryptographic primitive, it is not yet possible to realize this scheme in practice.

#### 4.4.1 Variant 1: A Practical Protocol Based on Hash Schemes

Our proposed protocol focuses only on supporting the simple “blinding” (i.e., blocking or blurring) of certain parts of the image, rather than operations that apply to the whole image (e.g., cropping, color adjustment). Both our own experience, as well as other’s prior work (e.g., [79, 184]), has shown that area blinding is crucial for addressing privacy issues in lifelogging imagery.



The protocol builds on the system we reported in the previous section 4.3, which enabled the seamless sharing of lifelogs between co-located peers. One element of the sharing protocol we developed for this are *tokens* – numerical identifiers that each user’s camera broadcasts in real-time (using a short-range wireless technology), and which are frequently (e.g., every 5-10 seconds) updated. All images taken by a camera are stored in a user’s repository under the token that was active at the time. In this way, only those who received the tokens are able to query the sharer’s memory repository later to retrieve the image.

While we previously did not specify exactly how tokens were computed, we now introduce a new constraint about them: access tokens are computed as a function of the actual image content that was just captured. This allows the data sharer to not only regulate access to the image, but to also “commit” the image’s content publicly without actually sharing the original image itself. By furthermore signing tokens with the camera’s private-key, we can ensure image authenticity. Finally tokens allow us to support the verification of modifications, e.g., obfuscations, to a certain unmodified (but not shared) image. This process is described below (see also Protocol 2 – variant 1 in Figure 4.4), while Table 4.2 itemizes the used identifiers.

To compute a token  $\tau_I$  for an image  $I$ , the image is first chunked into “tiles” (step 1, Figure 4.4). A tile is defined as a rectangular area of arbitrary dimensions, and is the smallest area that can be modified (see Figure 4.5 for an overview). For each tile we then compute a *hashed fingerprint* following the procedure as in step 3 of Protocol 2: the tile’s content (i.e., the serialized set of its pixels) is concatenated with its row and column indices (from the tile array), as well as with an additional per-image *salt*, and then hashed using a secure hash function.

The fingerprints of all tiles are concatenated and hashed again to create the final image token  $\tau_I$  (step 4), which is immediately announced to co-located peers (step 5) through a short-range BLE broadcast (explained in Section 4.3).

The set of tile fingerprints, a signature over the final token, the image salt, and the signed platform attestation report  $R_{Att}$  obtained from the camera’s TPM (step 6), are encoded in a sharing certificate  $\Sigma_I$  (step 7).  $R_{Att}$  is generated using token  $\tau_I$  as a nonce (see section 3.3.3 for the certificate generation process).

Before making the image accessible from their repository at a later time, the data sharer can modify it by obfuscating any tile that contains sensitive information (as shown in Figure 4.5). The final  $\Sigma_I$  together with the captured image are then uploaded to the user’s memory repository (step 8). Accessing the data sharer’s repository at the token address  $\tau_I$  will then yield the modified image  $I'$ , the tile hash-set  $T_I$  of the unmodified image  $I$ , the token’s signature  $\sigma_{\tau_I}$ , the attestation report  $R_{Att}$ , and the corresponding *salt* <sub>$I$</sub>  (steps 9, 10).

---

**Protocol 2** Verifying Shared But Modified Images (Variant 1)
 

---

**A. Generate and disseminate image sharing token**

 for each newly captured image  $I$ 

1.  $tiles_I = splitImage(I, \kappa)$   
//split image into rectangular tiles of size  $\kappa$
2.  $T_I = []$  //empty set for storing tile fingerprints
3. for columns  $i$  in  $tiles_I$   
    for rows  $j$  in  $tiles_I$ 
  - (a)  $h_{i,j} = H(i||j||\phi(t_{x,y})||salt_I)$   
    //where  $\phi(t_{x,y}) = p_1||p_2||\dots||p_{m*n}$ , is a string serialization of  
    //all pixels that are in tile  $t_{i,j}$
  - (b)  $T_I.add(h_{i,j})$
4.  $\tau_I = H(\phi(T_I))$  //where  $\phi(T_I) = h_{0,0}||h_{0,1}||\dots||h_{m,n}$
5.  $broadcastToken(\tau_I)$  //broadcast  $\tau_I$  to co-located peers via BLE  
    //as explained before in Section 4.3
6.  $\sigma_{\tau_I} = sign(\tau_I, SK)$   
     $R_{ATT} = platformAttest(nonce = \tau_I)$   
    // sign  $\tau_I$  and generate a fresh TPM signed camera platform attestation  
    //bound to image  $I$
7.  $\Sigma_I = \{T_I, salt_I, \sigma_{\tau_I}, R_{Att}\}$  //encode everything in a sharing certificate  $\Sigma_I$
8.  $uploadData(I, \Sigma_I)$  //upload  $\Sigma_I$  to user's memory repository and link it  
    //with image  $I$

**B. Verify authenticity and modifications of a shared image**

 for a token  $\tau_I$  that was received from a co-located peer

9.  $I', \Sigma_I \leftarrow downloadImage(\tau_I)$ ,  
    //obtain an image and its certificate using token  $\tau_I$
  10.  $\{T_I, salt_I, \sigma_{\tau_I}, R_{ATT}\} \leftarrow \Sigma_I$  // extract the certificate
  11. if  $verifyPlatform(R_{ATT}, nonce = \tau_I)$   
    if  $verify(\sigma_{\tau_I}, \tau_I, SK)$  && if  $H(\phi(T_I)) == \tau_I$ 
    - (a) Split the received image  $I'$  and compute a tile set  $T'_I$  following  
    steps 1-3 from above
    - (b) for index  $i$  in range of  $length(T'_I)$   
    if  $T'_I[i] \neq T_I[i]$   
         $drawFrame(I', T'_I[i])$  //draw a red frame in  $I'$  around  
        //the area of tile  $T'_I[i]$
- 

Figure 4.4. Pseudocode of the image verification protocol (variant 1).

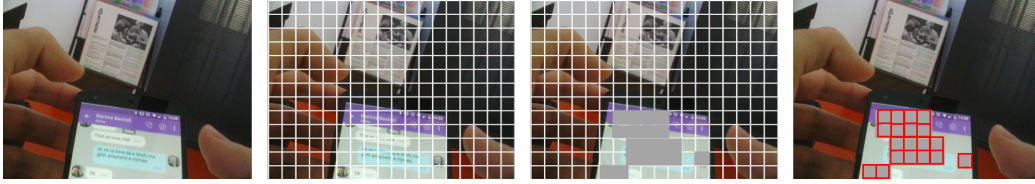


Figure 4.5. The process of “blinding” an image region before sharing. From left: (1) the unmodified image  $I$  of size  $900 \times 650$  pixels; (2) the image divided into  $18 \times 13$  tiles, each made of  $50 \times 50$  pixels; (3) blocking the tiles that contain information which should not be disclosed; and (4) the final modified image  $I'$  which is ready to be shared.

Following step 11 from Protocol 2, the data recipient can now verify which tiles have seen modifications, and which tiles come from the original unmodified image  $I$ . At the outset, she will verify that the token  $\tau_I$  used to access the image is indeed a hash of the concatenated tile hash-set  $T_I$ . Next, she will chunk the received modified image  $I'$  (using the same tile size used in the unmodified image  $I$ ) and then inspect each tile of  $I'$  individually using the following procedure. For each tile  $t'_{i,j}$  of  $I'$ , she checks its integrity by computing the tile’s hash  $h'_{i,j} = H(i||j||\phi(t'_{x,y})||salt_I)$  and matching it with the value given in the corresponding tile hash-set  $T_I$ . Now, all modified tiles (i.e., where the hashes do not match) can be marked, e.g., by drawing a red frame around them in the displayed image  $I'$ , allowing the receiving user to easily verify which tiles have been obfuscated (or otherwise modified). Finally, based on the verification results, recipients can draw their conclusion regarding the “trustworthiness” of the obtained image, i.e., whether to accept or discard it as a memory cue.

#### 4.4.2 Variant 2: A Protocol Based on Homomorphic Encryption

Beyond the operations of blocking or blurring certain regions of an image, some peers may like to perform global image modifications, such as compressing their raw images, adjusting image colors, or extensively retouching the image. Using the previous protocol variant, such changes would result in all or very many tiles being marked as “modified”, making it impossible for the recipient to truly understand the extent of the modifications: did the sender just adjust the colors, or did they completely redraw the image (e.g., adding or removing objects)? All of which constitutes a modification of the originally signed image that cannot be addressed using the previous protocol variant.

Therefore, to realize a scheme that would account for diverse image operations, we investigated the opportunities offered by *homomorphic cryptosystems*. A homomorphic cryptosystem is a special kind of cryptosystem that allows simple algebraic operations (such as addition and multiplication) to be performed on the data after encryption and obtaining the same result as if the algebraic operations were performed on the data before encryption [185]. For example, given two messages  $x$  and  $y$ , a homomorphic encryption scheme  $E()$  fulfills the following condition:

$$E(x \oplus y) = E(x) \otimes E(y), \quad (4.1)$$

where  $\oplus$  and  $\otimes$  represent some operations (such as addition, subtraction or multiplication) on the plaintext and ciphertext domains, respectively. Such systems have been employed in various scenarios like secure electronic voting, private information retrieval systems [186], privacy protection schemes [187], and sharing secret data [188].

In the context of sharing memories, a homomorphic scheme would allow the data recipient to verify any modifications (that the sharer claims to have done on the original image as captured by her camera) by performing the same modifications on the original but encrypted image, and hence without disclosing the sensitive information contained in the sharer's unmodified image. Since the recipient peer will receive the modified image, she can check if the modifications that the sharer performed on the original image match with the modifications that she performs on the unmodified but encrypted image.

## Requirements

At the outset, we delineate a set of requirements that we deem necessary for realizing such a scheme:

- **R1: Pixel-based encryption.** In order to support image processing on the encrypted domain, image encryption has to be performed on pixel level, where each pixel will be encrypted separately. This requirements comes from the fact that usually image processing approaches operate on pixels, for example by changing pixels' colors according to some parameters. Furthermore, the pixel-wise encryption scheme should not come at the cost of weakened security. In order to prevent an adversary from deducing any information about a plaintext pixel from the ciphertext of another pixel, identical pixels should map to different ciphertexts.

- **R2: Support for fully homomorphic encryption.** Most image processing techniques involve both the operations of addition and multiplication. Hence, it is necessary that the underlying cryptosystem is *fully homomorphic*, i.e., supports both these operations on the ciphertext domain.
- **R3: Ensure confidentiality of the original image.** Verification of the claimed modifications done on an image should be performed without revealing the original unmodified image.

### Protocol Description

Similarly as with previous protocol variant (section 4.4.1), this variant is also tightly coupled with the system we reported previously (section 4.3). To meet the aforementioned requirements, here we propose a different approach for computing access tokens, which we describe below (see also Protocol 3 in Figure 4.6).

For a captured image  $\mathbf{I}$  at time  $t_0$ , the user's camera, at time  $t_1$ , will generate a new token by encrypting the image using a homomorphic encryption scheme prior to signing it with its internal key. Such encryption is performed on the pixel level. Thus, given an image  $\mathbf{I}$  with dimensions  $w$  and  $h$ , and a key material composed of a long-term secret  $K$  and  $n = w \times h$  randomly generated and distinct keys  $r$  (one per pixel), each pixel will be encrypted sequentially as follows:

$$c_{ij} \leftarrow E(p_{ij}, K, r_n), \quad (4.2)$$

where  $p_{ij}$  is the pixel at the  $i$ -th row and  $j$ -th column and  $c_{ij}$  is the computed ciphertext. Produced ciphertexts are then packaged in a  $w \times h$  ciphertexts array:

$$\mathbf{C}_I = \{c_{ij}\}. \quad (4.3)$$

The purpose of having a different random key  $r$  (alongside the long-term key  $K$ ) for each pixel is to ensure that identical pixels will always map to different ciphertexts. For simplicity, let us combine the computations from both equations 4.2 and 4.3 in a single function *ImgEnc*, that given as input an image, a long-term key and a list with the other random keys, it will sequentially encrypt each pixel of the image and return an array with the computed ciphertexts:

$$\mathbf{C}_I \leftarrow \text{ImgEnc}(\mathbf{I}, K, \mathbf{R}). \quad (4.4)$$

---

**Protocol 3** Verifying Shared But Modified Images (Variant 2)

---

**Data Sharer (Prover)****Data Recipient (Verifier)** $t_0$  : take original image  $\mathbf{I}$  $t_1$  : Encrypt  $\mathbf{I}$  and sign it $\mathbf{C}_I = \text{ImgEnc}(\mathbf{I}, K, \mathbf{R})$  $\sigma = \text{Sign}(\mathbf{C}_I, P_{\text{secret}})$ 
 $\xrightarrow{\mathbf{C}_I, \sigma}$ 
 $t_1 + \delta$  : obtain  $\mathbf{C}_I, \sigma$  $t_2$  : Modify  $\mathbf{I}$  according to  $\mathbf{T}$  $\mathbf{M} = F(\mathbf{I}, T)$ Encrypt  $\mathbf{M}$  $\mathbf{C}_M = \text{ImgEnc}(\mathbf{I}, K, \mathbf{R})$ 
 $\xrightarrow{\mathbf{M}, \mathbf{C}_M, \mathbf{T}}$ 
 $t_3$  : obtain  $\mathbf{M}, \mathbf{C}_M, \mathbf{T}$ Verify the signature of  $\mathbf{C}_I$ if  $\text{Verify}(\sigma, P_{\text{public}})$  continue:Translate  $\mathbf{T}$  to ciphertext domain $\Psi \leftarrow \text{Transform}(\mathbf{T})$ Apply  $\Psi$  to  $\mathbf{C}_I$  $\overline{\mathbf{C}}_M \leftarrow F'(\mathbf{C}_I, \Psi)$ 

Verify claimed processing

if  $\overline{\mathbf{C}}_M == \mathbf{C}_M$  accept image  $\mathbf{M}$ 


---

Figure 4.6. Pseudocode of the image verification protocol (variant 2).

The array of ciphertexts  $\mathbf{C}_I$  represents the image's token. Upon computing such token, the sharer's camera will first sign it with its internal private key and then send the token and its signature to co-located peers (by means of broadcast). The token and its signature are then picked-up by the recipient's camera at  $t_1 + \delta$ .

As previously stated, the choice of using a homomorphic cryptosystem is to allow recipients to verify any transformations that might have been performed on the obtained image. An image transformation is a process applied to its pixels.

Thus, given an original image  $\mathbf{I}$ , with dimensions  $w$  and  $h$ , a processing function  $F()$  can be seen as follows:

$$\{\hat{p}_1, \hat{p}_2, \hat{p}_3, \dots, p_{(\hat{w}*\hat{h})}\} \leftarrow F(p_1, p_2, p_3, \dots, p_{(w*h)}, \alpha_1, \alpha_2, \alpha_3, \dots, \alpha_k), \quad (4.5)$$

where  $p_i$  and  $\hat{p}_i$  are the respective original and modified image pixels, and  $\alpha_k$  is a transformation parameter. A simpler matrix-based notation yields the following:

$$\mathbf{M} \leftarrow F(\mathbf{I}, \mathbf{T}), \quad (4.6)$$

where  $\mathbf{I}$  and  $\mathbf{M}$  are the original and the modified images, respectively, and  $\mathbf{T}$  is the transformation *kernel matrix*. One can vary the contents of the kernel matrix in order to perform different image transformations (e.g., pixel blurring, color adjustment, image scaling, etc.). Having already sent out the token of the original image, the data sharer is free to apply any transformation to it before sharing the actual image with others. She will then share the modified image  $\mathbf{M}$  alongside the transformation kernel matrix  $\mathbf{T}$  in order to let the recipient know what sort of manipulation was done on the original image  $\mathbf{I}$ . Moreover, she will also share an encrypted version of the modified image  $\mathbf{M}$  (for reasons that are explained below), computed as in equation 4.4, using the same long-term secret  $K$  and the same set of random keys  $\mathbf{R}$ .

$$\mathbf{C}_M \leftarrow \text{ImgEnc}(\mathbf{M}, K, \mathbf{R}). \quad (4.7)$$

At the end of the image sharing process, i.e., at time  $t_2$ , the recipient peer would have obtained the following information: (i) the unmodified but encrypted image  $\mathbf{C}_I$ , (ii) the modified image  $\mathbf{M}$ , (iii) its encrypted version  $\mathbf{C}_M$ , and (iv) the kernel matrix  $\mathbf{T}$ . At this point, in a two-steps process the recipient peer can check if the claimed processing (as given by the kernel matrix  $\mathbf{T}$ ) is performed on the original image  $\mathbf{I}$  in order to get the modified image  $\mathbf{M}$ . Since the peer does not have the original unmodified image  $\mathbf{I}$ , the verification has to be performed using the encrypted image only.

In a first step, the recipient peer verifies the signature of the unmodified encrypted image  $\mathbf{C}_I$ . If it is valid, then the peer modifies  $\mathbf{C}_I$  using the provided kernel matrix  $\mathbf{T}$ . Thanks to the homomorphic encryption, the outcome of this will be the modified image but in encrypted form.

Let  $F'()$  be a processing function that runs over encrypted image pixels. Note that in order to perform any processing on the encrypted domain it is necessary to first transform the processing parameters  $\alpha_k$  (from the kernel matrix  $\mathbf{T}$ ) into the same domain space as that of the ciphertexts  $c_i$ . The recipient peer can then proceed with the transformation on the encrypted image  $\mathbf{C}_I$  as follows:

$$\{\hat{c}_1, \hat{c}_2, \hat{c}_3, \dots, c_{(\hat{w}*\hat{h})}\} \leftarrow F'(c_1, c_2, c_3, \dots, c_{(w*h)}, \psi_1, \psi_2, \psi_3, \dots, \psi_k), \quad (4.8)$$

where  $c_i$  is the encrypted pixel of the unmodified image,  $\hat{c}_i$  is the modified encrypted pixel, and  $\psi_k$  is the processing parameters transformed in the same domain as the ciphertexts. Simplifying this with matrix notation yields:

$$\overline{\mathbf{C}_M} \leftarrow F'(\mathbf{C}_I, \Psi). \quad (4.9)$$

In a second step the data recipient can compare if the outcome of step 1 (i.e., modifying the encrypted original image) matches with the encryption of the modified image that the data sharer sent her:

$$\begin{aligned} F'(ImgEnc(\mathbf{I}, K, \mathbf{R}), \Psi) &= ImgEnc(F(\mathbf{I}, \mathbf{T}), K, \mathbf{R}) \equiv \\ F'(\mathbf{C}_I, \Psi) &= ImgEnc(\mathbf{M}, K, \mathbf{R}) \equiv \\ \overline{\mathbf{C}_M} &= \mathbf{C}_M \end{aligned} \quad (4.10)$$

If there is match, the recipient peer can be sure that only the claimed modifications were done on the original image as captured by the sharer's camera. While in principle the recipient peer could have encrypted the modified image  $\mathbf{M}$  (since she has it in clear) and match it with modifications she did on the encrypted image (or even decrypt the modified encrypted image  $\overline{\mathbf{C}_M}$  and match it with  $\mathbf{M}$ ), however, both these operations require that the sharer exchanges with her the encryption key material (i.e.,  $K$  and  $\mathbf{R}$ ). Obviously this would allow the recipient to decrypt the unmodified original image  $\mathbf{C}_I$ , hence the choice for making the sharer to instead share the encrypted modified image  $\mathbf{C}_M$  together with  $\mathbf{M}$ .

### Choosing the Underlying Homomorphic Encryption

Building and implementing this scheme in practice is a challenging task. One of the challenges arises from the choice of the underlying homomorphic cryptosystem to be used. Most of the existing homomorphic cryptosystems are partially homomorphic, i.e., they support either the operation of addition or multiplication but not both. For example, both RSA [189] and the El Gamal [190] cryptosystems support the operation of multiplication only, while the Paillier system [191] supports the operation of addition only.

Gentry [192] was arguably the first to propose a fully homomorphic encryption scheme, i.e., a cryptosystem that supports both addition and multiplication on ciphertexts. His scheme is based on ideal lattices: after fixing a ring  $R$  and a basis  $\mathbf{B}_I$  for an ideal lattice  $I \subset R$ , it picks a public key  $\mathbf{B}_J^{pk}$  as well as a private key  $\mathbf{B}_J^{sk}$ , which are the basis of some other ideal  $J$ , such that  $I + J = R$ . Encryption of a plaintext  $p$  is computed as follows:  $c \leftarrow p + I \bmod \mathbf{B}_J^{pk}$ . In decryption,  $p \leftarrow (c \bmod \mathbf{B}_J^{sk}) \bmod \mathbf{B}_I$ . Gentry's scheme encrypts a single bit



at a time, and in theory it can be used to construct the aforementioned pixel-wise image encryption from equation 4.4. However, his scheme is too complex (especially for a low-powered camera device) to be implemented in practice. According to Gentry's implementation notes [193], even when choosing a very low security parameter for the encryption function (i.e.,  $n = 512$ ) it takes about 30 seconds to encrypt a single bit (using a machine with a 64-bit CPU and with a large memory).

In recent work [194], Yang et al. present a fully homomorphic encryption system based on Gentry's scheme, tailored for encrypting images. They optimize Gentry's scheme by (i) changing it from a public key encryption to a symmetric encryption (in order to use shorter keys without sacrificing security), and (ii) encrypting a byte (i.e., a pixel) instead of a single bit at a time. According to their implementation notes, encrypting a 1-megapixel image (using a machine with a double-core CPU at 3.1 GHz and with 4 GB of RAM) finishes in about 10 seconds.

Yang et al. have managed to cut down significantly the required resources (both for processing and storage). Nevertheless, their fully homomorphic image encryption scheme still remains suitable only for powerful server machines. There is much work to be done in order to execute the second variant of our protocol in a low-powered camera. Unlike this variant, our previous protocol variant (from section 4.4.1) runs seamlessly and efficiently on embedded cameras, as we will see from our implementation notes reported in the following sections.

## 4.5 Implementation

We implemented the proposed data sharing and data verification protocol (variant 1) to better assess their practical feasibility when run on our low-powered camera. Our implementation consists of two main services: *a beacon transmitter* and *a beacon scanner*. The transmitter service generates a public/private key-pair every time it starts and then broadcasts the public key. These keys are generated with Elliptic Curve Cryptography (ECC) using a curve over a 256 bit prime field. The private key is kept secure by encrypting it using the Storage Root Key (SRK) available from the camera's TPM (see section 3.3.3 for more information on this process). The transmitter also generates and broadcasts an access token at a specific frequency. Tokens are generated using the procedure from Protocol 2 (variant 1, see section 4.4). The scanner service listens for nearby beacons of the same type. Once it detects a beacon, it will store both access tokens and captured public keys in the camera's internal storage.

BLE Header (10 bytes)	Beacon Header (2 bytes)	TX Power Reference (1 byte)	Actual Data (25 bytes)
--------------------------	----------------------------	--------------------------------	---------------------------

Figure 4.7. The beacon protocol data unit.

The implementation uses Bluetooth Low Energy (BLE) as the underlying technology for sending and receiving both keys and access tokens. Specifically, we use the Alt Beacon Library<sup>5</sup> to create a BLE beacon-like device. BLE version 4 allows up to 28 bytes (excluding the BLE header bytes) for manufacturer specific data in the advertisement packet. Three bytes are consumed by the beacon library (2 for specifying a mandatory beacon identifier, and one for a mandatory power reference value<sup>6</sup>) leaving 25 bytes for actual data (see Fig 4.7 for the beacon layout). This size limitation restricts our choice of key and token length. As a result, the system generates tokens  $token_{it}$  of exactly 25 bytes (200 bits). Since 200 bits is not sufficient for public key transmission  $Pub_{ij}$ , we instead broadcast a 25 byte identifier for a key uploaded to a known *key server*; peers can use the identifier to retrieve the key from this known server.

While in sharing mode, our system alternates between session public keys  $Pub_{ij}$  and access tokens  $token_{it}$ . We use the 2 byte Beacon Header field (see Figure 4.7) to differentiate between key and token data. Since tokens are refreshed much more frequently than public keys, the system is configured to broadcast token beacons with a higher frequency than keys. However, a too low frequency of key distribution will result in delayed “registration” of peers. We experimented with different transmission schedules in order to find the best ratio between public key and token transmission frequencies and report results in section 4.7.

## 4.6 Security Analysis

We take a two step approach for the security analysis of the proposed system. At the outset, we analyse the data sharing system concerning its ability to prevent memory oversharing and user tracking. We then examine the image modification protocol (variant 1) with regard to the detection of hidden image modifications.

<sup>5</sup>see <https://altbeacon.github.io/android-beacon-library/>

<sup>6</sup>The TX Power Reference Value is a pre-measured signal strength at 1m distance from a beacon, which allows a recipient to estimate the actual distance of the signal sender.

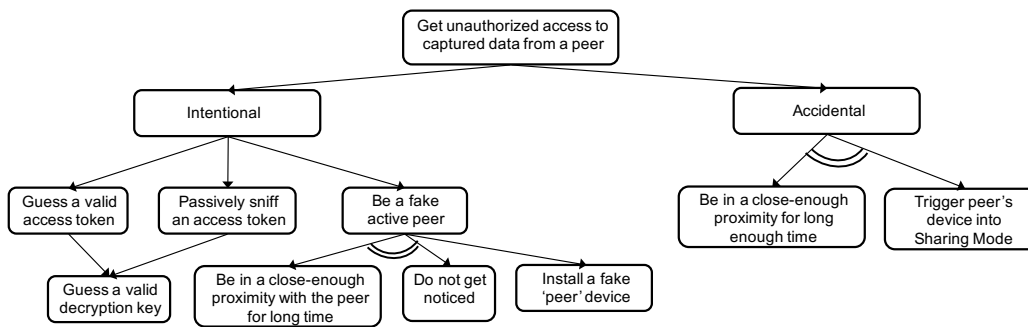


Figure 4.8. Attack tree visualizing the different ways for obtaining unauthorized access to a peer’s captured data. Tree nodes represent sub-actions that must be accomplished for the attack to succeed. Double half-arcs represent an AND relationship among nodes indicating that all such node actions have to be achieved in order for the parent action to succeed.

#### 4.6.1 Experience Sharing System

One of the goals of the proposed data sharing scheme is to protect the captured memories of a user from unauthorized access as well as prevent tracking the user’s location. As a result, we primarily try to minimize “oversharing”, i.e., the accidental inclusion of a peer’s device into our captured data stream, without actually being part of our experience (e.g., a shared meeting).

A second goal is to minimize our tracking envelope. Tracking is a prominent risk in our system since a device advertises its willingness to share captured data by broadcasting announcements (i.e., public keys and access tokens). An attacker could simply listen passively for such advertised information and thereby track a user’s location. To counter this threat, a device does not send out announcements all the time. As described in section 4.5, we envision that the system announces itself only when it detects the *presence* of an appropriate peer. The design of such peer detection is outside the scope of this thesis – we previously described several approaches for peer detection based on a social detector, see “*Smart Memory Beaconing Based on Improved Peer Detection*” in section 4.5. Moreover, both public keys and tokens are frequently updated to prevent an adversary from associating them to a specific user.

The system should avoid oversharing (i.e., allowing non-authorized parties to access captured data) by granting access only to peers who were present and engaged with the user at the time of capture. Figure 4.8 depicts the different ways an unauthorized person can try to get access to a user’s captured data. An attacker can try to construct a data URL (one that leads to actual data of a user)

by guessing a valid access token or passively sniffing for tokens being sent out. Either way, the attacker then has to also guess a valid decryption key (i.e. user's private ECC key) in order to successfully get a valid data URL. Guessing both an access token of 25 bytes (200 bits) and a valid decryption key (the private part of a 256 bits ECC key-pair) is extremely unlikely. Even if an attacker manages that, this would only allow her to get a small set of data points and not compromise the whole system, as other data are protected by different access tokens and eventually different encryption keys.

While our protocol against oversharing can protect from non-participants getting data of an experience, it cannot protect against their collusion with legitimate participants (i.e., one trusted participant passing on the recordings the received). To address this one can consider technologies such as "Digital Rights Management" (DRM), usage control models [195, 196], or watermarking to help identify the source of a leak should it, for example, surface on the Internet. Implementing any of these solutions is out of scope of this thesis. However, to counter such peer collusion under a certain extend, we propose a tangible control interface. The interface allows one to quickly stop recording and sharing through physical gestures. This could allow users to react on time (e.g., by stopping the recording) should they, for example, judge that what they are currently experiencing is too sensitive to be captured in the first place.

Furthermore, our solution does not guard against secret recordings – either out-of-band (e.g., a hidden camera or microphone) or through a "hidden" (fake) peer device that exploits the system by pretending to be a peer capture device. Countering hidden cameras or microphones is beyond the scope of this thesis. However, our tangible interface can be again used to counter fake peers under certain conditions (a formal meeting in an office setting). The interface is also equipped with a low-powered display which the number of connected peers that one is sharing data with. This could allow meeting participants to detect data over-sharing through a quick head count mismatch. We present the design of such interface and evaluation results of a user study in chapter 5.

## 4.6.2 Image Verification Protocol

In order for an attacker to change any tile's content unnoticed, two options exist: (1) to manipulate the image before the tile's hash is computed, or (2) to manipulate it in such a fashion that the tile's hash does not change. The first option is ruled out by virtue of the secure camera hardware. Here, the camera's firmware is attested by the its trusted computing platform TPM, so changing the camera's principal operations should not be possible. The second approach requires the

attacker to perform a *second pre-image attack* on the underlying hash function. Given a secure hash function (we use SHA3-256 in our implementation) this should be equally infeasible.

We also need to ensure that the recipient cannot uncover the original contents of a tile, based on the shared information. To achieve this, a recipient has to perform a *pre-image attack* on the hashes of the obfuscated tiles: given the hash  $h$  of a tile that is blocked in the modified image, find a value  $t$  such that  $H(t) = h$ . Given that the hash function concatenates the tile contents with both the tile indices  $i, j$ , and a salt, even identical tiles in the original image should hash to different values. A brute force attack is thwarted by the large search space: a pixel is composed of three bytes, one byte for each RGB color. Trying all pixel colors has a time complexity of  $256^3 = 2^{24}$  per pixel, hence iterating through a single tile would take  $(2^{24})^{mn}$  time, where  $m \times n$  are the tile dimensions in pixels. Even the smallest tile size of  $5 \times 5$  pixels that we evaluated our system on (see section 3.4.5) would require  $(2^{24})^{25} = 2^{600} \approx 4 \times 10^{180}$  operations for a single tile.

While brute force is not an option for a malicious recipient peer, preimage attacks can be realized through a more sophisticated approach based on pre-computed hash tables, also known as a *rainbow table attack*[197]. Such attacks tend to reduce the time complexity of a brute force attack on the cost of increasing the storage complexity. A malicious peer can compute a very large rainbow table once (or even use existing tables shared by the community) and then reuse it to disclose information from every obtained image. Though a rainbow table attack is probabilistic and there is no guarantee that it will always find a valid hash input, however, even assuming that such an attack would always work, the use of a different salt for every other image prevents the reuse of existing tables: a recipient peer needs to compute a new table for every new image. By carefully choosing the salt length one can make this attack unpractical: for a 32-bit salt the malicious peer needs to compute  $2^{32} \approx 42$  billion different rainbow tables.

## 4.7 Evaluation

We conducted several measurements to validate the practical feasibility of the proposed system. Specifically, we measured the performance of the BLE beacon protocol and its discovery range, the implementation runtime overhead as well as the camera's energy consumption.

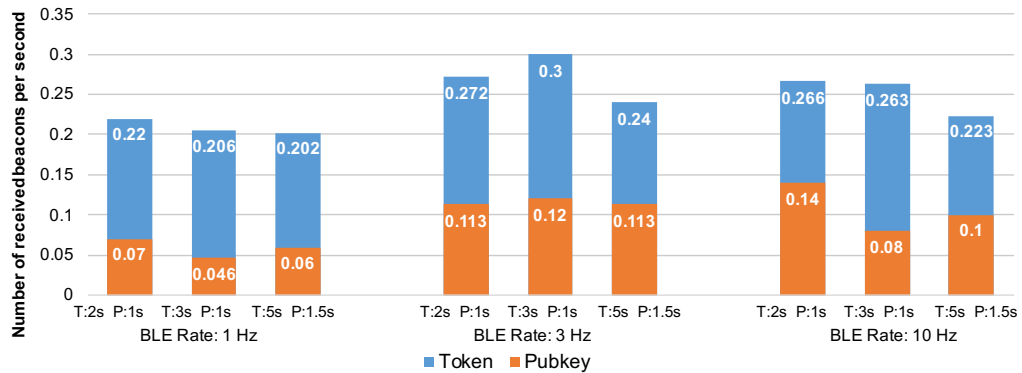


Figure 4.9. Average beacon reception rates per second when using three token-pubkey ratios (all given in seconds) 2.0-1.0 / 3.0-1.0 / and 5.0-1.5, using three BLE announcement frequencies 1 Hz; 3 Hz; and 10 Hz.

#### 4.7.1 Beacon Reception Rates and Proximity Range

We initially measured the performance of the underlying BLE beacon transmission. More specifically, we simulated a three-party social encounter, measuring how long it took for the peers' devices to "register" each other once they get in proximity and how reliably a device can pick up other peers' access tokens. Due to hardware limitations, we were not able to trigger the BLE radio of the Raspberry Pi 3 B+ board to both transmit and scan for BLE beacons at the same time. Consequently, for this test, we used a smartphone that was capable to perform this functionality (Google Nexus 5X running Android 6). For this we developed a smartphone app that performs the core functions of the data sharing system, following the implementation details as described in section 4.5.

In our tests, we compared three different token-public key ratios (2:1, 3:1, and 5:1.5, all given in seconds) over the three BLE announcement rates supported by Android 6 (1 Hz, 3 Hz, and 10 Hz). For example, in the first case (1 Hz rate) the system would send out 2 tokens in 2 seconds, followed by sending the device's session public key one within the next 1 second; at a rate of 3 Hz, this would mean 6 token packets (still taking 2 seconds) followed by 3 public key packets (taking 1 second).

Figure 4.9 shows the public key and token reception rates per second, averaged over the results of all 3 phones running with each configuration for a period of 5 minutes. Increasing the transmission frequency to 3 Hz improves reception rates, but a further increase to 10 Hz adds delay when simultaneously transmitting and scanning for packets. Thus, we use 3 Hz for the transmission rate.

With a 3 Hz packet transmission, the highest beacon reception rates (0.3 token/s and 0.12 keys/s) are achieved with 3 seconds of token transmission and 1 second of key transmission. Therefore, it may take up to 3 seconds before a device reliably receives a new token from another device; public key beacons will be picked up in 8.3 seconds. In practice, this means that once two devices are in range, it should take no more than 9 seconds (average 4.15 seconds) for them to “register” each other and, after sufficient dwell time  $t_{dwell}$  has elapsed, to add each other’s public keys to their respective encrypted URL uploads. Tokens should therefore not be updated more frequently than once every 3 seconds, otherwise a peer may miss a token (and thus be unable to access captured data for this period). These results were more or less dependent on the performance of the smartphones that we had, and clearly better results may be achieved with more recent or even future hardware. Furthermore, there is a trade-off between token update frequency and data over-sharing. The lower the frequency, the more data will be overshared and vice versa. One should choose the right update frequency depending on the envisioned privacy requirements.

While results show that our protocol performs reliably well with three peers, it is to be expected that the beacon reception rate would decrease as the number of peers would increase. This is due to packet collisions that can occur when many peer devices would transmit simultaneously. However, in this work we assume sharing to happen among relatively small encounters (i.e., groups of no more than 10 participants). We looked at prior work in order to understand what would be the implications on beacon reception rates from such group sizes. In one such work, Treurniet et al. [198], report a decrease of about 6% in packet reception rates when having 10 peer devices. This would add about half a second to the total of 8.3 seconds needed for a device (in the case of three peer devices) to pick-up the beacons from others. If we would account for additional other devices that can communicate through BLE (e.g., IoT devices, BLE beacons, etc.) the total number of BLE devices can easily be in the range of 20–30 at any time. Results from the work of Treurniet et al. show that with 30 devices, BLE reception rates can decrease up to 20%. This would introduce an additional latency of only 2 seconds, resulting in a total of about 10 seconds before peer devices running our sharing protocol can register each other. These figures suggest that our protocol would still perform well even in such moderately crowded BLE environment.

BLE has a maximum range of about 100 m, allowing signals to be “heard” even by devices which are not in a close enough proximity to be considered co-located peers. However, when transmitting with the lowest power level that our test devices could achieve (i.e., approximately -12 dBm), the proximity detection distances were lowered significantly (e.g., about 7 m in open environments

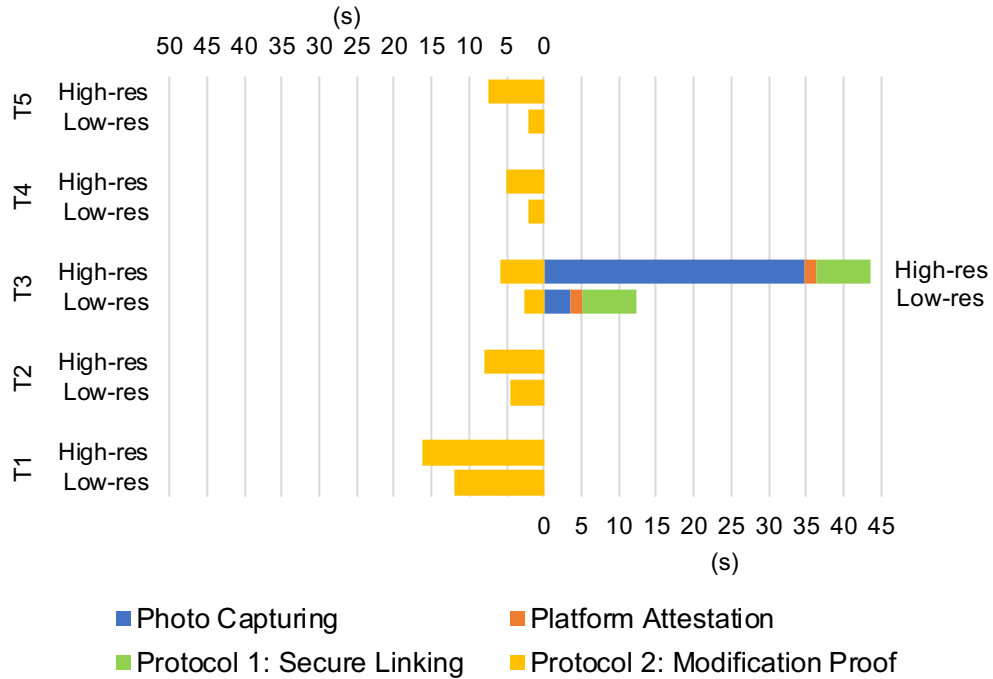


Figure 4.10. Implementation runtime overhead (in s). Measurements were conducted with low-resolution images (640×480 pixels) and high-resolution images (4096×3072 pixels). Protocol 2 was evaluated with five different tile sizes. For the low-res images we tested the following tile sizes in pixels (T1: 5×5; T2: 10×10; T3: 20×20; T4: 40×40; and T5: 160×160), and for the high-res images (T1: 32×32; T2: 64×64; T3: 125×125; T4: 256×256; and T5: 1024×1024).

without any obstruction, to about 3 m in office-like environments). These results provide satisfactory accuracy for reliably exchanging experience data in different scenarios. For instance, when recording images of a hiking activity, a distance of 7 m allows one to exchange experience data with a larger group of other hikers, while a distance of 3 m would be enough for exchanging data of a meeting with other attendees.

#### 4.7.2 Runtime Overhead and Energy Consumption

In additional tests, we evaluated the variant 1 of our proposed protocol for computing an image modification proof following the process from Figure 4.4. Tests were conducted using our TPM-enabled wearable camera, and with two different images sizes: a low-resolution image with 640×480 pixels (the maximum



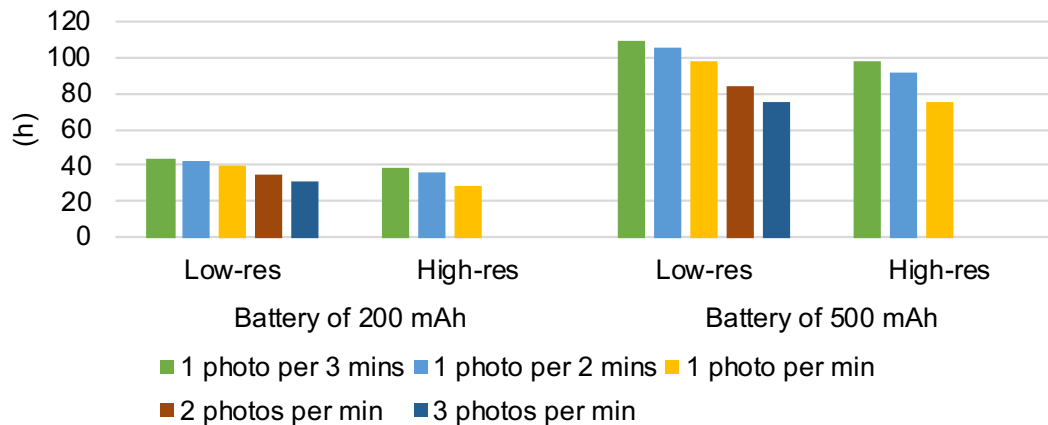


Figure 4.11. Estimated camera operational time when running our system with various photo capturing frequencies and powered with a battery of 200 mAh and 500 mAh, respectively. For Protocol 2 we used medium tile sizes (i.e., T3).

resolution of the installed camera module), and a high-resolution image with  $4096 \times 3072$  pixels. For the tests featuring a high-resolution image, we manually put such an image into the camera's internal storage. The protocol's code would then fetch the image contents directly from there and not from the camera's module. We furthermore measured the overhead of splitting the image into five groups of tile sizes. When selecting a tile size, we considered the trade-off between obfuscation granularity and performance efficiency. For the smaller image we started with a tile size of  $5 \times 5$  pixels and increased it up to  $160 \times 160$  pixels. We applied proportionally larger tiles to the larger image, from  $32 \times 32$  pixels up to  $1024 \times 1024$  pixels.

Figure 4.10 summarizes the runtime overhead (in seconds). For completeness, we also included the execution time of the other camera processes that were presented in Chapter 3 (i.e., capturing an image, generating a TPM platform attestation report, and adding the image to a secure link). As we can see from the figure, our proposed schemes work reasonably well on our camera platform. A low-resolution image is captured and processed in less than 25 seconds (3.5 s for taking the photo, 7 s for Protocol 1 and 12 s for running Protocol 2 with the smallest i.e., most processing intensive, tile size of  $5 \times 5$  pixels). Less than 60 seconds are needed for a high resolution image (34 s for taking the photo, 7 s for Protocol 1 and 16 s for Protocol 2 again with the smallest tile size  $32 \times 32$  pixels).

Finally we measured the energy consumption of the camera when running the proposed schemes. Again, these tests were performed using the camera processes (from capturing a picture, adding it to the secure chain, generating

a TPM-signed platform report, computing the image modification proof, and exchanging this it with co-located others). Energy figures were measured using the “Keweisi KWS-V20” USB power tester<sup>7</sup>. The average power consumption for both low-res and high-res images (with medium tile sizes of  $20 \times 20$  and  $125 \times 125$  pixels) is 2.1 mAh and 5.6 mAh, respectively. Out of these values, 1.06 mAh and 3.60 mAh were consumed by the Raspberry Pi 3 B+ module alone. From these measurements we did not derive exactly how much power was consumed by the actual BLE transmissions. However, prior work has shown the BLE consumes very little energy per bit transmitted and scanned [200, 198]. In this regard we expect the actual packet advertisements over BLE to account for very little overhead to the overall energy consumption figures.

Using these power measurements, we estimated the camera’s operational time (see Figure 4.11) with a small battery of 200 mAh (same as that of the Narrative Clip 2) and a bigger battery with 500 mAh. With a capture frequency of one low-resolution photo per minute, the camera can be operational from 40 hours (smaller battery) up to 100 hours (bigger battery) on a single charge. As for high-resolution photos, the camera can run between 30 hours and 75 hours.

## 4.8 Chapter Summary

In this chapter we investigated the possibility of seamlessly exchanging captured experience data among co-located users. The rationale behind such data sharing emanates from a technical limitation of wearable cameras, which may not always produce good memory cues [52], as camera lenses can be obscured by hair or clothes, or simply face the wrong way [11]. In a previous experiment [58] we observed that at least 25 percent of lifelogging images captured by our study participants were occluded.

As a result, we highlighted two complementary data sources that can improve the quality of the produced memory cues. First, wearable cameras of others can offer a richer view than one’s own camera. For instance, while our own camera might fail to capture who is sitting next to us, the camera of the person opposite from us would. Secondly, the high-vantage point of infrastructure cameras allows them to capture comprehensive scenes, completely unobstructed [159].

We furthermore analyzed the potential threats that any such seamless data sharing would raise. Captured images will inevitably feature private information, which opens the door to accidental data spills. For instance, when capturing

---

<sup>7</sup>The “Keweisi KWS-V20” USB power meter that we used for these evaluation lists an accuracy error of up to 3%, as reported in [199].

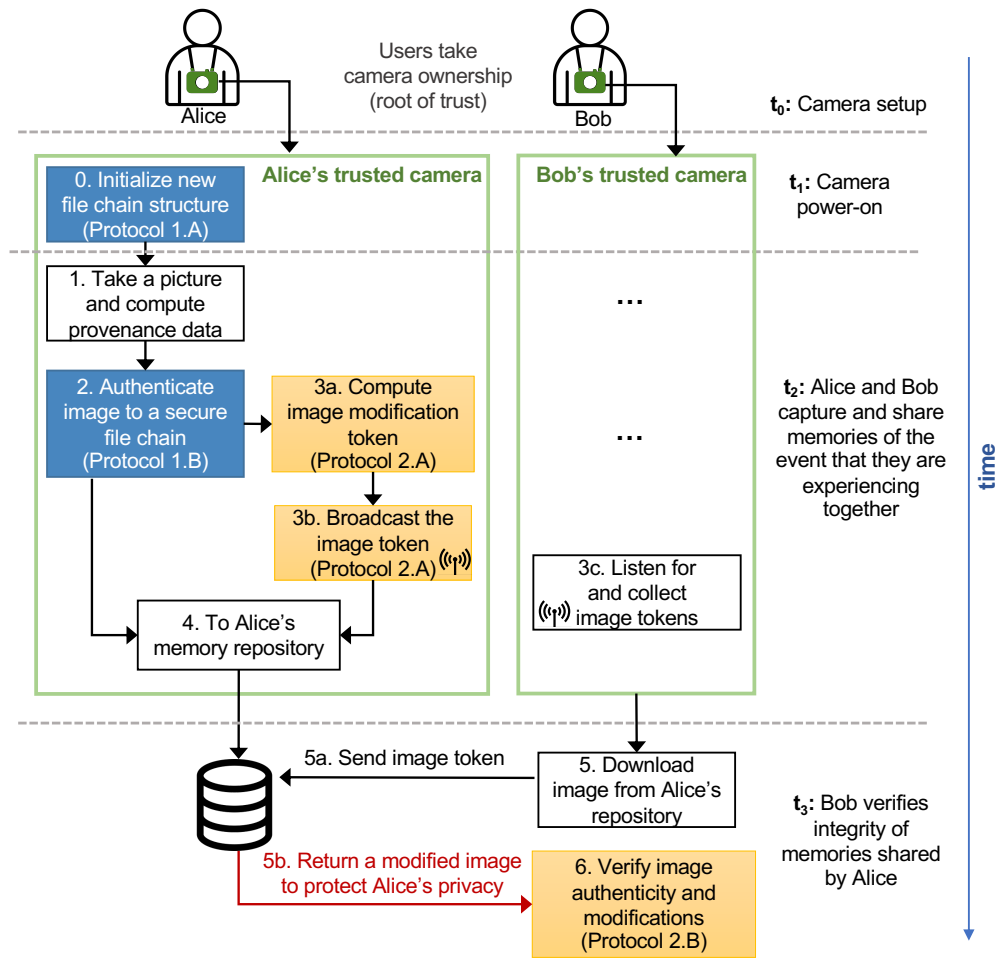


Figure 4.12. Overview of the event flow of using our system for securely capturing, storing, as well as securely sharing experience data with co-located others.

a work meeting one can mistakenly share data with the users who simply pass by the meeting room or share data of a different meeting event. Furthermore, any data that we receive from others may endanger our overall memories of that event. Malicious others can share with us intentionally fabricated images that do not truly reflect the experience. Reviewing such falsified images can utterly reshape what we remember about the experience.

Therefore, in this chapter we presented an approach to enable a seamless but secure exchange of experience data. Our system is built on top of the results from Chapter 3 for the secure capture and storage of experience data. Figure 4.12 provides a complete view of our system by incorporating components presented in both Chapter 3 and this chapter.

Our system allows a user's camera to advertise its willingness to both share self-captured data and to acquire data captured by other co-located peers. For this reason, by means of the short-range BLE technology, users' cameras broadcast periodically updated access tokens and temporal public keys. The purpose of tokens is twofold. First, they represent a way of letting others know where to access data one is willing to share. Second, a token allows one to commit an image's content publicly without actually having to share the original image itself. One can then decide to obfuscate any sensitive information prior to sharing it with others. Using such tokens, recipients can verify any modifications performed on the original, unshared image (**RQ3**). On the other hand, a device can grant access to the data that it is willing to share to co-located peers only. This is achieved by encrypting the shared data with all peers' public keys. Consequently, only those who possess the valid private key will be able to access the shared data (**RQ2**).

Our tests confirm that the proposed scheme can reliably and efficiently run on a low-power camera device. When transmitting the radio-packets with the lowest possible power level, we were able to reduce the BLE detection range to 7 meters in open unobstructed environments and 3 meters in closed office-like spaces. Devices can reliably exchange access tokens and public keys with a maximum rate of 0.3 tokens/s and 0.12 keys/s, respectively. Furthermore, the camera can compute a modification proof for a low-resolution image in about 12 seconds, while 16 seconds are needed for a high-resolution image. When including the runtime overhead of the other schemes from Chapter 3, a low-resolution image can be captured and processed in less than 25 seconds, whereas processing a high-resolution image takes about 60 seconds. When processing one low-resolution image per minute, we measured that the camera can be operational from 40 hours (with battery of 200 mAh) up to 100 hours (battery of 500 mAh) on a single charge. As for capturing high-resolution photos, the camera can run between 30 hours and 75 hours.

In this chapter we addressed the challenge of accidental oversharing of experience data. In the following chapter we investigate the possibility of empowering users with more controls regarding both memory the capturing and sharing.

## **Part III**

# **Memory Capture and Access Control**



## Chapter 5

# A Tangible Interface for Controlling Memory Capture and Sharing

As we have seen in Chapter 4, the ability to exchange captured experiences among co-located peers is a useful desideratum of memory augmentation systems. It can offer a more comprehensive capture of an experience, something that one's own wearable camera may not be able to match. However, despite such benefits, sharing self-captured images with others can at the same time seriously jeopardize users' privacy. In the previous chapter we focused primarily on preventing situations of accidental oversharing that can happen due to system errors, e.g., mistakenly considering a bystander as participant of a common event. As a result, the proposed system attempts to disseminate one's self-captured data only with peers that are co-present with the user. Ideally, no data will be shared should there be no co-located peers that the user is interacting with.

However, such an automated approach may not always offer the desired outcome. Let us consider several situations where users would benefit from having more control over the practices of capturing and sharing their memories. While an experience is being captured, there may be moments that a user would not want the system to record in the first place (e.g., discussing confidential matters in a meeting). In other situations, a user would want to capture the experience for themselves but would not feel comfortable sharing such data with others (e.g., while working in front of a computer). In yet other situations, a user may be willing to share data only with a particular set of other users, but would want to limit adding further peers. All these examples highlight one essential requirement: users should be able to continuously control and express their capture and sharing preferences of the event they are experiencing, as the event moves across different levels of sensitivity and privacy. What is more, in case that one forgets

to react in time, it should be possible for one to perform after-the-fact deletion of such data, as soon as it is noticed. Not to mention that access should be revoked if the “problematic” data was already shared with others.

Therefore, in this chapter we set out with the goal of providing a solution that features a more balanced tradeoff between the benefits of sharing personal experience data and protecting users’ privacy. To this end we developed MemStone, a prototype of a tangible user interface (TUI) that allows users to control access to (and the sharing of) captured memories in-situ. We conducted a user study with 20 participants with the goal of investigating the suitability of a set of gestures to control data capturing and sharing, as well as comparing the usability and efficiency of MemStone with a more “traditional” mobile app user interface. Our study included an open-ended discussion session to better understand users’ perceptions of such tangible interface. In this chapter we describe MemStone’s design and functionality, report on the results of the user study, and conclude with discussing the implications of our results and outlining directions of future research. The work presented in this chapter addresses the first par (underlined below) of the following research question:

- **RQ4:** What interfaces and policy-based access control models can we use to exercise control over data capture as well as to prevent the disclosure of private and sensitive information when sharing experience data?

*Parts of this chapter are based on the following publication:*

- **A. Bexheti**, A. Fedosov, I. Elhart, and M. Langheinrich, “Memstone: A Tangible Interface for Controlling Capture and Sharing of Personal Memories,” in *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI ’18. New York, NY, USA: ACM, 2018, pp. 20:1–20:13

## 5.1 Related Work

The work in this chapter is mainly inspired by and based on previous research in: privacy issues with visual lifelogs, techniques for enhancing privacy of such data, and gesture-based interactions.



## Privacy Issues with Visual Lifelogs

Visual lifelogs can be the basis for highly effective memory cues. Thanks to body-worn cameras, which allow us to seamlessly capture continuous logs of our daily experiences, visual lifelogging lays at the sweet spot between memory recall and ease of capture. However, previous studies have found that such unobtrusive capture can infringe both user and bystander privacy. Clinch et al. [58] have conducted a multi-day experiment where they provided all their participants with a wearable camera which continuously captured users' working activities. They found that such cameras repeatedly captured participants' computer screens and phones. As a consequence, they observed that such capture of private spaces as well as the presence of specific objects in images made users concerned about their privacy.

Similar privacy concerns with images showing specific objects or taken at particular locations, but also portraying other known people, bystanders, or user activities, have also been observed by other studies [97, 105, 201]. Price et al. [202] noticed that users are less concerned when sharing images with a group of other lifeloggers than with non-lifeloggers, further suggesting that this could re-define what a private space means when lifelogging in a group. In another work, Adams [115] proposes a privacy model that considers *image receiver* and *purpose of usage* as additional factors that influences sharing decisions. All these studies confirm the privacy challenges of visual lifelogging and highlight the need for techniques for privacy-aware data capture and data sharing.

## Privacy-enhancing Techniques of Visual Logs

Several solutions have been proposed for regulating access and controlling data sharing [18], however, they usually require active user input in order to specify fine-grained access control and privacy policies. Due to the large volume of captured experience data, this would be a cumbersome process for lifeloggers [58], leading to mistaken disclosures ("misclosures" [111]). For instance, a Narrative Clip camera produces 120 pictures in one hour, or 1'500 in a day. One option is obviously not to capture all that information in the first place, but the challenge is that one cannot foresee which data might be a valuable memory trigger.

Prior work has made attempts to automate to some extent such decision efforts by designing algorithms that can understand both capture context and captured data. For example, Fan et al. [203] propose a mobile-based technique that stops lifelogging capture when it detects that a user is in a restroom. Moncrieff et al. [204] leverages background audio, but also other sensors, to determine

the context in surveillance scenarios running in private environments, such as smart homes. Based on the inferred context, the system will activate a predefined privacy policy and enforce it using a combination of data hiding techniques. Other approaches rely on computer vision algorithms to study the captured images themselves and flag those that contain specific places [80], particular objects [112], computer screens [79], or even images that portray activities [113].

While all these (semi-)automated solutions can potentially improve user privacy, however, they may reduce the utility of a memory augmentation system. As Adams [115] notes, privacy issues related to captured experiences often rely on users' implicit assumptions of its usage and intended receiver, and as such they can vary with person and context [58]. For instance, an image that can infringe a user's privacy because it contains a computer screen can be *the* strongest memory cue; or a user might want to share such computer screen image with only a particular other user that she trusts more. In contrast, we propose a solution based on (manual) in-situ user input. In-situ controls offer greater flexibility to users and allow them to react in real-time based on their impressions of the context, but still keep user involvement lower than post-hoc solutions. Hoyle et al. [110] also confirm that lifeloggers prefer in-situ control more than manual post-hoc filtering. Ultimately, our in-situ control can be used side-by-side with an automatic control approach and complement it.

## Gesture-based Control Interfaces

Prior research has explored the opportunities of controlling virtual information using objects from the physical world. Fitzmaurice et al. [205] propose a technique for manipulating digital data using graspable wooden blocks, with the goal of augmenting traditional graphical user interfaces. The *tangible bits* vision by Ishii and Ullmer [206] aimed at bridging the gap between digital bits and graspable objects, where objects from the physical world would both manipulate and visualize digital content. Similarly, Fishkin et al. [207] present the paradigm of embodying physical manipulations to computational devices, so that the device's physical body becomes also its interface.

Other research has particularly focused on box-shaped physical interfaces. For example, Rekimoto and Sciammarella proposed ToolStone [208], a cordless tangible interface controllable by physical manipulations. ToolStone would be operated by users' non-dominant hand and would complement the traditional computer input device (i.e., the mouse) in various applications where such a bimanual interface could be appropriate, e.g., choosing a color from a palette; zooming, scrolling, or rotating contents; controlling a virtual camera, etc.

Sheridan et al. [209] explored the *affordance* of a cube as control interface. Through a user study they developed a classification of 16 distinct gestures (or “non-verbal dynamics”) that users performed with a cube, such as placing the cube in a particular place or position, turning it, rotating, tapping, shaking, squeezing, or fiddling with it, etc. Van Laerhoven et al. [210] built such a cube that embodies gesture recognition and showed how it can be used as an input device for desktop applications involving selection and navigation operations.

All these studies show the feasibility and psychological affordance of box-like interfaces, however they mostly focus on applications for extending traditional input devices or GUIs. Moreover, in these works the box was used only as an input device and was not utilized to also provide feedback back to the user. In our work we adopt the concept of a box-shaped interface and apply it in a scenario that goes beyond extending conventional input devices, i.e., allowing one to control and observe *how*, *when* and with *whom* one’s lifelogging devices are capturing and sharing data that constitutes one’s memories.

## 5.2 MemStone Interface

We designed MemStone inspired by a mix of both practical and theoretic knowledge. More specifically, we were motivated by the ToolStone interface from Rekimoto and Sciammarella [208], particularly by its design and shape. We furthermore grounded MemStone’s design in a set of theoretical design principles regarding interactive products [206, 211, 212].

### 5.2.1 Design Principles

Our design of MemStone was strongly influenced by the design principles from Norman, presented in his seminal book “The Design of Everyday Things” [212]. Rogers et al. [213] provide a nice overview of Norman’s main principles, which we briefly restate here:

- **Visibility:** When creating interactive objects it is important that its functions and components are easily visible to users. Norman stresses the importance of *visibility* through an example of car controls. The controls for most operations (e.g., headlights, horn, indicators, windshield wipers, etc.) are positioned in such a way that it becomes easy for the drivers to find and use them. Usage is hindered when we cannot understand how to operate a device. For instance, in most automatic faucets it is not clear that they are triggered by motion sensors, thus it becomes difficult to operate them.

- **Feedback:** Upon a user performing an action, the device should follow with an immediate and synchronized feedback information. This not only informs the user what action have they just triggered, but also whether the action has been accomplished or not. Rogers et al. [213] amplify the importance of this point by imagining how some everyday situations would be like without *feedback*. For instance, if the guitar or the pen would not send any immediate feedback, there would be an unacceptable delay before any sound was produced or any line written on a paper. This delay would render these devices almost useless. Feedback can take different forms, such as *audio*, *visual*, *verbal*, *tactile*, or various combinations of these.
- **Constraints:** Sometimes it is desirable to restrict some interface actions from being triggered. This can prevent users from performing a wrong action, hence reducing chances of mistakes. As reported by Rogers et al. [213], Norman classifies the different ways how this can be achieved into three categories of constraints: physical, logical, and cultural. A physical constrain prevents an undesirable physical movement on an object. For instance, when inserting a 3.5" computer disk into a drive, the drive's physical design prevents putting the disk in the wrong way. Not all physical constrains do offer a good solution. Probably the most familiar example of a bad physical constrain is found in the USB type A interface. The paradox is that no matter how hard you try, almost always you fail to get it right on the first try. The rationale behind an unflippable design was stirred by cost factors – a flippable design would require twice the wiring, and hence higher costs<sup>1</sup>.

On the other hand, logical constrains rely on common semantical knowledge to understand how a device works. A typical example of such constrains is disabling menu options when they are not appropriate for the given context and task.

Cultural constrains rely on universally accepted and learned conventions. For example, using the red color for warning, a triangle icon for play, alarm sound for danger, etc.

- **Mapping:** This concept is about making the connection between controls and their actions. Good examples of such mapping is found in the “up” and “down” arrows of computer keyboards, or the sequence of the buttons of an MP3 player (i.e., with the play button being in the middle, while the rewinding and fast forward buttons appearing on its left and right).

---

<sup>1</sup><https://www.cnet.com/news/the-reason-why-you-always-plug-in-a-usb-wrong/>

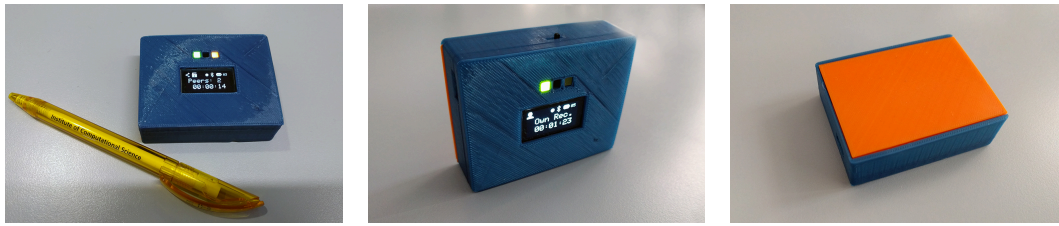


Figure 5.1. Overview of the developed MemStone prototype that allows users to in-situ control the capture and sharing of personal memories through a set of five physical gestures. MemStone’s front side (left) features a central screen and two LEDs that describe in detail its active operation. MemStone has a different colored back side (right) that allows users to see its position even from distance and denote its action.

- **Affordance:** This refers to the act of making it obvious how to use an object by perceiving its attributes. For instance, the shape of a door handle tells whether to pull or push it in order to open the door. Similarly, a computer mouse affords clicking, a virtual scroll bar affords moving up and down, a button affords clicking, etc.
- **Consistency:** When designing interfaces, it is crucial to have similar tasks be performed by similar operations or rules. For instance, the left mouse button is always used to select different elements in an interface, or the mouse scroll button will always vertically shift screen contents. Consistent interfaces are generally easier to learn and use.

### 5.2.2 Interface Description

MemStone (depicted in Figure 5.1) is a rectangular-shaped 3D printed box (measuring  $67\text{ mm} \times 52\text{ mm}$ ); augmented with an embedded computing platform (NodeMCU ESP8266<sup>2</sup>), an accelerometer sensor, a vibration mini motor, BLE radio, and a lithium battery. In principle it can communicate directly with a range of capture devices (e.g., body-worn cameras, smartglasses, audio recorder bands) in order to allow users to in-situ control, with simple physical manipulations, what memories such devices can capture and share with co-located others.<sup>3</sup>

Its shape (rectangular) and visual appearance (bi-colored) aims to allow users to easily understand its current operation (to some extent also its available ges-

<sup>2</sup><http://www.nodemcu.com>

<sup>3</sup>The current prototype of the MemStone focuses only on gesture recognition but it does not yet implement a protocol for communicating with other devices.

tures at a glance. The front side (see Figure 5.1–left) has a central screen (with a resolution of  $128 \times 64$  pixels and a diagonal size of  $33\text{ mm}$ ) that provides feedback on the system’s current mode of operation. Specifically, the screen shows several aspects of the active action, such as elapsed time of current data capture (if active), the number of peers one is sharing data with (if any), if newly appearing peers are allowed to join the sharing session, and the device’s remaining battery level. The front side has two additional LEDs that also give details about the current operation. The LED on the left will signal the user when the experience is being captured while the other LED will indicate that this data is being shared. The back side (see Figure 5.1–right) has a different color than the other sides to allow the user, as well as other co-located people, to see the device’s state from a distance and thus allow them to note its active operation without having to closely look at its screen.

### 5.2.3 Gestures and Control Actions

Starting from the privacy challenges presented at the beginning of this chapter, we have derived five different aspects that one could control when capturing and sharing experience data. Each such control action can be executed by performing a particular physical gesture using MemStone, as shown in Figure 5.2. For the gesture selection we were in part inspired by the work from Sheridan and her colleagues [209]. Through a user study, Sheridan et al. explored the natural affordance of a cube-shaped device and came up with a classification of 16 physical gestures that a cube affords. Starting from their result, we selected 5 such physical gestures that we believe offer a good match to the actions for controlling the practice of experience capture and sharing:

- The *face-down* gesture (Figure 5.2-A) stops both *data capture*, sourced from any of the user’s capture devices, and *data sharing* with other co-located people. This way a user can let others know that she does not record anything herself. Whilst this may also signal that a user does not want others to record her, the MemStone cannot control the capture operations of other users.
- By putting the MemStone device *face-up* (Figure 5.2-B), the user triggers her data capture from any of her lifelogging gear. In addition, the user informs co-located peers that she is recording and expresses her willingness to exchange data with them. Sharing commences automatically with all other peers that have similarly positioned their device face-up.

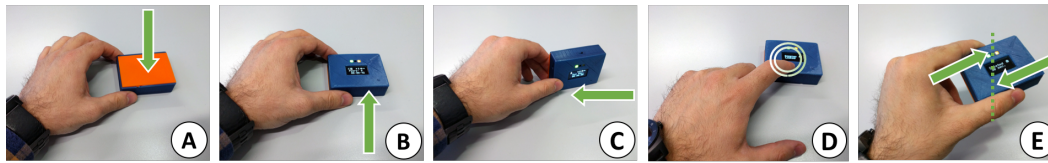


Figure 5.2. The current MemStone prototype supports five physical gestures: (A) *face-down*; (B) *face-up*; (C) *stand-on-side*; (D) *double-tap*; and (E) *shake*.

- In order to capture data from one’s own recording gadgets but inform others that one does not want to share data of that particular moment, one places the MemStone in a vertical *stand-on-side* position, facing oneself (Figure 5.2-C). Any active sharing session with any peer stops immediately.
- *Double-tapping* the MemoryStone (Figure 5.2-D) “locks” the data exchange with the current set of co-located people and prevents any further peers from joining (though peers leaving will still be removed from the common data exchange). A subsequent double-tap will remove this lock. A similar access control mechanism, but using the metaphor of a virtual wall, has been proposed by Kapadia et al. [214].
- The *shake* gesture (Figure 5.2-E) allows the user to delete the last 30 seconds (this is configurable) of captured data. By repeating this gesture the user can delete data captured for longer periods.

Gesture recognition is based solely on data obtained from a 3D accelerometer sensor. Triggered actions are confirmed with the help of distinct vibration patterns as feedback. For the lock/unlock action, two different vibration patterns will signal the user the current lock state after the initiated change.

#### 5.2.4 Envisioned Usage Scenario

To better illustrate the vision of using the MemStone’s concept as a tangible control interface for memory augmentation systems, consider the following *meeting capture* scenario, as shown in Figure 5.3.

Team Alpha, with its members Bob, John, and Alice have recently decided to use a memory augmentation system during their weekly meetings. During one meeting, Bob and John use a body-worn camera that automatically captures an image every 30 seconds. Alice uses a set of smartglasses that, similarly to the cameras, capturing 2 photos per minute. Bob also brought a wristband that captures the last 30 seconds of audio with a single tap. Each device uploads the

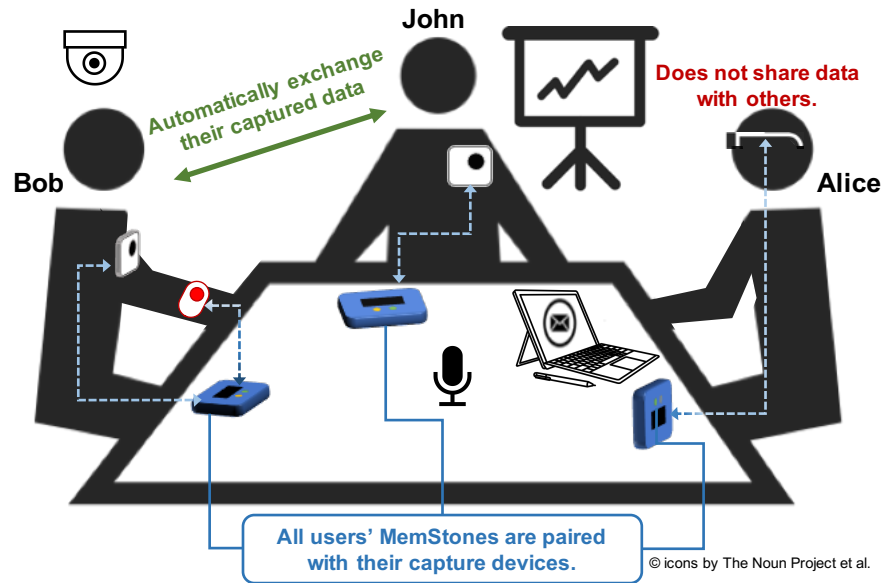


Figure 5.3. Illustration of the envisioned usage scenario of MemStone.

data it captures to the respective user's memory repository, where such data is eventually processed in order to generate memory cues, that will help them be more prepared for their next meeting.

All team members use the MemStone device to control their meeting capture and sharing practices. In the scenario in Figure 5.3, Bob and John have positioned their MemStones *face-up* notifying others that they are capturing this part of the meeting, as well as expressing their willingness to exchange data with their peers. As a consequence, their memory augmentation system would access images captured from each other's cameras, in order to improve the quality of the generated memory cues. John is particularly interested in obtaining images from Bob's camera, as Bob's camera occasionally captures the whiteboard (which is not the case with John's camera). In addition, John would also obtain audio snippets recorded by Bob's wristband. Alice, on the other hand, fears that her glasses might capture some sensitive information from her laptop's screen, which she has in front of her. She decides not to share any data with the rest, but only records for herself. To do that and also to inform others that she is not willing to share anything, Alice has put her MemStone on the *stand-on-side* position.

Note that even though Alice has decided to not share her captured data, John and Bob may very well have included Alice in their peer sharing activity. Whether or not sharing requires reciprocity (i.e., a user that stops sharing will



also be barred from receiving other shares) depends on each user's preferences – the protocol cannot enforce that peers actually share anything. If Alice continues to send out her public key and if John and Bob have not locked their session, her key will be included in the broadcast encryption. Similarly, John may signal that he is willing to share but eventually decide not to make any captured data available at the token address that his system broadcasted – Bob is unable to ensure the future sharing behavior of John. Note that while this seems to encourage (i.e., pretending to share in order to receive data from peers while not sharing anything) such behavior will in most cases eventually be sanctioned by social conventions: Bob will soon realize that John never actually shares anything so will be less inclined in the future to continue sharing with him.

In this scenario, we envision that MemStone devices will be part of the meeting room, just as one finds a remote controller for the projector or whiteboard markers in such a room. Before the meeting starts, each attendee picks up a MemStone from a small receptacle at the entrance and connects it with their meeting capture gear (e.g., by simply physically touching those devices together [215]). The connection would remain active as long as the devices are in close proximity. After the meeting ends, users would return the MemStones to their receptacle or simply leave them on the table. Once a user leaves the room, the MemStone would disconnect from their capture devices and reverts back to an idle state.

### 5.2.5 Using MemStone to Control Infrastructure Sensors

In addition to data sourced from their personal devices, team Alpha members occasionally use the room's built-in capture infrastructure to also make recordings of their meetings. The room is equipped with a fixed-camera, a central audio recorder, as well as a smartboard that captures snapshots of its contents.

Meeting attendees could use their MemStone to implicitly control experience capture from such infrastructure devices. A strict privacy-aware approach would stop recording from any fixed sensor as long as there is at least one participant that has positioned their MemStone face-down (i.e., does not record for themselves and does not want to exchange any data with others, Figure 5.2-A) or also on a vertical stand-on-side position (records only for themselves without sharing any data with others, Figure 5.2-C). Another approach would be that recording is based on a majority decision, or even unanimity (i.e., infrastructure sensors stop recording only if *all* users disable recording). Alternatively, the room's capture devices could also be controlled by a designated "room-MemStone". Its operation would then need to be agreed on by all participants.

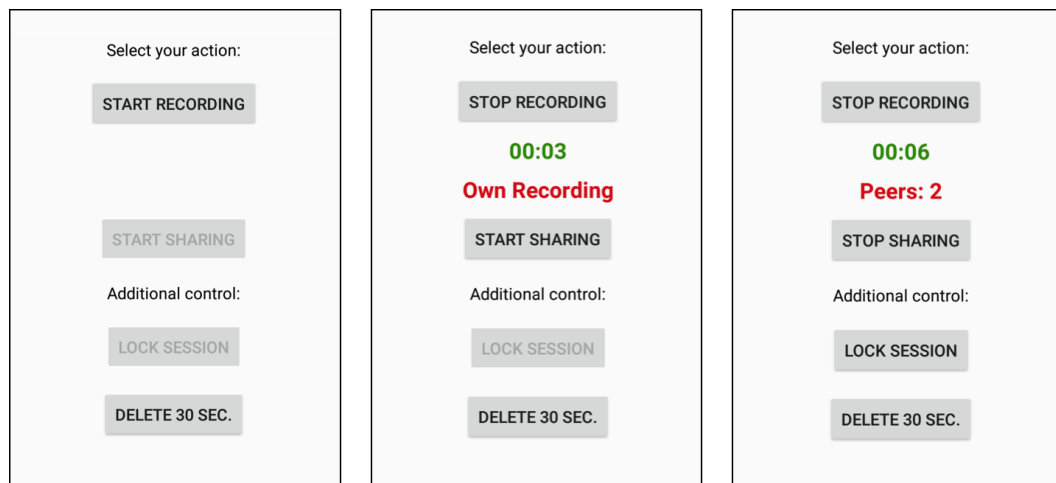


Figure 5.4. A phone-based interface for controlling data capture and sharing of daily experiences. Screenshots of different interface states. Left: initial state when no capturing or sharing is taking place; Middle: state of an ongoing data capture but not sharing; Right: interface’s state after activating data sharing with co-located peers.

In the remainder of this chapter, we consider a similar scenario (i.e., a meeting capture between two attendees in which data is sourced only from users’ wearable devices) in order to:

1. investigate how controlling data capture and sharing practices using physical gestures compares with a more traditional alternative, such as smartphone app; and
2. understand user perceptions regarding the MemStone tangible interface.

### 5.3 Phone-app as an Alternative Interface

Prior research in lifelog privacy [110] suggests that lifeloggers prefer in-situ control over end-of-day filtering for deciding which visual logs are shareable with others. Traditionally, such in-situ privacy control approaches are implemented as mobile apps running on a touch-based smartphone [216, 217]. While emerging body-worn devices, such as smartwatches or wearable glasses, could in principle become viable alternative platforms for designing such lifelogging control apps, the widespread use of phone apps means that a smartphone-based tool represents the most realistic alternative to our proposed tangible interface today.

We thus created a simple smartpone app that can act as a baseline alternative to MemStone. The phone UI, displayed in Figure 5.4, allows one to perform the same five control actions that can be performed using a MemStone. Note that since we wanted to compare the gesture-driven MemStone against the idea of using the smartphone as control interface (and not compare with a particular phone interface design per se), we went with a rather simplistic but intuitive phone UI: operated by buttons, each with a short label clearly describing its action. In the previous meeting capture scenario, users would install this app on their personal phones and pair it with their lifelogging gear.

## 5.4 User Study

We conducted a user study aimed to address the following research questions:

1. Is the proposed gesture-based interface usable for in-situ controlling data capture and data sharing in meeting capture scenarios?
2. Are the chosen gestures easily remembered even after longer time periods?
3. How does the use of gesture-interactions to control data capture and sharing perform against conventional (i.e., mobile app-based) user interfaces?
4. What are user perceptions on such gesture-based control interface within the context of memory augmentation systems?

For this study we recruited 20 participants (7 of them female) using snowball sampling [218]. Their age ranged from 22 to 63 years old, with an average age of 28.75 years ( $SD = 9.31$ ). They had different education levels, 4 of them had only a high-school degree, 9 were bachelor graduates and 7 had a masters degree. Most of our participants stated that they had an affinity for technology. All said that they used a smartphone several times a day and a laptop few times a week. No remuneration was provided to study participants.

### 5.4.1 Study Design and Procedure

We performed a comparative user study, employing a within-subjects design in a counterbalanced order, where each participant tried both interfaces in differing order (i.e., MemStone–Phone or Phone–MemStone).

We considered a meeting capture scenario between two people, similar to the previous scenario in Figure 5.3. To strike a balance between validity and repeatability, we prepared two different videos, each showing a meeting. We then asked

participants to watch those videos and pretend to be one of the attendees. Each such recorded meeting contained 10 tasks related to controlling capturing and sharing of memories within the meeting (note that a detailed description of both meeting videos and tasks is provided in the next section). Prior to watching a meeting video, participants were given either our MemStone or an Android mobile phone with the corresponding control app installed (accessible via a shortcut from the home screen). They were then instructed to use the respective tool for handling these control and sharing tasks. The phone was a Nexus 5X running stock Android 8.0 with only our app additionally installed. Note that the Nexus 5X features a default screen timeout of 15 seconds, which we did not change. While no screen lock (e.g., PIN) was setup, participants had to turn on the phone and then swipe up the lockscreen before they could interact again with the app after a screen timeout.

A session with a single participant lasted on average 60-70 minutes. At the outset, we briefly introduced the study, asked participants to sign a consent form, and to provide basic demographic information. Then, the session proceeded with the following stages:

1. Introduction of the vision for technology-driven memory augmentation, with a focus on how it could be applied to the envisioned meeting scenario.
2. A short demonstration of one of the interfaces followed by a trial session where participants try the different functionalities of the chosen interface.
3. Participants watch a video of a recorded meeting and use the assigned interface to control access to (and sharing of) their meeting memories.
4. Users fill a SUS questionnaire to express their perceived usability of the interface.
5. Repeat steps 2–4 using the other control interface and the second video.
6. Semi-structured interview, reflecting on the experience with both interfaces.

Participants' interactions with the interfaces (steps 2–4) were video recorded using a wide-angle camera. The produced video data was later used to compute the devices' efficiency and effectiveness for performing the specified control actions. The goal of the semi-structured interviews was to better understand the user experience with both control interfaces, and to also explore user perceptions on the proposed in-situ control interface. We recorded these sessions using a voice recorder, and then transcribed the recorded interviews. To analyze this data, we followed an iterative process, going back and forth between the data

Task	MemStone Action	Phone Action
<b>T1:</b> Capture and share with co-located peers	Face-up	Press “Start Recording” and then “Start Sharing” buttons
<b>T2:</b> Check elapsed time of current capture	Shown on the small display	Shown in app
<b>T3:</b> Stop capturing and sharing	Face-down	Press “Stop Recording”
<b>T4:</b> Capture only for oneself and do not share with others	Stand-on-side	Press “Stop Sharing” if currently sharing
<b>T5:</b> Capture and share with co-located peers	see T1	see T1
<b>T6:</b> Delete the last 30 seconds of captured data	Shake device	Press “Delete 30 Sec”
<b>T7:</b> Lock current sharing session	Double-tap device	Press “Lock Session”
<b>T8:</b> Verify with how many peers the system is sharing data	Shown on small display	Shown in app
<b>T9:</b> Stop capturing and sharing	see T3	see T3
<b>T10:</b> Capture only for oneself and do not share with others	see T4	see T4

Table 5.1. List of tasks and corresponding device actions.

and the researchers’ notes [219]. This technique helped us to organize participants’ feedback related to our TUI’s physical design, its interaction, as well as participants’ perceived usefulness of the system.

### 5.4.2 Recorded Meetings and Tasks

To better simulate a real meeting scenario, we created two videos depicting a meeting between an instructor and a teaching assistant, in which they discuss the progress of a course they teach together. In the video, both attendees capture the event using a wearable camera, while one of them also uses a wristband audio recorder. During the video, the attendees discuss both non-sensitive and sensitive issues (e.g., student grades), thus requiring several control actions on the capture and sharing system. Each time such a control point comes up, the “actors” explicitly announce this need for control (e.g., “Let me pause recording

<b>Video 1</b>	T1	T2	T3	T4	<b>T5</b>	<b>T6</b>	T7	T8	<b>T9</b>	T10
<b>Video 2</b>	T1	T2	T3	T4	<b>T9</b>	<b>T5</b>	T7	T8	<b>T6</b>	T10

Table 5.2. Task sequence for both videos.

for a moment!” or “Can you delete this part, please?”), yet without explicitly stating what exactly they have to do. A small icon at the bottom-right corner of the screen additionally triggers the need for action. To allow for different participant reaction times, the study administrator would then remotely pause the video and resume it only after the participant would perform an action, or would say that they would not know what action to perform. For each action, we measured participants’ reaction time and task completion rates.

In total, there were 10 tasks involving all five control actions that were presented previously (3 of them were repeated twice), plus two additional information gathering tasks related to the feedback given by the control device. Study tasks and the corresponding actions are summarized in Table 5.1.

The produced meeting videos are slightly different in order to minimize participants’ learning bias when they have to watch both videos for trying the two interfaces. In addition, the videos also feature slightly different task sequences (see Table 5.2 for comparison). During the study the videos and control interfaces were used together in a balanced order, that is, both videos were used in 10 runs with each interface.

## 5.5 Results: Interface Comparison

We report study results of the comparison of the two interfaces. Specifically, we report on the efficiency and effectiveness to perform the study tasks. We then look at the aspects of usability and learnability, as well as intuitiveness and enjoyment.

### 5.5.1 Efficiency and Effectiveness

Initially we compared both interfaces in terms of *effectiveness* (i.e., task completion rate) and *efficiency* (i.e., task completion time). For each device we collected data from 10 interaction tasks from a single participant, resulting in a total of 400 tasks performed from by 20 participants with both control interfaces.

For computing the task error rate, we only looked at the human error aspect and did not consider mistakes from the system (e.g., if the MemStone device

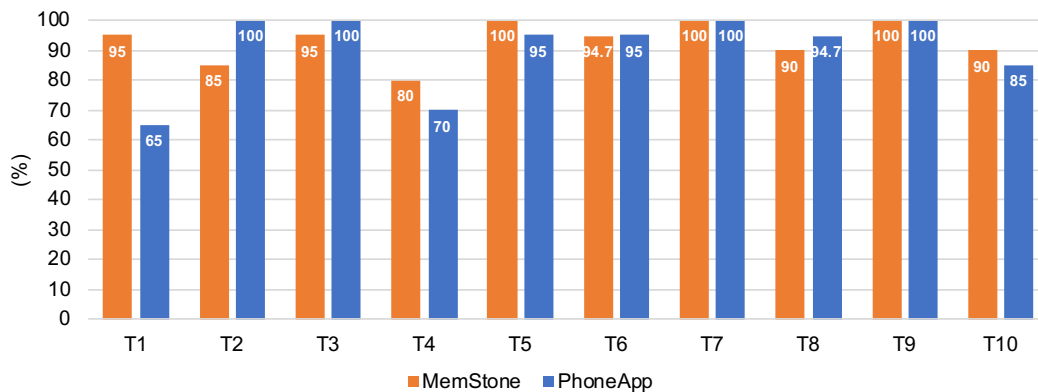


Figure 5.5. Average task completion rate across all participants. See Table 5.1 on page 129 for a description of tasks T1 through T10.

failed to recognize the performed gesture). As displayed in Figure 5.5, participants on average performed equally well with both interfaces. However, when looking at individual tasks, MemStone users performed less mistakes than phone users in tasks related to ‘*capturing and sharing with others*’ (T1 and T5) and ‘*capture only for oneself*’ (T4 and T10). The challenge with performing tasks T1 and T5 using the phone UI is that one has to press two buttons, one for recording and for sharing, as opposed to the single gesture (face-up) with MemStone. From our observations, we saw that most participants pressed only the “start recording” button in these cases. In two cases, we observed that participants confused the MemStone gesture for performing T4 and T10 (i.e., stand-on-side) with those of T3 and T7 (i.e., face-down and double-tap, respectively). When using MemStone, two participants failed to do T2 (reading the elapsed time of the current capturing) and T8 (verifying that that the preceding task on locking the sharing session) since they did not notice that such information was displayed in the MemStone’s screen.

For each task we also measured the time between the moment the visual clue was provided (an icon being displayed in the video of the recorded meeting) to the moment participants performed the correct action. Such information was precisely computed from the session video recordings using the timestamps overlaid on those videos. For all tasks, MemStone outperformed the phone interface in this aspect: using the MemStone, participants could perform a given task on average 2.5 times faster than when using the phone interface (see Figure 5.6). We conducted a repeated measures ANOVA to compare the effect of control interface (MemStone and Phone app) on task completion times. The results show

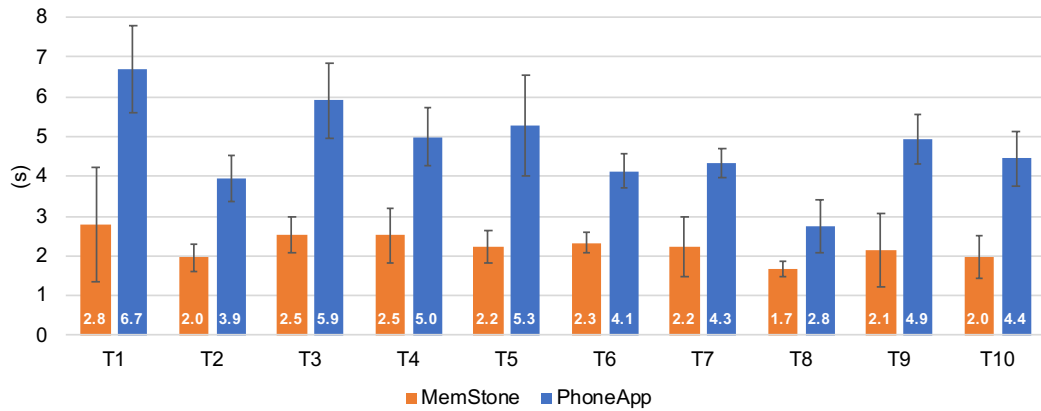


Figure 5.6. Average task completion time in seconds (includes data from successfully completed tasks only). Error bars represent 95% confidence interval.

that MemStone was significantly faster than the Phone app (Wilks' Lambda = 0.485,  $F(1, 180) = 190.799$ ,  $p < 0.001$ ).<sup>4</sup> Moreover, we performed a Sobel test [220] to check if these results are not due to the fact that some of our participants had not recently used an Android phone (and that they could have spent more time performing the study tasks using the Android-based test phone, thus, making the control with the phone slower). Test results confirmed that the effect of control interface on task execution times was not significantly mediated by participants' smartphone ownership ( $z = 0.056$ ,  $p = 0.954$ ).

This was also reflected in participants' perceptions during the interview sessions, where the MemStone was perceived to be more efficient in performing the given actions:

*"I think the dedicated device is better, it is just easier. You do not need to check your phone and you are not losing time. While in the phone maybe you get a message and you want to read it, hence it is not efficient."* (P13, similarly P14, P19, P20).

### 5.5.2 Perceived Usability and Learnability

We also compared the two interfaces with regard to users' perceived usability and learnability. After watching a recorded meeting and using one of the two interfaces for controlling meeting capture and sharing, participants evaluated the interface using a SUS questionnaire. SUS is a ten-item questionnaire that has

<sup>4</sup>Statistics were computed using data only from successfully completed tasks.



been extensively used by usability practitioners for assessing the perceived usability of a system [221]. Recent work [222, 223] suggests that the SUS result can be decomposed into two components for measuring both usability and learnability of a system. After applying this approach, the average usability scores for both the MemStone and the phone interfaces are 75.31 (SD=18.38) and 82.81 (SD=16.42), respectively. This suggests that the gesture-based MemStone prototype is in principle usable above average. However, its lower score against the phone app reflects user concerns regarding the necessity for an additional control device, which we discuss in section 5.8.

As for the learnability scores inferred from SUS results, MemStone scored an average of 78.75 (SD=22.61) and the phone's score is 89.37 (SD=13). Unsurprisingly, the phone – being an already established concept and an artifact that most of us know how to use and operate – scored higher than the novel interface concept of MemStone. All phone buttons were also unambiguously labeled, making it easy to use without having to remember much. However, the relatively small difference in their scores implies that participants find MemStone not significantly harder to learn. This was also highlighted by some participants during the interview. They believed that MemStone could be quickly learned also by children:

*“I think it was a bit easier to use than the phone app. You do not have to unlock your phone and choose the right button, so it was quite intuitive and user friendly. And I think even children would be able to use this correctly and learn it in few minutes.”* (P2, similarly P7, P14).

### 5.5.3 Perceived Intuitiveness and Enjoyment

During the interview sessions, we asked participants to reflect on how they would compare these interfaces in terms of intuitiveness and enjoyment. Enjoyment is an important part of the usability of a product, as it positively influences both a user's willingness to learn and their tolerance for interface shortcomings [224]. Participants acknowledged the fact that the phone UI was clear and intuitive, and that it was similar to many other apps that they use everyday:

*“The phone is more intuitive. You have the feedback you know exactly what is going on. As for the buttons you know what each of them does.”* (P13, similarly P19).

However, they also regarded the MemStone as almost equally intuitive to interact with:

*“The phone is labeled, something you use everyday. It is easy and more intuitive. It does not mean that the box [MemStone] was harder, it was also really easy to understand.”* (P19, similarly P14, P16, P20).

Others believed that the MemStone could actually become more intuitive once users would get to know it better:

*“Once you pass the learning phase it becomes more mechanic, probably without thinking so much you would just use the MemStone.”* (P17, similarly P5).

Participants were more decisive in expressing their opinion on the devices' perceived enjoyment. A clear majority said that *using the MemStone was more enjoyable and fun* (P1 but also P2, P4, P6, P9, P13-P20). The reasons for this choice were various: *the phone UI was just another app* (P13, P14, P16); *the MemStone was a novel concept operated by physical manipulations and is something that one will not encounter frequently* (P13-P20); or also because the phone is more invasive and requires active user focus:

*“The phone took out the fun the moment I had to stop whatever I was doing and focus on it to search the necessary functions. That was not fun! The cube [MemStone] was definitively more fun.”* (P20, similarly P17).

Others compared the experience of using the MemStone with that of a toy:

*“The MemStone is more fun to use, it is a toy essentially. . . .”* (P13),

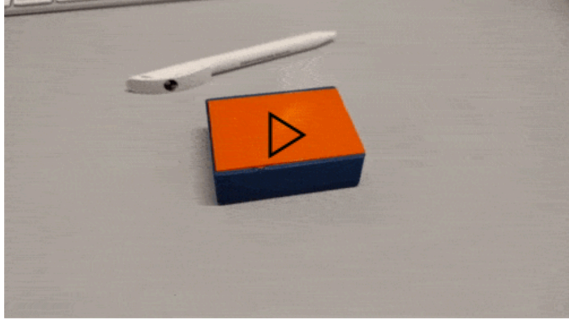
while another participant got emotionally attached to it comparing it with a pet:

*“And I like the vibration, it becomes like a pet in a way. I guess you can get very emotionally bonded to it.”* (P20).

## 5.6 Results: Long-term Gesture Memorability

In addition to exploring several characteristics of MemStone gestures, including efficiency, enjoyment, learnability and usability, we also explored gesture *memorability*. Prior research highlights memorability as a key characteristic of gesture-based interfaces [225, 226], since easy-to-remember gestures can reduce mistakes and frustration (especially when one is focused on other important things

9. Which of the following actions will be triggered when performing the face-down gesture?



1. Capture meeting data only for myself.       5. Delete the last 30 seconds of captured data.

2. Capture and share data with meeting attendees.       6. This gesture wasn't used in the previous study.

3. Don't capture and don't share any data.       7. I don't remember the action for this gesture.

4. Lock sharing to only the current set of attendees and don't share with anyone that might join afterwards.

Figure 5.7. Illustration of a survey question from the follow-up study. A small video shows how the gesture in question is performed.

during a work meeting). Additionally, memorability can also increase adoption of a gesture-based interface [227].

After conducting the comparative user study, we administered a follow-up study with the goal of investigating participants' long-term memorability of the MemStone's physical gestures. This follow-up inquiry was conducted four months after the first study. We contacted all prior participants by email and asked them to participate in a short follow-up online survey. Eleven participants from the prior study (55%) participated in the follow-up study. As in the first study, no incentive payment was provided to participants for this follow-up study.

### 5.6.1 Follow-up Study

The survey contained two types of multiple-choice questions. The first 10 questions showed a MemStone gesture (e.g., "face down") and then asked participants to select which of the given five actions would be triggered after performing the gesture (as illustrated in Figure 5.7). For each gesture the survey showed a small embedded video of how it is performed. As the original study only had

five gestures for five actions, we included five *fake* gestures that were not used in the previous study, yet which were somehow similar to the original five gestures: 1) *squeeze*, 2) *rotate*, 3) *swipe*, 4) *tap-on-table*, and 5) *slide-device-back-and-forth*. The fake gestures were meant to understand how memorable the original gestures were. Given that not all of the 10 gestures mapped to an actual action, participants could answer “This gesture was not used in the study”, as well as “I do not remember this gesture”. A second part of the survey questions then investigated the reverse, i.e., the mapping of actions to gestures. Here, 5 questions showed an action (e.g., “pause recording”) and participants then had to select which of the 10 previously shown gestures (i.e., 5 real gestures and 5 fake gestures) would trigger this action. An 11th option again was “I do not remember the gesture for this action”. All 15 questions were presented to participants in random order.

### 5.6.2 Results

Figure 5.8 provides an overview of the results from the long-term gesture memorability survey. From participants’ individual responses (columns P1 to P11 in Figure 5.8) we observe that five participants answered all questions correctly, three participants have a correct response rate between 60% and 73%, while the other three participants (P5, P10 and P11) could only successfully answer less than 60% of the questions. We investigated further their individual performances achieved during the first study. There was no indication for P5’s low performance, however, for both P10 and P11, their low level of engagement during the first study and also their lack of interest for extra electronic devices seem reasonable explanations for their low performance. This can also be confirmed by their perceived SUS scores for MemStone’s usability (32 and 55).

When looking at the gesture-action mapping (see rows Q1 to Q10 from Figure 5.8 with a combined order of gesture and related action questions), both the ‘face-down’ gesture and the ‘no capture’ action were successfully mapped in 90% of the cases. The second best such mapping (with a correct response rate of 72%) is for the ‘shake’ gesture and ‘delete 30 seconds of data’ action. The weakest mapping with only 45% was for the ‘double-tap’ gesture and the ‘lock sharing’ action. By further investigating this association, we observed that in four cases the ‘double-tap’ gesture was mistaken with another action (once with ‘delete 30 seconds of data’ and three times with ‘capture and share data’). When asked to identify the gesture for the ‘lock-sharing’ action, four participants said that they do not remember it, and two others selected a wrong gesture (‘face-down’ and the ‘squeeze’ fake gesture).

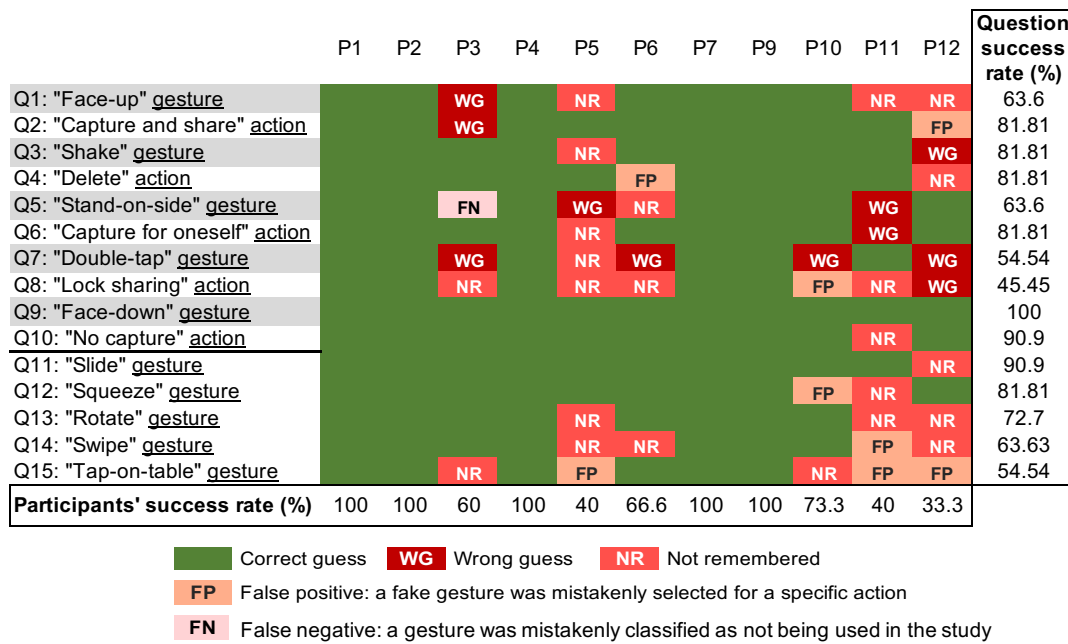


Figure 5.8. A heatmap illustration of the distribution of answers from the survey on long-term gesture memorability. Odd numbered questions from 1 to 10 ask participants to map a given gesture to an action, while even numbered questions ask the opposite. Questions from 11 to 15 quiz participants on the action that would be triggered from a fake-gesture that was not used in the first study.

Among the fake gestures, i.e., physical gestures that were not used in the first study (rows Q11 to Q15 from Figure 5.8), the ‘tap-on-table’ and the ‘swipe’ were wrongly selected as to trigger an action in 3 and 2 cases, respectively. In line with the previous outcome, the ‘lock sharing’ was the mostly mistaken action to be triggered by such fake gestures.

All in all, even after four months, participants were able to successfully recall the relationship between most physical gestures and control actions. Apart from the association of the ‘double-tap’ gesture and ‘lock sharing’ action, all others gesture-actions were correctly identified in more than 60% of the survey questions, which suggests that they can be successfully used with the proposed tangible interface concept.

## 5.7 User Perceptions of the MemStone Device

We now report the qualitative results from the open-ended discussion session. This data unfolds users' perceptions of the MemStone interface, specifically focusing on the MemStone's design, interaction technique and visibility in conveying feedback to users.

### 5.7.1 Suggested Design Improvements

Participants generally liked the concept of the rectangular-shaped box. They also liked its bi-colored design, the two LEDs (which indicate capturing and sharing activities), and its central screen. However, they proposed several improvements related to its physical design.

Participants believe that making it *smaller* and *lighter* would also make the device to better fit their hands, thus being also easier to be used (P2, P13). Moreover, it would also become less of a burden to carry it with them:

*"I think you can do this device smaller, like a USB stick. I have 3-4 sticks with me and it is not a big problem."* (P13, similarly P14).

Additionally, the current prototype was perceived as being a bit slippery, so designers should take into account materials which provide better grip:

*"The current prototype is a bit slippery and it can easily escape from the hands."* (P15).

To further improve the device's feedback, participants suggested to include an additional LED for the 'lock sharing' action, similarly as for the 'capturing' and 'sharing' actions. Participants expressed contrasting opinions regarding the amount of information displayed on the MemStone screen. Some would like to have a less cluttered screen that would show only the number of peers one is sharing data with, together with the elapsed time of sharing, while others would want even more information, such as as detailed profile information on the connected peers and their physical distance. This suggests that one should consider a customizable interface that can be switched between such simple and comprehensive views.

## 5.7.2 Interaction Techniques

### Gesture Affinity

Study participants expressed an affinity for the underlying physical gestures. The mapping between gestures and their control actions was well aligned to users' mental models, thus it was easy to associate them during the study:

*“I think it is very intuitive, if you have it face-up, it kind of radiates through versus everybody sitting around, facing yourself it is just you and face-down is off. You have chosen the functions nicely with the physical movements and the device's position.” (P7);*

*“Once you rationalize them, you see that some are quite easy, such as turning on the other side. But also other ones pair to their actions and make absolutely sense. From that point of view they become easy to remember and can be used without thinking.” (P16, similarly P9, P14, P17-P20).*

P16 further commented on the low physical demands for performing such gestures, suggesting that the interface can be also *operated by people with slight physical disabilities*.

### Gesture Challenges

Users also expressed some concerns and challenges with the 'shake' gesture, as well as with the 'double-tap' one. While the mapping between 'shaking' and 'delete data' was not necessarily questioned, it was seen as a *rather hard-to-perform gesture* (P4, but also P5, P7, P14, P17, P18). Moreover, it was suggested that it may be even considered as *not polite to perform it* while one is speaking in a meeting. Suggested alternatives were 'swipe' or 'squeeze'.

While 'double-tap' was considered an *easy-to-perform gesture*, most participants expressed their concerns regarding its relationship with the 'locking sharing' action. They also believe that 'locking' should be extended beyond a binary lock/unlock model, so that one can be more selective on whom to keep and remove from their locked session.

## 5.7.3 Role of Device Visibility

Participants appreciated the fact that MemStone is more visible and transparent than the phone in conveying its feedback to all co-located users. Such openness

had a two-fold effect on participants' perceptions about the device. First, participants said that they *felt more confident about their privacy*. For instance, should users agree not to record some part of the meeting, then by looking at others' MemStones they can easily understand if they are behaving according to the agreed protocol:

*"If there's something that doesn't have to be recorded and I ask for it, then I can see if people are following. This gives me more confidence. It's a way to see what people do and how they behave; it's an additional feedback."* (P15, similarly P4).

The increased confidence, however, may also come with a cost. Some participants said that the device's transparency could also influence their data capturing and sharing practices:

*"It would extremely change my behavior even if I might try not to let it influence me. Thinking that somebody is watching me, or even recording me, you become self-aware, it changes your behavior; you want it or not."* (P17, similarly P2, P4, P14).

However, others suggested that their decision to share or not actually *depends on the context and on the other attendees* (P13, P15, P18, P19). This outcome is in line with prior work [228], which suggests that knowledge sharing behavior is influenced by multiple factors.

Lastly, some participants expressed their belief that by observing the action of others one can increase their meeting concentration:

*"Since one is recording, then maybe someone will say something important and I should record too."* (P14, similarly P16, P17).

## 5.8 Discussion

Overall, our TUI was efficient in allowing users to control access to, and sharing of, captured experiences in a meeting context. It was also perceived as user-friendly and enjoyable to use, which is in line with findings from Hoyle and colleagues [105], suggesting that users preferred an in-situ control method rather than other post-hoc approaches, such as [80, 112].

MemStone's efficiency and effectiveness, but also our participants' affinity to its gestures, highlight that the device's affordances (i.e., range of possible activities) are visible and clear to users. This is in line with findings from Sheridan



and colleagues [209] on the affordances of box-shaped interfaces, also following Norman's insight that such affordances are useless if they are not visible to users [229]. We also found that participants could relatively well remember the physical gestures and their corresponding actions even when asked four months after. Our results suggest that a reasonable relation between gestures and their actions is what makes them memorable, as it is also suggested by Nacenta et al. [227]. Moreover, such memorable gestures do not only confirm users' perception of an easy-to-use device, but they also suggest that MemStone can be reliably used in infrequent settings, e.g., monthly meetings.

In general, most of our study participants expressed the desire to use MemStone in a context where devices are provided at the event location, e.g., meeting rooms (P2, P5, P13, P14, P15, P17, P19, P20). However, despite our prototype outperforming a comparable smartphone application, the interview sessions uncovered our participants' concerns for having to carry yet another device. Participants expressed their disinclination to carry a personal MemStone device with them during their everyday activities. While this suggests that convenience trumps efficiency [230], it might also be a as-of-yet too infrequent use case (controlling capture devices) to be of much use to people. A future filled with a plethora of capture situations may very well change this perspective. Nevertheless, we believe that this requires additional investigation to shed more light on users' concerns.

MemStone is more efficient in capture and sharing controls than a smartphone for static experiences where a user does not move very often (e.g., in work meetings). This allows users to perform the necessary physical gestures easily, but also the device can pick them up with a higher accuracy (since there are no other physical movements that could trick the device's gesture detection logic). On the other hand, a smartphone might be better for other activities where users are in constant move (e.g., having a walk with a friend). Here, a touch-based interaction might be less error-prone than MemStone's physical gestures. Even assuming a smaller MemStone that can fit in a pocket and can be easily carried around, it still can be more cumbersome for one to perform specific gestures, especially those that require the MemStone to be put in a specific position (e.g., face-down, face-up, and stand on side position).

We believe there is value in further improving the initial concept of MemStone based on our participants' suggestions. Thus, we highlight several improvement ideas in this section. One such improvement that follows naturally is to allow users to share their memories even with others that were not part of the same event. For example, by 'touching' two MemStones together, users could share data captured in the last hour, e.g., following a similar sharing process from the

work by Geronimo et al. [231] on mid-air gestures. Furthermore, our results highlight an interesting dichotomy regarding the level of presented information on the MemStone's screen. On one hand, some participants liked a minimalistic screen showing as few as possible information. On the other hand, other participants advocated for a more comprehensive description of the capturing and sharing practice. Based on this, we believe that the MemStone can be configured with two such display modes.

The current prototype is passive and it responds only to user input. As a future improvement, the MemStone can be made more proactive and context-aware. By inferring the context of its owner (e.g., location, event, time of day, etc.) it can also automatically suggest sharing decisions, for instance by beep or vibration.

In addition to physical gestures, MemStone can be extended with voice interaction, similarly as the emerging intelligent voice assistants (e.g., Amazon Alexa, Google Home or Apple's Siri). However, reflecting on many security incidents with these devices [232, 233], this complementary feature could create more tensions among the MemStone users, instead of capitalizing any user benefits. Related to the aspect of security, our current MemStone prototype does not authenticate user gestures. A malicious person can 'easily steal' one's MemStone and then delete one's experiences by shaking the MemStone. This can be prevented by authenticating all user actions. One can imagine a MemStone equipped with several fingerprint sensors or even the whole surface acting as a sensor to allow for a seamless authentication. Additionally, the way how the gesture is performed could in itself be used to identify the real owner of the device, following similar authentication systems based on physical gestures [234, 235].

So far we focused our discussion on ideas that can further enhance user experience with the MemStone. But there is another dimension of how we envision it to be used, which goes beyond acting as a personal memory control interface. In fact, the concept of such a visible and open interface can help users navigate social constraints inherent in lifelogging. Unlike taking pictures using regular phones or cameras, the act of capturing pictures automatically using wearable cameras can create discomfort and social tensions. As we previously described in Section 2.2.4, automatic capture equipment can create fear of covert recordings as they tend to capture others without consent. These fears have been manifested in several unpleasant situations in the past. In 2012, Steve Mann – a lifelogging pioneer – was physically assaulted in a McDonald's restaurant in Paris for wearing digital eye glasses [236]. A similar incident happened later on in 2014: while a person was entering a bar in San Francisco, the situation escalated and she was verbally assaulted because of her Google Glass [237].

Koelle et al. [106] think that non-existent or poor feedback mechanisms of wearable cameras are to be blamed for the lack of bystander awareness, which prevents them from reacting on or objecting to being recorded. In an effort to increase bystander awareness of capture situations, they propose eight camera designs that go beyond having a simple LED status light. However, extending the camera design with additional physical artifacts may not always deliver the intended result. As we discussed in Chapter 4, wearable cameras are very often obscured by hair or clothes. This would make it hard to observe any such physical feedback artifact on the camera. What is more, following the paradigm of ubiquitous computing, the already diminutive cameras will become smaller and smaller until we will start implanting them in our lenses or body, hence making them invisible to others.

In light of this, we believe that the concept of a tangible interface such as MemStone, can offer a viable complementary feedback alternative. While a person sitting across us may not be able to verify whether our camera is on or off simply by looking at it, she is more likely to infer this information by verifying the color of our MemStone (i.e., seeing the orange colored back side of MemStone is an indicator that we are not recording). Others would furthermore know whether we are also sharing such captured data or we are only capturing for ourselves (i.e., if our MemStone is standing on side it means that we are not sharing any data with anyone). Moreover, the MemStone can also be used to convey privacy messages to other lifeloggers and explicitly express our disagreement from appearing in their recordings. For instance, by putting our MemStone face-down we can decide to opt-out from the recordings of co-located others. For this to work, cameras of others lifeloggers need to detect and recognize the position of our MemStone, and consequently block our face on the images that capture. Prior work has proposed similar approaches for privacy-respecting capture. Using techniques from computer vision they can block faces of bystanders by recognizing their hand gestures or particular visual markers that they wear, such as colored hats or vests [95, 238, 239]. Obviously one can also mistakenly detect a MemStone of another person sitting close us, who might have different privacy preferences than ours. Hence, any approach requires careful design and engineering in order to avoid this risk.

As a possible limitation of our study, we acknowledge the fact that participants had to pretend that they are participating in a recorded meeting, which they watched through a laptop. However, even if we would have organized real meetings with our participants, they would also have to be scripted in one way or another, and hence would still be just as artificial as the recorded meetings. Nevertheless, we believe that evaluating the prototype in such lab scenarios still

allowed participants to control different aspects of a meeting capture scenario. Results obtained from this study allow us to answer our research questions about the comparison of our interface with a more traditional smartphone interface, and explore participants' perceptions on our TUI prototype.

## 5.9 Chapter Summary

In this chapter we investigated the feasibility of controlling the capture and sharing of experience data through in-situ physical gestures. To this end, we presented MemStone, a tangible user interface for achieving fine-grained control over those practices. MemStone is a small 3D printed box powered by an embedded computing platform. Its rectangular shape and bi-colored design allow its owner, but also other co-located users, to infer its current operation at a glance and even from a distance. On its front side, there is a central screen and two LEDs, that all together provide additional feedback regarding the data capturing and data sharing processes.

MemStone supports five different controls regarding how one captures and disseminates experience data. Each of those controls is triggered by performing a physical gesture using the MemStone. For instance, putting the MemStone 'face-down' stops one's gear from sourcing any information and prevents the system from sharing it with other co-located peers. Contrary to this, putting the MemStone 'face-up' triggers data capture from the user's lifelogging gear, as well as informs co-located others that the user is willing to exchange data with them. Additional controls include: capture data only oneself and not share it with others, lock a sharing session and prevent peers appearing later join the sharing practice, and delete the last 30 seconds (or more) of captured data.

We evaluated our prototype in a meeting capture scenario with 20 participants. We found that our participants were significantly quicker in performing data capturing and sharing controls using MemStone than using a more conventional mobile app interface. The concept was highly valued by the participants, it was perceived as user-friendly, quick to learn, and easy and fun to use. Participants also expressed a positive attitude for the physical gestures and their relationship with the control actions. We also found out that participants were able to remember the control gestures even after a long time period, which suggests that such TUI is suitable to be used in less frequently occurring events. However, in spite of its better performance and its high perceived value as an ambient-based control device, participants were very much divided about the convenience of having to carry an additional personal control device with them

for their everyday activities. Lack of frequent data capture practices could well be the reason of why our study participants did not perceive the benefits of a dedicated control interface.

Finally, we believe that a tangible user interface can help users navigate some of the social constraints inherent in automatic data capturing systems. Their open and transparent feedback mechanism can increase recording awareness, allowing bystanders to react or even object to being recorded. Additionally, MemStones could also convey privacy preferences to the cameras of others. For example, A MemStone that is positioned 'face-down' is an indicator that its owner does not want to be recorded by others. Consequently, upon recognizing such situation, a privacy-respecting camera would then blur any face that appears close to such MemStone.



## Chapter 6

# Access Control for Memory Augmentation Systems

In Chapters 3–4 we described our approach based on trusted camera sensors for recording, storing, and exchanging experience data with other co-present peers in a secure fashion. With the goal of protecting users' privacy, in Chapter 5 we furthermore presented a tangible interface (MemStone) for regulating different aspects regarding the capture and sharing of experience data through usable in-situ controls. Our study results indicate concrete benefits when using MemStone in collaborative context, where captured data is meant to be exchanged between all users that are experiencing an event together.

Irrespective of the actual interfaces for influencing how memories are captured and shared, this requires the availability of a context-aware access control mechanism that can both express and enforce user sharing preferences. Addressing these controls requires filtering both the input to users' memory repositories (i.e., controlling memory capture) and the output from those repositories (i.e., controlling sharing of captured data) based on a set of user preferences. Figure 6.1 depicts our holistic approach to a privacy-aware and secure system for human memory augmentation. In this chapter, we set out with the goal of surveying existing access control mechanisms and evaluating their suitability towards such demands of pervasive memory augmentation systems.

At some level, this may seem to represent a traditional data access control challenge. However, as indicated by prior research, conventional access control models, such as, Mandatory Access Control (MAC), Discretionary Access Control (DAC), and Role-based Access Control (RBAC), are designed for operating systems and closed databases and are not suitable for context-aware pervasive systems [18, 240]. In fact, they are either not enough flexible and too

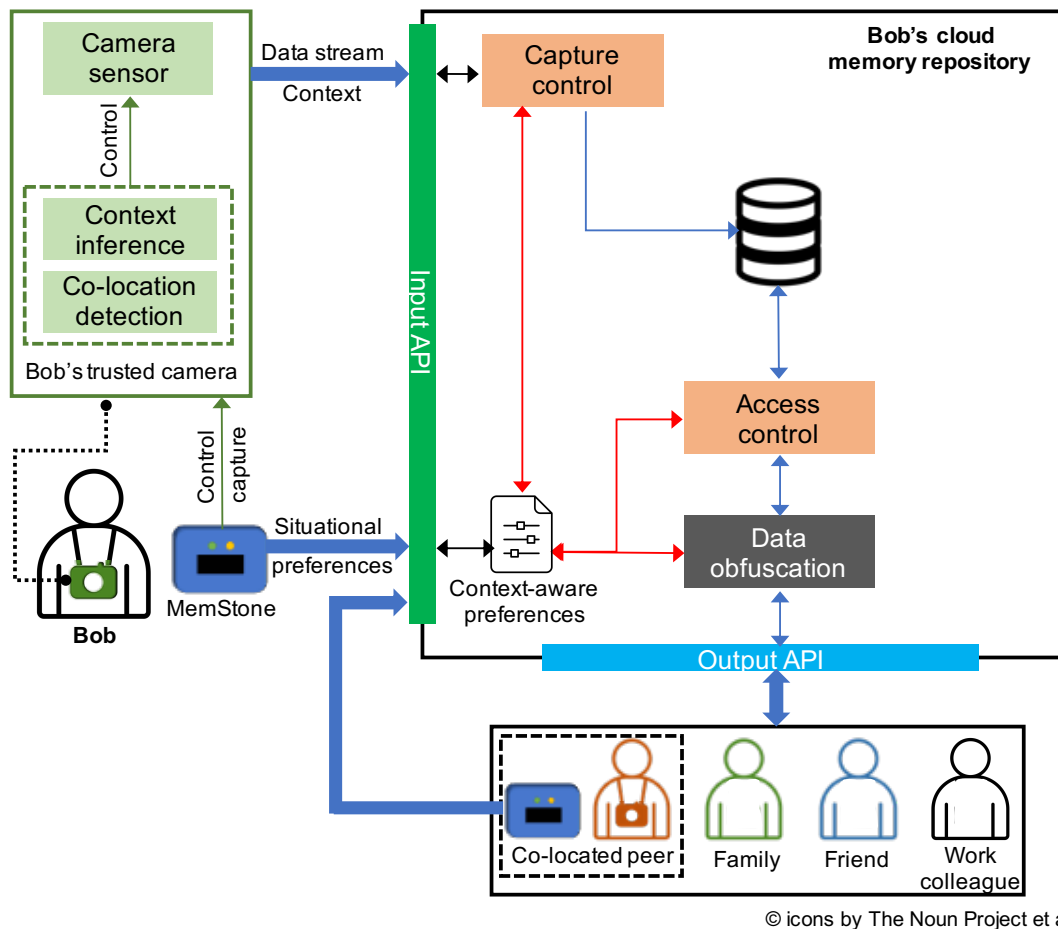


Figure 6.1. A holistic approach towards a privacy-aware and secure system for human memory augmentation.

structured, or not sufficiently robust to support such dynamic user needs. To this end we shift our focus on the novel class of context-aware access control solutions. We identified existing access control systems from several survey studies centered on privacy protection in pervasive systems, social networks, and collaborative systems [18, 240, 241, 242]. We also looked at existing papers with more than 10 citations indexed in Google Scholar that had “context”, “pervasive computing”, “privacy”, and either “access control” or “policy language” in title, abstract, or keywords. We selected only those papers that also outlined examples (at least conceptually) showing practical implications of the proposed access control system.

We start out by formalizing the aforementioned needs and delineate a finer-grained set of capture and access control requirements. In doing so, we rely on



findings from the literature and from our own experience working with memory augmentation systems. Finally we evaluate the suitability of novel access control systems with respect to the elicited requirements. Our analysis highlights that there is a growing interest from the research community for addressing privacy and security challenges of pervasive systems. However, there is no *one-size-fits-all* solution that considers the demands of pervasive memory augmentation systems and that more work is needed to efficiently address them. In this chapter we address the second part (underlined below) of the following research question:

- **RQ4:** What interfaces and policy-based access control models can we use to exercise control over data capture as well as to prevent the disclosure of private and sensitive information when sharing experience data?

*Parts of this chapter are based on the following publication:*

- **A. Bexheti** and M. Langheinrich, “Understanding Usage Control Requirements in Pervasive Memory Augmentation Systems,” in *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM ’15. New York, NY, USA: ACM, 2015, pp. 400–404

## 6.1 Control Requirements

In this section we explore each of the aforementioned control needs and derive a comprehensive set of capture and access control requirements (see Table 6.1). Our findings are motivated by past work in the areas of memory augmentation systems and lifelogging.

### 6.1.1 Controlling Experience Capture

It is crucial for users to control which of their experiences will be captured and how will this be achieved. This is not only important for shaping how they will remember their prior experiences, but also for reducing the chances of capturing any private and sensitive information in the first place. With respect to this, we foresee the following control requirements.

**CR1: Context-based experience capture.** Most of the time the recording system will be by default activated and will capture everything in an implicit manner. However, we believe that users would want to control which events cannot be

<b>Requirement</b>	
<b>CR1:</b>	Context-based experience capture
<b>CR2:</b>	Control capture data types and frequency
<b>CR3:</b>	Control capture devices
<b>CR4:</b>	Sharing based on context and interpersonal relationships
<b>CR5:</b>	Control sharing data type and granularity
<b>CR6:</b>	Support for automatic data obfuscation
<b>CR7:</b>	Delayed data sharing
<b>CR8:</b>	Time-limited sharing and access revocation
<b>CR9:</b>	Support for obligations.
<b>CR10:</b>	Support for collective ownership

*Table 6.1.* Elicited control requirements.

recorded should they have any concerns on what is being captured. While one can prepare daily (or even weekly) policies to specify which upcoming events she would like to have captured, this requires active user involvement and can only work if one has a fine-grained plan of their day or week. We believe that a more practical and realistic approach is to filter based on context. For example, one can record all meetings at work, weekend activities with family, or school lectures and seminars. Using the same control mechanism one can also say which events they would not like to be captured.

**CR2: Control capture data types and frequency.** In certain situations one may want to control how an experience is captured, i.e., which sensors are used and at what frequency. Capturing both images and audio-snippets may be desirable for leisure activities, e.g., a hiking trip with the family, but recording audio could be problematic and not acceptable for work-related activities, e.g., recording a meeting while company's financial situation is being discussed. Moreover, for situations where there are not so many scene changes (e.g., in a meeting room) it may be desirable to capture fewer images (since otherwise most of the images will be similar and portray the same scene).

**CR3: Control capture devices.** In addition to memories captured by their own devices, users can also benefit from data captured by infrastructure devices, as well as data captured by the devices of others. The drawback with such an approach is that one has to rely on capture devices that are owned and controlled by third-parties, as opposed to, e.g., personal cameras. As we saw in previous

chapters, a malicious service provider may intentionally modify data prior to sharing them with the user, in order to influence how the user will recall an event and eventually manipulate the user's episodic memory. Therefore, we believe that users should be able to specify the different sources they are willing to accept data from.

### 6.1.2 Memory Access Control

After an experience has been captured and stored in a secure storage, one has to regulate access and specify who can obtain which data. We previously described our system that enables a secure exchange of captured experiences among co-located peers (Chapter 4). We furthermore presented a tangible interface for controlling some aspects of such sharing practice through physical gestures (Chapter 5). Here we go a step further, as we describe a series of finer-grained controls for addressing privacy risks stemming from the sharing of experiences.

**CR4: Sharing based on context and interpersonal relationships.** Similarly as in CR1 (i.e., using context to control experience capturing), context can also play an important role in specifying which memories are deemed shareable. For example, one can decide that all memories of work-related events can be shared with (co-located) colleagues. To this end, the system should also permit one to specify to whom they are willing to disclose these data. Prior work has shown that interpersonal relationships among users (e.g., friends, work colleagues) can be an intuitive way of specifying recipients of data in access policies [240].

**CR5: Control sharing data type and granularity.** Context can offer a first-level of control but it might not be enough in certain situations. Whilst an event might be flagged as shareable, there could be particular moments that, if shared, could potentially risk users' privacy. For instance, in a meeting capture scenario where data is seamlessly shared with all participants, attendees agree to leave their capture gear on even during the meeting break because there could be some fruitful discussions happening at that time. However, one user opens their laptop and in this case, their camera could have captured the laptop screen (e.g., while entering credit card details) or the card itself (showing the numbers printed on the card). The user would clearly not be happy sharing any of the images captured around that time. In fact, findings from several studies highlight the necessity for such control when the sharing of captured experiences is considered [58, 97, 105, 243].

**CR6: Support for automatic data obfuscation.** Denying access to images that contain sensitive information can protect users' privacy but at the same time it limits the volume of shared experiences. Instead, a user might still share a modified version of those images with sensitive parts being obscured or blurred. Ideally, such blurring can happen automatically without active user involvement.

**CR7: Delayed data sharing.** A prior work by Efstratiou et al. [244] studied user privacy concerns when sharing sensory data in social network sites (e.g., location tracking, conversation monitoring and interaction with physical objects). The authors suggest the possibility of delayed sharing of potentially sensitive information as a way of increasing user control over their data. This can be a complementary step for the automatic data obfuscation operation (e.g., a user can manually verify if the system has properly cleaned the sensitive information), or it can be a fail-safe approach handing control back to the user in case the system fails to automatically clean the data. This would then allow users to manually run scripts to, for instance, identify memory cues that can contain sensitive information, and perform the actual data cleaning.

**CR8: Time-limited sharing and access revocation.** Yet another useful control is the possibility to share memories for a limited time frame (e.g., 1 week or 1 month) [245]. During that time, recipients may access the memories one is willing to share with them. Once the time expires, the data sharing link will not be accessible anymore and access will be revoked to any data that has not already been downloaded. This control is motivated by similar time-limited sharing approaches employed in several location sharing systems [246].

**CR9: Support for obligations.** A user that is willing to share her personal memories might do so under certain conditions or obligations [21]. An obligation can be a pre-sharing agreement regarding how shared data can be used, how should it be stored and processed or whether it can be re-shared and disseminated with others, etc.

**CR10: Support for collective ownership.** Looking at the different stakeholders of memory augmentation systems, Gurrin [8] categorizes them into four groups: 1) *the lifelogger* is the individual who captures an experience using their own recording gear and stores captured data in a personal repository; 2) *the bystander* is a person who has been accidentally captured and appears in one's data, but without having interacted with the lifelogger; 3) *the subject* is a person that the lifelogger has been actively interacting with and appears in several instances within the captured data; and finally 4) *the host* is an individual or an entity

that manages the storage of one's lifelog (can also be the lifelogger herself). Gurrin assumes a simple ownership model: captured data is owned by the person who captured it (i.e., the lifelogger in this model). Davies et al. [9] point out that this simple ownership model may not be appropriate in situations where captured data is used as a memory cue. In their view it is not clear who "owns" the memories of collective events, e.g., a meeting involving three people. The challenge here is that there are multiple people that contribute with their data to create the memories of the event and that in some way they are all owners of such data. Moreover, if meeting attendees discuss about some secret matters of a company, than the the company representatives could also claim to own the captured data. The meeting scenario, and other similar examples of collective data capture, highlight the necessity for multi-user ownership models where all involved stakeholders can somehow specify access preferences.

## 6.2 Evaluation of Access Control Models

In this section we present evaluation results of the suitability of access control models with respect to the identified requirements.

### 6.2.1 Support for context-based capture and sharing

Memory augmentation systems will, by default, be always activated and will capture a plethora of users' daily activities featuring different contexts. The variance in users' activities and context can play an important role for users to determine which activities they would like to capture (requirement CR1) but also data from which activity they would feel comfortable disclosing to others (requirement CR4). This has prompted us to explore access control models that accommodate the use of *context* in access decision making.

Most of the existing access control models tailored to pervasive systems allow the expression of *temporal* [247] and *location-based* [248] contextual constraints. For instance, using such systems one can capture and share all their moments during work hours, or all weekend activities happening outside of the home environment.

However, memory augmentation systems claim for more sophisticated controls to express complex conditions using multiple contextual data. Such controls are offered by several works [249, 250, 251, 252, 253, 254]. For instance, Corradi et al. [253], and Ahn et al. [254] propose systems that distinguish between *physical* and *logical* context. Similar to previous models, a physical context

specifies a physical location denoted by geographical coordinates. A logical context identifies higher-level states of both users and the environment (e.g., the activity that one is engaged in or the role that a user has). While a user at any time can belong to only a single physical context, they can very well be associated with several logical contexts. This allows for the specification of finer-grained capture and sharing controls. For instance, one can specify a rule that would capture and share data from work meetings, but will not share any data captured while one is using the laptop during such meetings (to prevent the disclosure of any sensitive data featuring the laptop's screen).

The UbiCOSM system from Corradi et al. [253] offers a similar feature as our system for sharing data with co-located peers. In particular, an access rule can account for the presence of other users that operate in the same physical context. This is achieved by associating permissions to multiple contexts through a *dependence* relationship.

Choi et al. [255] propose a model that allows access policies to also consider a user's physiological state (e.g., user engagement, anxiety, or stress level). A user may want to capture exciting moments but at the same time might not feel comfortable sharing any data of stressful moments.

Apart from having a rich context model, other desiderata for access control models include flexibility and ease of use. Covington et al. [256] propose a generalized version of the role based access control model (RBAC). In their model, the notion of *role* does not only apply to user-related roles (as in the original RBAC) but it also applies to environment roles, which can be used to capture different aspects of the environment. By building on top of a well-established access control model, this work offers an elegant means of using context in access control policies.

Any system that regulates access to one's memory repository must take into account that access policies should be specified to not only govern outgoing traffic (i.e., data flowing out of a repository) but also any incoming traffic (i.e., data being added to a memory repository). In principle, all of the aforementioned models can be extended to govern incoming traffic, however, models explicitly designed with such functionality have been proposed [257, 249]. For instance, the Confab toolkit for developing privacy-sensitive systems, proposed by Hong and Landay [257], targets both the upload and the release of personal data. Confab's data model can be used to represent different entities, such as people, places, and things. Data about these entities is organized in *infospaces* – logical storage units reachable through a network. Both incoming and outgoing requests to infospaces are regulated by two general access operators, *in* and *out* operators, which in turn are driven by user specified policies.

The main strong point of context-based control techniques consists in the flexibility of expressing various access policies through rich context-based reasoning. This approach can introduce additional security concerns should the authenticity and integrity of contextual data be compromised. In their access control for the Aware Home project, Covington et al. [258] have built a trusted version of the Context Toolkit [259] in order to ensure security and reliability of collected environment data.

### 6.2.2 Support for sharing based on interpersonal relations

Through everyday life experiences, but recently also by means of online social network platforms, users establish various interpersonal relationships with others. Prior research has identified the importance and characteristics of interpersonal relationships in specifying data sharing preferences in online social networks [260, 261, 262, 263]. Based on these works but also on our own experience with lifelogging applications [58], we believe that constraints based on interpersonal relationships are important for efficient and intuitive sharing of personal memories (i.e., requirement CR4).

Role-based Access Control (RBAC) introduced the notion of a role to simplify the specification of access control policies. While roles can be used to specify interpersonal relationships, however, the limitation of this model is that it assumes domains with highly structured role hierarchies, such as companies or hospitals. As a result, several proposals have extended the initial RBAC model to also support *dynamic* and context-dependable roles [248, 249, 252, 253, 254, 258]. For instance, the UbiCOSM system [253] permits the creation of user roles as a specific type of the logical context. One can create different roles (e.g., *Family Member*, *Classmate*) by first defining a logical context for it and then describing the role through a set of constraints. The Houdini framework [249], developed at Bell Labs by Hull and his colleagues, provides a general-purpose framework for privacy-aware data sharing of mobile users. Their access control mechanism accounts for four kinds of information when deciding to give access to a particular data, among them being the resource owner's view of the resource requester (e.g., work colleague, friend).

In order to empower users with additional control in the dissemination of their data, research in the domain of online social networks and collaborative systems proposes a novel access control paradigm based on interpersonal constraints, known as Relationship-Based Access Control (ReBAC) [240, 264]. They consider extra information for deciding to whom access should be granted, such as the *depth* of the relationship (i.e., the length of the shortest path between

two users, for controlling the radius of the social circle) and the *strength* of the relationship (i.e., the level of trust of a relationship) [265].

### 6.2.3 Support for multi-user governance

When exchanging captured memory cues with co-located peers, each involved stakeholder may claim a share of ownership on the captured data. To this end, each involved entity should be able to specify rules that control access to and sharing of the produced event data. A final merging of these rules is required in order to determine the actual access permissions on the data in question.

Some of the systems that we surveyed in this work support a basic form of handling the governance of data owned by multiple stakeholders [248, 250, 251]. All co-located users can decide to *delegate* access control and sharing decisions to a trusted person. Through a mutual agreement (e.g., a majority voting) they can first agree on who they think should be in charge of the data captured during the event and then specify rules to delegate all access requests to the access control system of the chosen entity. In case of a meeting capture scenario, the meeting leader can be in charge of the meeting's captured data. This model relies on the fact that co-located users can reach a consensus in selecting a representative to safeguard their collective data, which might not always be the case. Moreover, this approach limits the access control decisions to be specified by a single entity only.

Other trends allow data control permissions to be specified by all co-located users, and access can then be granted only if all user permissions allow for that. This approach has been followed by the Virtual Walls policy language proposed by Kapadia et al. [214]. Using the abstraction of physical walls, their system allows users to control the privacy of their digital traces much in the same way as users control privacy of the analog data in the physical world. In their system, each co-located user can in principle specify their own virtual wall to protect their personal data, and the final access rule can be derived by selecting the rule from the most restrictive virtual wall.

A number of models, stemming from research on online social networks and collaborative systems, highlight the importance of multi-party access control mechanisms when sharing photos online [240]. One of the proposed approaches towards models involving multiple stakeholders is through selective encryption of photographs [266, 267, 268]. Illia et al. [267] propose an access control model that changes the control granularity from the level of photographs to that of identified faces appearing in those photos. At the outset the system will identify depicted faces in an image and will associate them with users' identities.



All identified users from this step can then control the exposure to their own face by specifying access permissions. Finally, when a subject requests a photo, the system will determine which faces should be hidden based on the identity of the requester, and only serve a processed version of the photo by blurring all those faces that the requester is not authorized to see. While the focus of this work is on protecting identified faces only, this concept can be extended to include other objects appearing in an image (e.g., laptop and phone screens, documents, etc.) in order to accommodate for privacy risks in memory augmentation systems.

All of the presented examples of multi-party access control assume an equal level of authority among all of the involved stakeholders associated with a data object. However, more recent research proposes techniques for enabling asymmetric multi-party access control models [269] where the decision power that a stakeholder has over a particular data resource can depend on their relationship with the resource (e.g., whether the person produced the data or is tagged in it) [270]. This is inline with the stakeholders model proposed by Gurrin [8] (see requirement CR10). When computing the final access permissions, policies specified by *the lifelogger* (an individual who captures an experience using own recording gear) can have higher priority than those of *the subject* (a co-located person that interacts with the lifelogger and appears in several instances of their data), which in turn can have higher priority than those permissions specified by subjects that might administer the fixed capturing infrastructure and that are not co-located with the lifelogger at the time of capture.

#### 6.2.4 Data obfuscation and sharing granularity

Access control systems will either deny or grant access to requested memory cues depending on a set of contextual attributes evaluated against access permissions. Denying access to particular data can prevent unsolicited disclosure of private and sensitive information that may otherwise be present in the requested data, and hence protect users' privacy. However, this may also decrease the utility of memory augmentation. A more flexible solution than that of sharing "all or nothing", is to extend access control with data transformation mechanisms in order to *obfuscate* any private data before granting access to it (requirements CR6). Here we focus on mechanisms that address the complementary problem of recognizing and obfuscating private information that may appear in captured experience data. We then shed some light on how access control systems can be extended with data obfuscation mechanisms in order to empower users with even finer-grained privacy control.

Most the systems that support privacy-preserving sharing of visual data rely on computer vision techniques coupled with machine-learning to recognize sensitive content depicted in images [79, 80, 83, 84, 112, 113, 271]. For instance, Korayem et al. [79] investigate the feasibility of automatically detecting computer screens appearing in lifelog images. This work offers promising results showing that it is possible to reliably detect computer screens even from low quality, blurry and possibly occluded images captured by wearable cameras. Using a similar approach applied to lifelog images, Templeman et al. [80] developed a system that can reasonably well infer the space where an image is taken, and hence avoids the accidental sharing of images captured in sensitive places, such as bathroom or bedroom. Note that more work is needed to assess the actual sensitivity level of the experience during which an image was taken. For example, in the event of a detected computer screen, it is not clear whether the screen depicts any private information at all. A user might still agree to share an image of a computer screen that was captured during a work meeting, but the same user might not want to share a similar image that was captured during meeting break, at a time when she was completing an online purchase.

Additional work has shown that it is possible to infer higher level information from visual data captured by wearable cameras [82, 83, 84]. For instance, Castro et al. [83] propose a machine-learning approach to learn and predict everyday activities from lifelog images. Their technique achieves a high overall accuracy in identifying up to 19 different activities (such as working, watching TV, reading, having a work meeting, cooking, eating, driving, etc.). Fathi et al. [84] propose an approach for recognizing social interactions such as discussion, dialogue, and monologue from first-person perspective day-long videos. Their method relies on two kinds of data sources: detected faces and attention patterns. At the outset, their system will try to detect all faces appearing in the video and then estimate their location and orientation. In a second step, the system infers any attention pattern such as who looks at who or whether all users look at a common place. The combination of such information was shown to provide enough insights regarding the type of the social interaction. For instance, if there would be multiple faces detected and if most of them are looking at one person for a longer-period of time then this would be classified as a monologue. Finally, any such high-level information that can be inferred using these approaches should be correlated with specific parts of the image in order to understand if and how the image should be modified.

All of the aforementioned mechanisms for privacy-aware data sharing are intended to work on already captured data, e.g., a resource that is inspected at the time of an access request. Contrary to such *after-the-fact* approaches,

Steil et al. [272] propose a solution that targets the problem at *the source*, i.e., at the time of capture. Their solution, referred to as *PrivacEye*, uses a combination of deep-learning and computer vision techniques to gauge the privacy-level of a given situation and then disable the camera by occluding it with a shutter. This way the camera will *not* capture the privacy-sensitive experience at all. To open the camera's shutter again, *PrivacEye* makes use of estimated privacy-level of an experience, but this time using a secondary camera that captures the user's eye movements. Whilst this approach might further reduce chances of unsolicited privacy infringements (since no sensitive data is recorded in the first place), the drawback of this mechanism is that it lowers the volume of captured data. In situations when captured data is used as a basis for generating memory cues, users might still want to capture data of a privacy-sensitive experience for themselves and not share it with others. As a consequence, an after-the-fact mechanism can be more appropriate for memory augmentation systems as long as it can correctly and efficiently recognize private information as well as be easily integrated with dynamic access control solutions.

To this end, we investigated how the access control models surveyed in this work can be extended with such data filtering techniques. As a first step, we looked at the possibility of encoding any data filtering controls in order to allow users to write full-fledged access policies using the system's choice of language. Secondly, we explored whether the specified filtering controls are supported and can be carried out by the system's underlying policy enforcement component.

We observed that most of the surveyed access control systems support the specification of data filtering or transformation policies [247, 248, 249, 251, 257, 273]. Bagüés et al. [274] propose the *SenTry* privacy policy language as part of their Unified User-centric Privacy Framework. *SenTry* allows the specification of fine-grained access policies and it is specifically tailored to applications from the pervasive computing domain. One particular feature of *SenTry* is its support for *transformation* policies in order to allow users to better specify their privacy preferences. Similarly, the *Ponder* language [251] proposed by Damianou et al. supports information filtering policies that can be applied to both incoming and outgoing data (like *Confab* [257]). In *Ponder*, multiple filter expressions can be specified for a data request and each filter can contain an activation condition. Both the *SenTry* and the *Ponder* languages provide rich expressive capabilities and can easily support the specification of data filtering policies for memory augmentation applications. However, they fall short when it comes to performing the actual transformations on visual data. In fact, any filtering can be only applied to contextual data encoded in string representation, such as location coordination, calendar entries, etc.

In trying to implement filtering techniques on visual data, a simple approach that is supported by most of the access control systems is to impose a temporal delay on the data to be shared. Instead of immediately granting access to a particular resource, the access control component can be configured to share it only after a certain time has passed from the moment the resource was captured. Users can then manually (or automatically by means of automated scripts) execute a data filtering solution to remove any private information before the data becomes available to others. However, such an approach does not allow one to fully exploit the capabilities of dynamic access control models, i.e., one cannot write full-fledged privacy policies that also include data filtering rules.

A better approach that can reduce this gap between the specification of data filtering policies and their enforcement is the support for *external* functions as early as the policy specification stage. We observed that such approach is present in very few systems [257, 275]. For instance, the Privacy Markup Language (PRML) [275] aims at extending the capabilities of corporate privacy policies with data handling capabilities, thus closing the gap that existed between these two concepts. A PRML policy, written using the XML language, specifies that a *role* can do an *operation* on a *data group* owned by a *subject* if certain *constraints* are met. Optionally, the data resource can be subject to a particular *transformation* before an *operation* (such as reading or sharing) can occur. The transformation declaration is composed of a `<name>` element and an `<implementation>` element, with the later referring to an external function that implements the data transformation. Another system that can be extended with custom data filtering capabilities is the Confab privacy toolkit [257], which we described previously. In Confab, data flowing from and to an infospace (logical units containing data about people, places, or things) is filtered by *in* and *out* operators. By default Confab does not support any operators that can transform visual data. However, thanks to its modular design, Confab can be easily extended with custom operators tailored for manipulation of visual data and which can then be driven by user specified policies.

Our analysis highlights that there is no “one size fits all” solution for addressing privacy challenges when sharing captured memory cues with others. On one hand, there is a considerable amount of work on recognizing and eventually obfuscating sensitive information depicted in visual data (from single items that appear in pictures such as faces of people, computer screens or documents, to higher-level and more complex information such as activities one is engaged in or social interactions). On the other hand, more work is needed to integrate such solutions into existing access control systems with the ultimate goal of offering a seamless experience to users when controlling their captured experience data.

## 6.3 Evaluation Summary

Table 6.2 presents a summary of our analysis with respect to the requirements presented in Section 6.1. In the table, “✓” is used to indicate that the system fully satisfies the given requirement subject to potentially minor modifications of the access control system or its model; “○” indicates that the requirement can be still satisfied but this requires some substantial modifications of the evaluated system; “✗” indicates that the requirement is not satisfied by the system or at least it was not clear from the system’s description.

Requirements CR1 to CR3 concern the ability of an access control system to influence how an experience is captured. More specifically, they are related to defining the experience to be captured (i.e., CR1), what data modality should be used and at what frequency (i.e., CR2), and whether one is willing to accept any data captured by the cameras of others (i.e., CR3). For the evaluation of requirement CR1 we marked “full support” for those systems that support rich contextual models. “Full support” was marked also for systems with a rather limited context model (e.g., that can represent only *time* and *location* aspects), but that can be easily extended with a richer model. Consequently, “partial support” was marked for systems that only consider a limited but fixed representation of context situations, whereas models that were not designed for context-aware situations were marked with “no support”. Requirement CR2 is considered fully satisfied by systems that are able to filter both capture data types and capture frequency. In most access control systems that we surveyed, data types can be represented as *resources*, whereas capture frequency can be specified as *parameter* of the resource. Systems that satisfied only one of the these characteristics were marked with “partial support”. Regarding requirement CR3, a clear majority of the access control systems permit the specification of capture devices as *subjects* that can upload data to a memory repository. Therefore, these systems were marked with “full support”. On the other hand, systems that provide a less elegant way of filtering capture devices received the “partial support” flag. Looking at Table 6.2 we can observe that capture control requirements are well supported by many access control proposals.

Requirements CR4 and CR5 are intended to offer initial control on the memory cue sharing practice. Requirement CR4 concerns two aspects regarding this practice. First, using a similar approach as with requirement CR1, it relies on the context to be able to decide which memories can be disseminated to other peers. Therefore, to evaluate this aspect, we used a similar evaluation criteria as in requirement CR1. The other aspect concerns the ability to share memories based on the social relationship with the data recipients. Access control systems

	CR1: Context-based data capture	CR2: Control capture data types and frequency	CR3: Control capture devices	CR4: Sharing based on context and interpersonal relationships	CR5: Control sharing data type and granularity	CR6: Support for data obfuscation	CR7: Delayed data sharing	CR8: Time-limited sharing and access revocation	CR9: Support for obligations	CR10: Support for collective ownership
CPPL [252]	✓	✓	✓	✓	✓	✗	✓	○	✗	✗
PlexC [248]	✓	✓	✓	✓	✓	✗	✓	○	○	○
P2U [276]	✗	○	✓	✗	○	✗	✗	○	✓	✗
Rei [250]	✓	✓	✓	○	✓	○	✓	✓	✓	○
Ponder [251]	✓	✓	✓	○	✓	○	✓	✓	○	○
PRML [275]	✓	✓	✓	○	✓	✓	○	○	○	✗
Houdini [249]	✓	✓	✓	✓	✓	✗	○	○	✗	✗
PDL [277]	○	○	○	✗	○	✗	○	○	✗	✗
PDL-C [254]	✓	○	✓	○	○	✗	✗	✗	✗	✗
UbiCOSM [253]	✓	✓	✓	✓	✓	✗	○	○	✗	○
CoPS [247]	✗	○	✓	○	✓	✗	✓	✓	✗	○
CPE [278]	○	✓	✓	○	○	✗	✗	✗	✗	○
SenTry [274]	○	✓	✓	○	✓	○	✓	✓	✓	✗
Environmental Roles [256]	✓	✓	✓	✓	○	✗	✓	✓	✗	✗
Virtual Walls [214]	○	○	○	✗	✓	✗	✗	✗	✗	○
Confab [257]	✓	✓	✓	○	✓	○	✓	✓	✓	○

✓ Full support (can require minimal modifications)

○ Partial support (subject to substantial modifications)

✗ No support

Table 6.2. Evaluation of access control systems against control requirements.

that support dynamic and contextual relationship models were marked with “full support”. Models based on a less flexible relationship concept, such as fixed hierarchical roles, were marked with “partial support”. Requirement CR5 is quite similar to CR2, but instead of filtering incoming data, here the filter is meant to be applied to data leaving the repository. Our literature review shows that these requirements are largely recognized and addressed by access control systems.

The next set of requirements (i.e., CR6 to CR9) provide further control on the sharing of memory cues. Specifically, requirement CR6 concerns the “cleaning” of any sensitive information depicted on the data to be shared. Systems that permit the specification of data obfuscation rules and that actually perform the cleaning process were marked with “full support”. Systems that do not provide any data cleaning mechanism, but that, on the other hand, are able to execute external data obfuscation procedures through third-party libraries, are marked with “partial support”. A less flexible approach is to delay the actual data sharing process and allow the user to manually inspect data that is deemed shareable (requirement CR7). This requirement was considered as fully supported by those systems that permit the specification of time-based rules. Surprisingly, while most access control systems support the notion of time, sometimes it was not clear how to precisely specify when a rule should take effect. Requirement CR8 concerns the sharing of memory cues for a limited time only, with the possibility of revoking access to such data. While we may not be able to prevent others from accessing data that has already been downloaded on their repository, we want to prevent further downloads to the data in question from our repository. Access control systems that support access revocation were marked with “full support”. Systems that can be extended to efficiently implement such functionality were marked with “partial support”. Requirement CR9 was considered as fully supported by those systems that both allow one to specify obligations that data recipients have to perform immediately after getting access to shared data, as well provide mechanisms to enforce them. However, most access control systems provide a limited support for sharing obligations, and they do not elaborate further on how such obligations can be actually enforced. Looking at Table 6.2, we can see that several access control systems accommodate the specification of temporal constraints. Contrary to this, very little work accounts for integrating obfuscation mechanisms for visual data, as well as enabling conditional data sharing.

Finally, requirement CR10 relates to the governance of data captured during collaborative events (e.g., during a meeting). Access control systems that allow sharing decisions to be specified by all involved stakeholders, with the final decision being derived by a simple approach (e.g., by delegating it to a single entity,

or by selecting the most restrictive preference) are marked with “partial support”. A fully supported approach should consider more complex governance models: accounting for the relationship of the resource and stakeholders (e.g., a data producer or a person that appears in one’s data). As we can see from Table 6.2, only few proposals offer partial support to data governance models.

## 6.4 Research Challenges

Research results from the field of context-aware access control systems highlight a growing maturity of such mechanisms to adapt to the needs of dynamic pervasive systems. Despite this, there are still some challenges that require research attention before any access control solution can be seamlessly applied in the context of pervasive memory augmentation systems. In this section, we summarize what we believe are the key challenges.

- **Lack of fully integrated solutions.** Our study has revealed lack of comprehensive access control systems that incorporate data obfuscation capabilities. We identified several data obfuscation systems tailored in particular to visual lifelogging data. All these systems can detect and subsequently blur sensitive elements such as computer screens, faces, sensitive places, or even remove high-level activities. However, very few of the access control systems that we surveyed accounted for such mechanisms that can transform data prior to granting access.

As a result, we identified two simple ways how access control systems can be extended with such functionality. One way is to delay the sharing of a resource, thus giving enough time to its owner to manually remove any sensitive information. However, constantly reviewing shared data can be a tedious activity and puts a lot of burden on users.

The other approach relies on the fact that most access control systems support the execution of external functions. In principle, this feature can be used to allow them to execute any data filtering operation. Nevertheless, this is far from being ideal. External functions may not always be visible when specifying control policies, thus preventing users to write full-fledged access control policies.

More efforts are needed to close this gap and design access control systems that can seamlessly handle data obfuscation operations.



- **Capturing and enforcing preferences from multiple-stakeholders.**

We identified several research efforts of access control systems that support multiple-user governance models. These systems can be used in collaborative scenarios where captured data is exchanged among co-located peers that experience a same event together. In this case, each involved user should be considered as owner of the produced data, and thus should have a say in specifying data sharing preferences. However, there are two challenges that have not been fully addressed so far.

The first such challenge concerns how preferences from multiple-stakeholders can be efficiently captured. One possible way of addressing this is through the gesture-based MemStone interface that we presented in Chapter 5. However, additional work is needed to resolve issues raised by conflicting preferences, where co-located peers can have a different view on how event's data can be captured, but also with whom such data can be disseminated beyond the attendees of the event.

A second challenge, which is often overlooked by access control solutions, is how to enforce such multi-party preferences. In this vein, Ilia et al. propose a multi-party access control system for photo sharing on social network sites [267]. Their system will first identify people appearing in an image, will fetch their privacy preferences, before finally deciding whether to grant or deny access on a particular image. In another work, Kapadia et al. [279] anticipate that voting schemes based on secure multiparty computation (SMC) [280] can be helpful in devising a consensus when dealing with group policies. We believe that both these approaches may be helpful in designing a potential a group-based governance model for memory augmentation systems.

- **Controlling the Presentation of Memory Cues** Beyond the need for influencing the collection and sharing of personal memory streams, users will also need to control the presentation of these memories. Users will not have the time to explicitly and manually review their captured memory cues (a wearable camera will typically capture about 1,500 pictures in a day). Instead, a more practical approach is to review a compressed summary of a prior experience (e.g., the top ten pictures coupled with few keywords that summarize a discussion). Consequently, users should specify the memory goals at a high level of abstraction (e.g., faces of new encounters, eating healthy), and this should impact the cues that are created as triggers in order to support the memories associated with the specified high level goals.

From users' point of view, it would be beneficial if one could use the same policy language for the specification of memory goals, cue presentation, and access control. Ultimately, any such solution would allow one to specify full-fledged policies for controlling their memory augmentation system.

## 6.5 Chapter Summary

This chapter presents a detailed analysis of novel and context-aware access control solutions tailored to emerging pervasive systems. Our goal is to inform and motivate further research on the design and development of access control solutions suitable for pervasive memory augmentation systems.

At the outset we identified several requirements necessary for controlling different aspects related to the capturing and sharing of experience data. Our work has revealed many interesting access control approaches that meet the requirements but to a varying extent.

Our results show that context-based control techniques can express a wide range of access policies through their rich context-based reasoning. However, at the same time, they introduce additional security concerns should the authenticity and integrity of contextual data be compromised.

Several systems supported well the possibility of sharing data based on interpersonal relationships. Their main strong point is that they extend the traditional role-based access control (RBAC) model to also support dynamic and context-dependent roles.

Furthermore, our analysis uncovered at least three different trends for supporting a multiple-user governance models. The first such trend was to delegate all access control decisions to a trusted entity, which in turn is selected from a majority voting process. A second trend allows all involved users to specify their own permissions. Access then is granted only if *all* user permissions allow that. A third trend proposes the use of selective encryption when sharing photographs. Nevertheless, we highlight that more work is needed for a seamless negotiation of group-based access policies.

We also analyzed the possibility of extending access control systems with data obfuscation mechanisms. We found out that there is a considerable amount of work on recognizing and eventually obfuscating sensitive information depicted in visual data. However, more work is needed to actually integrate such solutions into existing access control systems with the ultimate goal of offering a seamless experience to users when controlling their captured experiences.

---

To summarize, our study uncovers a growing interest and maturity of proposed solutions in this area. There are many interesting access control approaches that meet some of the requirements but not all of them. However, in spite of such progress, there is still work to be done in devising an ultimate access control solution that will fully support all such requirements. Consequently, we also briefly discussed the key challenges and gaps that we have identified from our analysis, including both technical and usability challenges. Finally, we acknowledge that this work is inevitably incomplete due to the broad literature on access control systems.



## **Part IV**

# **Conclusion and Future Work**



# Chapter 7

## Conclusion and Future Work

Human memory is a complex system and there are still many unknowns that would allow us to completely understand its workings. One very important aspect related to the remembrance of past memories is that such process can be influenced by external stimuli. For instance, imagine how reviewing a photograph can help us recall many details of our last summer vacation.

This premise has in fact fueled many scientific writers and researchers in speculating how future technology can be put to a good use in creating effective memory stimuli. However, recent technological advances allow us to think more pragmatically about the design of memory augmentation systems, the benefits that it would bring to users, but also *to reason about the challenges and issues that are raised from it*. In this thesis we focused on the later aspect, that is, we studied two chief issues stemming from this kind of technology: 1) the threat of manipulating users' memories and 2) the risk of jeopardizing users' privacy. We investigated techniques and solutions to address these two threats with a goal of designing secure systems for human memory augmentation.

This thesis started with a description of pervasive memory augmentations systems (Chapter 1), followed by a presentation of the technical background that is necessary to realize such technology (Chapter 2). In a nutshell, pervasive memory augmentation embodies a three step process. Captured experience data (step 1), e.g., pictures captured in a day, are used to generate a set of stimuli or memory cues (step 2), which are then delivered back to users through ambient-fashion displays (step 3). By constantly reviewing such data, we can train our memories of past experiences. Ultimately, prior memories can be recalled without the help of any tool.

Beyond improved recall, reviewing a set of memory cues can also speed up the forgetting of memories of other related experiences that one did not review. While there is no inherent problem with our ability to forget certain past experiences, however, the fact that such system can be used to both reinforce and attenuate one's memories opens the door to malicious memory manipulation attacks. Anyone who can inject, modify, or even delete information from one's digital memory repository can implant fake memories that do not accurately reflect a past event, or that are related to a non-existent event. Therefore, in Chapter 3 we investigated the question *"How can we guarantee digital memory integrity and provenance to prevent memory manipulation attacks?"* (RQ1). We presented our solution based on a secure wearable camera coupled with a trusted computing platform (TPM). The camera not only ensures the provenance and integrity of data that it captures, but it also prevents the unnoticed deletion of data from one's repository through a custom protocol that we designed for this purpose.

In Chapter 4 we investigated the possibility of exchanging captured data with co-located peers. This is motivated by in fact that wearable cameras which often fail to capture important elements of an experience. However, cameras of peers or any fixed infrastructure camera, may well capture key elements of the scene that one's own camera did not. Therefore, with a view towards the secure sharing of captured experiences this investigation aimed at answering the question *"How can we seamlessly and securely share captured experiences with co-located others, avoiding the risk of accidental oversharing, i.e., sharing with the wrong audience, or sharing parts of a capture that we would otherwise have kept to ourselves?"* (RQ2). To this end, we designed a mobile system that seamlessly and automatically exchanges data with other peers that are in close physical proximity to a user. The secure wearable camera that we built is an integral part of this system.

Beyond the risk of accidental data disclosure, the sharing of memory cues opens the door to two more issues. First, any data that one obtains from others increases the attack vector on memory manipulation. Trusted co-located others can maliciously share fabricated images allegedly featuring an accurate reflection of a "real" experience. To answer the question *"How can we verify the integrity and provenance of experience data which we obtain from others to detect the sharing of falsified experience captures?"* (RQ3), we designed a protocol that allows a data recipient to verify if the such data has been captured by a trusted camera, and what modifications (if any) the sender performed on it in prior to sharing.

Secondly, the ability to automatically share captured data can seriously jeopardize users' privacy. Therefore, in the final part of this thesis (Chapters 5 and 6) we shift our focus towards investigating fine-grained access control solutions. With a goal of putting users in control of their digital memories, we set out to



answer the question “*What interfaces and policy-based access control models can we use to exercise control over data capture as well as to prevent the disclosure of private and sensitive information when sharing experience data?*” (RQ4). To this end, we designed and built a tangible user interface (TUI) for controlling different aspects related to the capturing and sharing of memories through in-situ gestures. We presented the interface and results of a user study in Chapter 5.

Beyond the actual interface for the access control, this requires the availability of an access control mechanism that can both express and enforce user access preferences and policies. Therefore, in Chapter 6, we delineated a set of finer-grained access control requirements. We then surveyed existing access control mechanisms and evaluated their suitability towards such requirements.

## 7.1 Summary of Contributions and Results

In this thesis we followed a cross-disciplinary research approach from computer security, performance evaluation of computer systems and user-centered research to answer four research questions. In the following we summarize the contributions made with regard to the research questions, and conclude with an outlook on future research directions.

### 7.1.1 Securely Capturing and Storing of Experience Data

Prior research from the fields of Neuroscience and Cognitive Psychology suggests that our memories can be manipulated with the help of fake stimuli. To this end, we presented a secure wearable camera based on a trusted computing platform (TPM). Our camera prevents an adversary from creating fabricated memories by means of injecting fake experience data, modifying and deleting existing data from a victim’s repository. Injection and modification attacks are prevented by means of digital signatures using keys that are securely sealed in the device’s TPM. However, the deletion attack cannot be prevented this way. Thus, we designed a custom protocol which joins captured images in a secure data structure, thus thwarting surreptitious data deletion attempts from compromised memory repositories. We designed this protocol to efficiently run on the resource-constrained camera device. Evaluation results show that the proposed scheme can efficiently run in low-power embedded cameras: a high-resolution image ( $4096 \times 3072$  pixels) is captured and added to a secure chain in less than 45 seconds. We further showed that verifying the integrity of a chain of images captured in a day can be done in about 22 seconds, or about 670 seconds for verifying longer chains of approximately one month worth of images.

### 7.1.2 Secure Memory Sharing with Co-located Others

With a goal of improving quality of captured experiences, in Chapter 4 we investigated the possibility of sharing captured data among co-located peers. Therefore, we extend our camera design from Chapter 3 with two additional protocols. The first protocol enables the sharing of captured data with all peers that are in a close physical proximity with a user. Sharing will stop once one leaves the proximity area. As part of this process, a users' camera periodically sends updated access tokens and temporal public keys. Such tokens are broadcast by means of the short-range BLE technology. They are encrypted with the public keys of all co-present peers. Consequently, only those who possess the valid private key will be able to access the shared information. We measured two essential aspects of this protocol, that is, the token detection range, and the token exchange rate. When transmitting the radio-packets with the lowest possible power level, we were able to reduce the BLE detection range to 7 meters in open unobstructed environments and 3 meters in closed office-like spaces. Devices can reliably exchange access tokens and public keys with a maximum rate of 0.3 tokens/s and 0.12 keys/s, respectively. In practice, this means that once two devices are in range, it should take no more than 9 seconds (average 4.15 seconds) for them to start exchanging images. Tokens should therefore not be updated more frequently than once every 3 seconds, otherwise a peer may miss a token (and thus be unable to access captured data for this period).

### 7.1.3 Verifying Shared but Modified Visual Cues

To protect the sharer's privacy, the system will only share a token in real-time, and postpone the actual disclosure of the shared data (e.g., an image) to a later time. The sharer can then obfuscate any sensitive information contained in the image prior to sharing it with others. To prevent any malicious modifications in this case (i.e., aimed at providing fake data to others), our second protocol allows recipients to verify the integrity of the data they obtain from others. This is achieved by computing access tokens as a function of the actual image content that was just captured. This allows the data sharer to not only regulate access to the image, but to also "commit" the image's content publicly without actually sharing the original image itself. By furthermore signing tokens with the camera's private-key, we can ensure image authenticity. Such "custom" tokens allow one to support the verification of modifications, e.g., obfuscations, to a certain unmodified (but not shared) image. Our tests show that the camera can compute a token (i.e., a modification proof) for a low-resolution image

in about 12 seconds, while 16 seconds are needed for a high-resolution image. When including the runtime overhead of the other schemes from Chapter 3, a low-resolution image can be captured and processed in less than 25 seconds, whereas processing a high-resolution image takes about 60 seconds. All in all, capturing one low-resolution image per minute, we measured that the camera can be operational from 40 hours (with battery of 200 mAh) up to 100 hours (battery of 500 mAh) on a single charge. As for capturing high-resolution photos, the camera can run between 30 hours and 75 hours.

### 7.1.4 In-situ Controls for Memory Capture and Sharing

In Chapter 5 we presented MemStone: a tangible user interface (TUI) for controlling capture and sharing of experience data. We designed MemStone inspired by a mix of both practical and theoretic knowledge, as well as design principles regarding interactive products. MemStone is operated by five physical gestures, where each gesture performs a different control action. For instance, by putting the MemStone on a stand-on-side position, one captures data for oneself but does not share it with others. By shaking the MemStone, one can delete data captured in the last 30 seconds. One can also see the device's current operation (i.e., how many peers one is sharing data with) by looking at its central screen. We administered a user study with a goal of evaluating the device's usability, and comparing it against a more traditional smartphone app designed for the same purpose. We first conducted a lab study with 20 participants following a meeting capture scenario. We found that our participants were significantly quicker in performing data capturing and sharing controls using MemStone than using a mobile app interface. The concept was highly valued by the participants, it was perceived as user-friendly, quick to learn, and easy and fun to use. Participants also expressed a positive attitude towards the physical gestures and their relationship with the control actions. From a follow up study, conducted four months after the prior one, we found out that participants were able to remember the control gestures even after a long time period. This suggests that such a TUI is suitable to be used also in less frequently occurring events. Results from our study resonate well with findings from prior research that uncover lifeloggers' affinity to in-situ control interfaces. However, in spite of the better performance and the high perceived value of tangible-based control devices, participants were very much divided about the convenience of having to carry an additional personal control device with them for their everyday activities. This could be well attributed to lack of frequent data capture practices, with participants failing to perceive the benefits of a dedicated control interface.

### 7.1.5 A Review of the Suitability of Access Control Models for Memory Augmentation Systems

In Chapter 6 we set out with the goal of investigating the suitability of emerging class context-aware access control solutions with respect to requirements of memory augmentation systems. There are many interesting access control approaches that meet some of the requirements but not all of them. For example, we found out that most systems that we surveyed can express a wide range of access policies through their rich context-based reasoning. The possibility to share data based on interpersonal relationships was also well supported by several access control systems. Furthermore, our analysis shows that there is a considerable amount of work on data obfuscation for privacy reasons, but it is not easy to integrate them with existing access control systems. All in all, our study uncovers a growing interest and maturity of proposed access control solutions. There is still work to be done in devising an ultimate access control solution that will fully support all such requirements.

## 7.2 Future Work

This thesis provides a set of tools and protocols for designing secure systems for human memory augmentation. In the course of this research, we identified new challenges that are beyond the scope of this thesis. In the following we point out some directions for future research.

### 7.2.1 Additional Memory Manipulation Threats

In the first part of this thesis (i.e., Chapters 3 and 4) we primarily focused on security threats that can happen at the stages of experience capture, data sharing, and data storage. Beyond these stages, we foresee additional threats that can endanger one's memories. Specifically, these threats target the memory cue selection and cue presentation stages.

#### Manipulating the cue selection process

A compromised repository would permit an attacker to also influence the *memory cue selection process*. While the underlying experiences will be unaltered, they will be carefully selected in such a way that reviewing them will re-enforce particular memories of an event while attenuating other memories of the same event (i.e., recall-induced forgetting). We envision a similar adversary model

as with the previously introduced threats T2, i.e., (i) the repository service provider, or (ii) a third-party that can compromise a memory repository.

One research direction that can lead to interesting findings is to provide some indicators that will describe the working of the cue selection process. In fact, both Bettini et al. [18] and Knowles [107] see the availability of transparency tools as an essential prerequisite of any privacy-protecting system. This is further amplified in systems that influence the selection of memory cues that one will review, ultimately shaping how one remembers a particular event.

Prior research highlights that making systems explain their actions through intelligible explanations can improve user understanding and trust [281, 282, 283]. In this vein, Lim and Dey [284, 285] investigate how such intelligibility mechanisms can be designed. They focus on answering questions *Why?* or *Why not?* did a system behave in a certain way. Whilst their findings are about a specific scenario (i.e., a social-aware mobile application which shares people's availability status), they nevertheless offer great insights into designing for transparency of context-aware systems in general. In fact, this can motivate promising research efforts on designing for intelligible explanations tailored to memory augmentation systems.

### Visualizing fake memory cues

Display devices can also be hacked to manipulate what we see on our screens. A team of computer security researchers lead by Ang Cui have found a way to hack into a particular model of Dell computer monitors, leaving nearly 1 billion monitors vulnerable to this attack [286, 287]. By injecting a malware, researchers were able to not only read values of the pixels but they could also overwrite any pixel color. In light of this, attackers could present us fake memory cues and even control their presentation schedule. The envisioned adversary is any third-party that is able to control our display devices.

One possible way to address this sort of attack would be to adopt our concept from the trusted wearable camera. As a result, we can ensure the security of the display's firmware through a trusted hardware platform. However we foresee a number of challenges that require carefully design solution. First, any such solution will need to target a number of different display devices – including picture frames, smartphones, smartwatches, public displays, TV sets, etc. – each with different software characteristics and properties. Second, to verify if a display firmware is intact one has to compare it with a target firmware which is attested by a trusted entity. With many different displays devices from different manufactures, this would require a scalable infrastructure (similar to a public-key infrastructure) to deploy the trusted firmware states.

### 7.2.2 Verifying More Image Modifications Beyond Blurring

In Chapter 4 we presented a two-party protocol that allows a recipient to verify the integrity and provenance of visual cues shared by others. Since we were designing for resource constrained device, our solution accounts for only one type of image modification, i.e., the blocking or blurring of sensitive parts of an image. However, real-life scenarios entail more operations than this. For instance, one may want to compress an image prior to sharing it. In other cases one may perform cosmetic changes such as adjusting the colors of the image or increasing the intensity of light. While there is nothing inherently wrong with such modification, our algorithm would not be able to reassure the receiver that the changes were purely cosmetic – it would simply mark all images as “changed”.

To accommodate for more image modifications, in Chapter 4, we also outlined one possible solution based on homomorphic encryption. While any such approach is at the moment too computationally expensive for low-power mobile sensors, we still believe that homomorphic encryption presents a very natural candidate for addressing this challenge. One promising approach is to investigate the possibility of extending the trusted domain beyond the camera itself, and subsequently offloading the expensive computation to more powerful devices. The concept of *fog computing* [288, 289] offers a great opportunity for realizing such setup.

### 7.2.3 Specifying Control Policies Through Abstractions

In Chapter 6 we investigated the suitability of existing policy-driven access solutions with respect to our security requirements of memory augmentation systems. However, we did not look at how user policies can be specified. In fact, policy specification is an ongoing field of research and there exist several policy languages that enable one to specify both security and privacy policies [290]. For instance, logic-based languages have proven to be very attractive for this purpose. However, most of these approaches are intended for system administrators and trained personnel, and they can be very complex for end-users.

To tackle this complexity, prior research proposes to model complex privacy policies into simplified metaphors. For instance, Lederer et al. [291] present one such metaphor where privacy policies are encoded as *situational faces*. For instance, when running into a situation which would require user intervention (e.g., giving consent for an event to be recorded) one can select a preferred face, i.e., an abstraction of a permutation of privacy preferences for the encountered situation. Another intriguing system (which we presented in Chapter 6) uses the

virtual wall metaphor. Similarly as with real walls from physical world, users can place different virtual walls to control how can read their data.

We believe that this promising idea should be adopted in the context of memory augmentation systems. In this regard, we think that results of our MemStone interface and its five physical controls can inform and motivate future research efforts in designing interesting policy abstractions, with the ultimate goal of simplifying the specification of memory control policies.





# Bibliography

- [1] C. D. Conrad, “A critical review of chronic stress effects on spatial learning and memory,” *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, vol. 34, no. 5, pp. 742–755, Jun. 2010.
- [2] V. Bush, “As we may think,” *The growth of knowledge: readings on organization and retrieval of information*, pp. 23–35, 1967.
- [3] J. Gemmell, G. Bell, and R. Lueder, “MyLifeBits: A Personal Database for Everything,” *Commun. ACM*, vol. 49, no. 1, pp. 88–95, Jan. 2006.
- [4] M. Langheinrich, “Privacy by Design — Principles of Privacy-Aware Ubiquitous Systems,” in *UbiComp 2001: Ubiquitous Computing*, ser. Lecture Notes in Computer Science, G. D. Abowd, B. Brumitt, and S. Shafer, Eds. Springer Berlin Heidelberg, 2001, no. 2201, pp. 273–291, [http://link.springer.com/chapter/10.1007/3-540-45427-6\\_23](http://link.springer.com/chapter/10.1007/3-540-45427-6_23).
- [5] M. C. Anderson, R. A. Bjork, and E. L. Bjork, “Remembering can cause forgetting: Retrieval dynamics in long-term memory,” *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 20, no. 5, pp. 1063–1087, 1994.
- [6] C. Cinel, C. Cortis Mack, and G. Ward, “Towards augmented human memory: Retrieval-induced forgetting and retrieval practice in an interactive, end-of-day review.” *Journal of Experimental Psychology: General*, vol. 147, no. 5, pp. 632–661, May 2018.
- [7] D. J. Shaw, *The Memory Illusion: Remembering, Forgetting, and the Science of False Memory*. Random House, Jun. 2016.
- [8] C. Gurrin, A. F. Smeaton, and A. R. Doherty, “LifeLogging: Personal Big Data,” *Found. Trends Inf. Retr.*, vol. 8, no. 1, pp. 1–125, Jun. 2014.

- [9] N. Davies, A. Friday, S. Clinch, C. Sas, M. Langheinrich, G. Ward, and A. Schmidt, "Security and Privacy Implications of Pervasive Memory Augmentation," *IEEE Pervasive Computing*, vol. 14, no. 1, pp. 44–53, Jan. 2015.
- [10] A. D. Baddeley, M. Kopelman, and B. A. Wilson, *The Essential Handbook of Memory Disorders for Clinicians*. John Wiley & Sons, Jul. 2004.
- [11] K. Wolf, A. Schmidt, A. Bexheti, and M. Langheinrich, "Lifelogging: You're Wearing a Camera?" *IEEE Pervasive Computing*, vol. 13, no. 3, pp. 8–12, Jul. 2014.
- [12] T. Dingler, P. E. Agroudy, H. V. Le, A. Schmidt, E. Niforatos, A. Bexheti, and M. Langheinrich, "Multimedia Memory Cues for Augmenting Human Memory," *IEEE MultiMedia*, vol. 23, no. 2, pp. 4–11, Apr. 2016.
- [13] A. Bexheti, E. Niforatos, S. A. Bahrainian, M. Langheinrich, and F. Crestani, "Measuring the Effect of Cued Recall on Work Meetings," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. New York, NY, USA: ACM, 2016, pp. 1020–1026.
- [14] A. Bexheti, M. Langheinrich, and S. Clinch, "Secure Personal Memory-Sharing with Co-located People and Places," in *Proceedings of the 6th International Conference on the Internet of Things*, ser. IoT'16. New York, NY, USA: ACM, 2016, pp. 73–81.
- [15] T. Dingler, A. Bexheti, E. Niforatos, and F. Alt, "Workshop on Mobile Cognition: Using Mobile Devices to Enhance Human Cognition," in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, ser. MobileHCI '15. New York, NY, USA: ACM, 2015, pp. 970–973.
- [16] F. Stajano, "Security Issues in Ubiquitous Computing\*," in *Handbook of Ambient Intelligence and Smart Environments*, H. Nakashima, H. Aghajan, and J. C. Augusto, Eds. Boston, MA: Springer US, 2010, pp. 281–314, [http://link.springer.com/10.1007/978-0-387-93808-0\\_11](http://link.springer.com/10.1007/978-0-387-93808-0_11).
- [17] R. K. Thomas and R. Sandhu, "Models, protocols, and architectures for secure pervasive computing: Challenges and research directions," in *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*, Mar. 2004, pp. 164–168.

- [18] C. Bettini and D. Riboni, "Privacy protection in pervasive systems: State of the art and technical challenges," *Pervasive and Mobile Computing*, Oct. 2014.
- [19] A. Bexheti, M. Langheinrich, I. Elhart, and N. Davies, "Securely Storing and Sharing Memory Cues in Memory Augmentation Systems: A Practical Approach," in *Proceedings of the 17th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'19)*, 2019, p. 10.
- [20] A. Bexheti, A. Fedosov, I. Elhart, and M. Langheinrich, "Memstone: A Tangible Interface for Controlling Capture and Sharing of Personal Memories," in *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '18. New York, NY, USA: ACM, 2018, pp. 20:1–20:13.
- [21] A. Bexheti and M. Langheinrich, "Understanding Usage Control Requirements in Pervasive Memory Augmentation Systems," in *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '15. New York, NY, USA: ACM, 2015, pp. 400–404.
- [22] R. J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 2nd ed. Tokyo, New York: Wiley, Apr. 2008.
- [23] W. Stallings and L. Brown, *Computer Security: Principles and Practice*, third edition ed., ser. The William Stallings Books on Computer and Data Communications Technology. Boston: Pearson, 2015.
- [24] B. W. Lampson, "Computer Security in the real world," in *IEEE Computer*, 2000, pp. 37–46.
- [25] D. G. Feitelson, *Workload Modeling for Computer Systems Performance Evaluation*. Cambridge: Cambridge University Press, 2015, <http://ebooks.cambridge.org/ref/id/CBO9781139939690>.
- [26] J.-Y. Le Boudec, *Performance Evaluation of Computer and Communication Systems*, 1st ed., ser. Computer and Communication Sciences. Lausanne: EPFL Press, 2010, oCLC: 723514457.
- [27] "ISO 9241-210:2019, Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems," in *International Standardization Organization (ISO)*, Geneva, Switzerland, 2019.

- [28] A. Fedosov, A. Bexheti, E. Ermolaev, and M. Langheinrich, "Sharing Physical Objects Using Smart Contracts," in *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, ser. MobileHCI '18. New York, NY, USA: ACM, 2018, pp. 346–352.
- [29] E. Niforatos, M. Laporte, A. Bexheti, and M. Langheinrich, "Augmenting Memory Recall in Work Meetings: Establishing a Quantifiable Baseline," in *Proceedings of the 9th Augmented Human International Conference*, ser. AH '18. New York, NY, USA: ACM, 2018, pp. 4:1–4:7.
- [30] E. Niforatos, V. Lim, C. Vuerich, M. Langheinrich, and A. Bexheti, "Pulse-Cam: Biophysically Driven Life Logging," in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, ser. MobileHCI '15. New York, NY, USA: ACM, 2015, pp. 1002–1009.
- [31] A. Bexheti, A. Fedosov, J. Findahl, M. Langheinrich, and E. Niforatos, "Re-Live the Moment: Visualizing Run Experiences to Motivate Future Exercises," in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, ser. MobileHCI '15. New York, NY, USA: ACM, 2015, pp. 986–993.
- [32] E. Niforatos, M. Langheinrich, and A. Bexheti, "My Good Old Kodak: Understanding the Impact of Having Only 24 Pictures to Take," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, ser. UbiComp '14 Adjunct. New York, NY, USA: ACM, 2014, pp. 1355–1360.
- [33] M. Weiser, "The Computer for the 21st Century," p. 8.
- [34] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," in *Handheld and Ubiquitous Computing*, G. Goos, J. Hartmanis, J. van Leeuwen, and H.-W. Gellersen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, vol. 1707, pp. 304–307, [http://link.springer.com/10.1007/3-540-48157-5\\_29](http://link.springer.com/10.1007/3-540-48157-5_29).
- [35] A. Dey, "Context-Aware Computing: The CyberDesk Project," p. 4.

- [36] P. D. I. Group, “Alexa, Schedule a Doctor’s Appointment,” <https://medium.com/digital-health-innovation/alexa-schedule-a-doctors-appointment-38dbf6febda1>, Apr. 2019.
- [37] M. Dodge and R. Kitchin, “‘Outlines of a World Coming into Existence’: Pervasive Computing and the Ethics of Forgetting,” *Environment and Planning B: Planning and Design*, vol. 34, no. 3, pp. 431–445, Jun. 2007.
- [38] S. Mann, “‘WearCam’ (The wearable camera): Personal imaging systems for long-term use in wearable tetherless computer-mediated reality and personal photo/videographic memory prosthesis,” in *Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215)*, Oct. 1998, pp. 124–131.
- [39] S. Mann, J. Fung, and C. Aimone, “Designing EyeTap Digital Eyeglasses for Continuous Lifelong Capture and Sharing of Personal Experiences,” 2005.
- [40] J. Gemmell, L. Williams, K. Wood, R. Lueder, and G. Bell, “Passive Capture and Ensuing Issues for a Personal Lifetime Store,” in *Proceedings of the the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*, ser. CARPE’04. New York, NY, USA: ACM, 2004, pp. 48–55.
- [41] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Wood, “SenseCam: A Retrospective Memory Aid,” in *UbiComp 2006: Ubiquitous Computing*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, P. Dourish, and A. Friday, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 4206, pp. 177–193, [http://link.springer.com/10.1007/11853565\\_11](http://link.springer.com/10.1007/11853565_11).
- [42] K. Kunze, M. Iwamura, K. Kise, S. Uchida, and S. Omachi, “Activity Recognition for the Mind: Toward a Cognitive ‘Quantified Self’,” *Computer*, vol. 46, no. 10, pp. 105–108, Oct. 2013.
- [43] J. Meyer, S. Simske, K. A. Siek, C. G. Gurrin, and H. Hermens, “Beyond quantified self: Data for wellbeing,” in *Proceedings of the Extended Abstracts of the 32nd Annual ACM Conference on Human Factors in Computing Systems - CHI EA ’14*. Toronto, Ontario, Canada: ACM Press, 2014, pp. 95–98.

- [44] M. Hughes, E. Newman, N. O'Hare, Z. Zhang, A. F. Smeaton, N. E. O'Connor, and G. Farrell, "A Lifelogging Approach to Automated Market Research," p. 2.
- [45] D. Byrne, A. R. Doherty, G. J. F. Jones, A. F. Smeaton, and K. Järvelin, "The SenseCam as a Tool for Task Observation," p. 4.
- [46] A. J. Sellen, A. Fogg, M. Aitken, S. Hodges, C. Rother, and K. Wood, "Do Life-logging Technologies Support Memory for the Past?: An Experimental Study Using Sensecam," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '07. New York, NY, USA: ACM, 2007, pp. 81–90.
- [47] M. Crete-Nishihata, R. M. Baecker, M. Massimi, D. Ptak, R. Campigotto, L. D. Kaufman, A. M. Brickman, G. R. Turner, J. R. Steinerman, and S. E. Black, "Reconstructing the Past: Personal Memory Technologies Are Not Just Personal and Not Just for Memory," *Human-Computer Interaction*, vol. 27, no. 1-2, pp. 92–123, Apr. 2012.
- [48] A. M. Surprenant and I. Neath, *Principles of Memory*. Psychology Press, Mar. 2013.
- [49] W. Koutstaal, D. L. Schacter, M. K. Johnson, and L. Galluccio, "Facilitation and impairment of event memory produced by photograph review," *Memory & Cognition*, vol. 27, no. 3, pp. 478–493, May 1999.
- [50] E. Berry, N. Kapur, L. Williams, S. Hodges, P. Watson, G. Smyth, J. Srinivasan, R. Smith, B. Wilson, and K. Wood, "The use of a wearable camera, SenseCam, as a pictorial diary to improve autobiographical memory in a patient with limbic encephalitis: A preliminary report," *Neuropsychological Rehabilitation*, vol. 17, no. 4-5, pp. 582–601, Aug. 2007.
- [51] E. Woodberry, G. Browne, S. Hodges, P. Watson, N. Kapur, and K. Woodberry, "The use of a wearable camera improves autobiographical memory in patients with Alzheimer's disease," *Memory*, vol. 23, no. 3, pp. 340–349, Apr. 2015.
- [52] M. L. Lee and A. K. Dey, "Providing Good Memory Cues for People with Episodic Memory Impairment," in *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, ser. Assets '07. New York, NY, USA: ACM, 2007, pp. 131–138.

- [53] M. Harvey, M. Langheinrich, and G. Ward, "Remembering through lifelogging: A survey of human memory augmentation," *Pervasive and Mobile Computing*, vol. 27, pp. 14–26, 2016.
- [54] V. Kalnikaite, A. Sellen, S. Whittaker, and D. Kirk, "Now let me see where i was: Understanding how lifelogs mediate memory," in *Proceedings of the 28th International Conference on Human Factors in Computing Systems - CHI '10*. Atlanta, Georgia, USA: ACM Press, 2010, p. 2045.
- [55] D. Reisberg and F. Heuer, "Remembering the details of emotional events," in *Affect and Accuracy in Recall: Studies of "Flashbulb" Memories*, ser. Emory Symposia in Cognition, 4. New York, NY, US: Cambridge University Press, 1992, pp. 162–190.
- [56] S. Hamann, "Cognitive and neural mechanisms of emotional memory," *Trends in Cognitive Sciences*, vol. 5, no. 9, pp. 394–400, Sep. 2001.
- [57] C. Sas, T. Fratzczak, M. Rees, H. Gellersen, V. Kalnikaite, A. Coman, and K. Höök, "AffectCam: Arousal- Augmented Sensecam for Richer Recall of Episodic Memories," in *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '13. New York, NY, USA: ACM, 2013, pp. 1041–1046.
- [58] S. Clinch, N. Davies, M. Mikusz, P. Metzger, M. Langheinrich, A. Schmidt, and G. Ward, "Collecting Shared Experiences through Lifelogging: Lessons Learned," *IEEE Pervasive Computing*, vol. 15, no. 1, pp. 58–67, Jan. 2016.
- [59] M. Aslan, "How do the rules on audio recording change under the GDPR?" <https://iapp.org/news/a/how-do-the-rules-on-audio-recording-change-under-the-gdpr/>, 2018.
- [60] A. R. Doherty, K. Pauly-Takacs, N. Caprani, C. Gurrin, C. J. A. Moulin, N. E. O'Connor, and A. F. Smeaton, "Experiences of Aiding Autobiographical Memory Using the SenseCam," *Human-Computer Interaction*, vol. 27, no. 1-2, pp. 151–174, Apr. 2012.
- [61] M. A. Conway, "Episodic memories," *Neuropsychologia*, vol. 47, no. 11, pp. 2305–2313, Sep. 2009.
- [62] Y. Ezzyat and L. Davachi, "What Constitutes an Episode in Episodic Memory?" *Psychological Science*, vol. 22, no. 2, pp. 243–252, Feb. 2011.

- [63] E. Tulving and D. M. Thomson, "Encoding specificity and retrieval processes in episodic memory." *Psychological Review*, vol. 80, no. 5, pp. 352–373, 1973.
- [64] J. M. Zacks, N. K. Speer, J. M. Vettel, and L. L. Jacoby, "Event understanding and memory in healthy aging and dementia of the Alzheimer type." *Psychology and Aging*, vol. 21, no. 3, pp. 466–482, 2006.
- [65] Joo-Hwee Lim, Qi Tian, and P. Mulhem, "Home photo content modeling for personalized event-based retrieval," *IEEE Multimedia*, vol. 10, no. 4, pp. 28–37, Oct. 2003.
- [66] D. Sadlier and N. O'Connor, "Event detection in field sports video using audio-visual features and a support vector Machine," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1225–1233, Oct. 2005.
- [67] C. Xu, J. Wang, K. Wan, Y. Li, and L. Duan, "Live sports event detection based on broadcast video and web-casting text," in *Proceedings of the 14th Annual ACM International Conference on Multimedia - MULTIMEDIA '06*. Santa Barbara, CA, USA: ACM Press, 2006, p. 221.
- [68] P. Atrey, N. Maddage, and M. Kankanhalli, "Audio Based Event Detection for Multimedia Surveillance," in *2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings*, vol. 5. Toulouse, France: IEEE, 2006, pp. V-813–V-816.
- [69] G. Foresti, L. Marcenaro, and C. Regazzoni, "Automatic detection and indexing of video-event shots for surveillance applications," *IEEE Transactions on Multimedia*, vol. 4, no. 4, pp. 459–471, Dec. 2002.
- [70] Z. Wang, M. D. Hoffman, P. R. Cook, and K. Li, "VFerret: Content-based similarity search tool for continuous archived video," in *Proceedings of the 3rd ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*. ACM, Oct. 2006, pp. 19–26.
- [71] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, vol. 1, no. 1, pp. 10–28, Jan. 1993.
- [72] A. R. Doherty and A. F. Smeaton, "Automatically Segmenting LifeLog Data into Events," in *2008 Ninth International Workshop on Image Analysis for*



- Multimedia Interactive Services*. Klagenfurt, Austria: IEEE, 2008, pp. 20–23.
- [73] M. A. Hearst and C. Plaunt, “Subtopic structuring for full-length document access,” in *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Jan. 1993, pp. 59–68.
- [74] A. R. Doherty, C. J. Moulin, and A. F. Smeaton, “Automatically assisting human memory: A SenseCam browser.” *Memory (Hove, England)*, vol. 19, no. 7, pp. 785–795, Oct. 2011.
- [75] M. Harvey and D. Elswailer, “Exploring Query Patterns in Email Search,” in *Advances in Information Retrieval*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, R. Baeza-Yates, A. P. de Vries, H. Zaragoza, B. B. Cambazoglu, V. Murdock, R. Lempel, and F. Silvestri, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7224, pp. 25–36, [http://link.springer.com/10.1007/978-3-642-28997-2\\_3](http://link.springer.com/10.1007/978-3-642-28997-2_3).
- [76] L. M. Zhou, B. Moynagh, L. Zhou, T. Ye, and C. Gurrin, “MemLog, an Enhanced Lifelog Annotation and Search Tool,” in *MultiMedia Modeling*, ser. Lecture Notes in Computer Science, X. He, S. Luo, D. Tao, C. Xu, J. Yang, and M. A. Hasan, Eds. Springer International Publishing, 2015, pp. 303–306.
- [77] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, “TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation,” in *2009 IEEE 12th International Conference on Computer Vision*, Sep. 2009, pp. 309–316.
- [78] M. Chen, A. Zheng, and K. Weinberger, “Fast Image Tagging,” in *International Conference on Machine Learning*, Feb. 2013, pp. 1274–1282, <http://proceedings.mlr.press/v28/chen13j.html>.
- [79] M. Korayem, R. Templeman, D. Chen, D. Crandall, and A. Kapadia, “Enhancing Lifelogging Privacy by Detecting Screens,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’16. New York, NY, USA: ACM, 2016, pp. 4309–4314.

- [80] R. Templeman, M. Korayem, D. Crandall, and A. Kapadia, "PlaceAvider: Steering first-person cameras away from sensitive spaces," in *Network and Distributed System Security Symposium (NDSS)*, 2014.
- [81] M. Iwamura, K. Kunze, Y. Kato, Y. Utsumi, and K. Kise, "Haven't we met before?: A realistic memory assistance system to remind you of the person in front of you," in *Proceedings of the 5th Augmented Human International Conference on - AH '14*. Kobe, Japan: ACM Press, 2014, pp. 1–4.
- [82] E. Thomaz, A. Parnami, J. Bidwell, I. Essa, and G. D. Abowd, "Technological Approaches for Addressing Privacy Concerns when Recognizing Eating Behaviors with Wearable Cameras," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '13. New York, NY, USA: ACM, 2013, pp. 739–748.
- [83] D. Castro, S. Hickson, V. Bettadapura, E. Thomaz, G. Abowd, H. Christensen, and I. Essa, "Predicting daily activities from egocentric images using deep learning," in *Proceedings of the 2015 ACM International Symposium on Wearable Computers - ISWC '15*. Osaka, Japan: ACM Press, 2015, pp. 75–82.
- [84] A. Fathi, J. K. Hodgins, and J. M. Rehg, "Social interactions: A first-person perspective," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2012, pp. 1226–1233.
- [85] A. J. Sellen and S. Whittaker, "Beyond total capture: A constructive critique of lifelogging," *Communications of the ACM*, vol. 53, no. 5, p. 70, May 2010.
- [86] A. Graham, H. Garcia-Molina, A. Paepcke, T. Winograd, and T. Winograd, "Time as essence for photo browsing through personal digital libraries," in *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries*. ACM, Jul. 2002, pp. 326–335.
- [87] K. Toyama, R. Logan, and A. Roseway, "Geographic location tags on digital images," in *Proceedings of the Eleventh ACM International Conference on Multimedia*. ACM, Feb. 2003, pp. 156–166.
- [88] C. Chen, M. Oakes, and J. Tait, "Browsing Personal Images Using Episodic Memory (Time + Location)," in *Advances in Information Retrieval*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan,

- D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 3936, pp. 362–372, [http://link.springer.com/10.1007/11735106\\_32](http://link.springer.com/10.1007/11735106_32).
- [89] M. Naaman, S. Harada, Q. Wang, H. Garcia-Molina, and A. Paepcke, “Context data in geo-referenced digital photo collections,” in *Proceedings of the 12th Annual ACM International Conference on Multimedia*. ACM, Oct. 2004, pp. 196–203.
- [90] H. V. Le, S. Clinch, C. Sas, T. Dingler, N. Henze, and N. Davies, “Impact of Video Summary Viewing on Episodic Memory Recall: Design Guidelines for Video Summarizations.” ACM Press, 2016, pp. 4793–4805.
- [91] H. Lee, A. F. Smeaton, N. E. O’Connor, G. Jones, M. Blighe, D. Byrne, A. Doherty, and C. Gurrin, “Constructing a SenseCam visual diary as a media process,” *Multimedia Systems*, vol. 14, no. 6, pp. 341–349, Dec. 2008.
- [92] K. O’Hara, M. Tuffield, and N. Shadbolt, “Lifeloggging: Issues of identity and privacy with memories for life,” 2008.
- [93] C. Gurrin, R. Albatal, H. Joho, and K. Ishii, “A privacy by design approach to lifeloggging,” in *Digital Enlightenment Yearbook 2014*, K. O’Hara, C. Nguyen, and P. Haynes, Eds. The Netherlands: IOS Press, 2014, pp. 49–73.
- [94] J. Jung and M. Philipose, “Courteous Glass,” in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, ser. UbiComp ’14 Adjunct. New York, NY, USA: ACM, 2014, pp. 1307–1312.
- [95] J. Schiff, M. Meingast, D. K. Mulligan, S. Sastry, and K. Goldberg, “Respectful Cameras: Detecting Visual Markers in Real-Time to Address Privacy Concerns,” in *Protecting Privacy in Video Surveillance*, A. Senior, Ed. London: Springer London, 2009, pp. 65–89, [https://doi.org/10.1007/978-1-84882-301-3\\_5](https://doi.org/10.1007/978-1-84882-301-3_5).
- [96] K. N. Truong, S. N. Patel, J. W. Summet, and G. D. Abowd, “Preventing Camera Recording by Designing a Capture-Resistant Environment,” in *UbiComp 2005: Ubiquitous Computing*, ser. Lecture Notes in Computer

- Science, M. Beigl, S. Intille, J. Rekimoto, and H. Tokuda, Eds. Springer Berlin Heidelberg, 2005, pp. 73–86.
- [97] M. Koelle, W. Heuten, and S. Boll, “Are You Hiding It?: Usage Habits of Lifelogging Camera Wearers,” in *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI ’17. New York, NY, USA: ACM, 2017, pp. 80:1–80:8.
- [98] “Physical assault by McDonald’s for wearing Digital Eye Glass,” <http://eyetap.blogspot.com/2012/07/physical-assault-by-mcdonalds-for.html>.
- [99] S. Slocum, “Google Glass Assault and Robbery at Molotov’s Bar, Haight St. February 22, 2014,” <http://ilovesocialmediainc.blogspot.com/2014/03/google-glass-assault-and-robbery-at.html>, Mar. 2014.
- [100] R. Gray, “The places where Google Glass is banned,” Dec. 2013, <https://www.telegraph.co.uk/technology/google/10494231/The-places-where-Google-Glass-is-banned.html>.
- [101] G. Press, “Glass Explorers: Do’s and Don’ts,” <https://sites.google.com/site/glasscomms/glass-explorers>.
- [102] R. Eveleth, “Google Glass Wasn’t a Failure. It Raised Crucial Concerns | WIRED,” <https://www.wired.com/story/google-glass-reasonable-expectation-of-privacy/>, 2018.
- [103] T. Denning, Z. Dehlawi, T. Kohno, T. Denning, Z. Dehlawi, and T. Kohno, “In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies,” in *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Apr. 2014, pp. 2377–2386.
- [104] D. H. Nguyen, G. Marcu, G. R. Hayes, K. N. Truong, J. Scott, M. Langheinrich, and C. Roduner, “Encountering SenseCam: Personal recording technologies in everyday life,” in *Proceedings of the 11th International Conference on Ubiquitous Computing - UbiComp ’09*. Orlando, Florida, USA: ACM Press, 2009, p. 165.
- [105] R. Hoyle, R. Templeman, S. Armes, D. Anthony, D. Crandall, and A. Kapadia, “Privacy Behaviors of Lifeloggers Using Wearable Cameras,” in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp ’14. New York, NY, USA: ACM, 2014, pp. 571–582.

- [106] M. Koelle, K. Wolf, and S. Boll, “Beyond LED Status Lights - Design Requirements of Privacy Notices for Body-worn Cameras,” in *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, ser. TEI '18. New York, NY, USA: ACM, 2018, pp. 177–187.
- [107] B. Knowles, “Emerging Trust Implications of Data-Rich Systems,” *IEEE Pervasive Computing*, vol. 15, no. 4, pp. 76–84, Oct. 2016.
- [108] A. D. Miller and W. K. Edwards, “Give and Take: A Study of Consumer Photo-sharing Culture and Practice,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '07. New York, NY, USA: ACM, 2007, pp. 347–356.
- [109] R. Templeman, Z. Rahman, D. Crandall, and A. Kapadia, “PlaceRaider: Virtual Theft in Physical Spaces with Smartphones,” in *Proceedings of The 20th Annual Network and Distributed System Security Symposium (NDSS)*, 2013, p. 15.
- [110] R. Hoyle, R. Templeman, D. Anthony, D. Crandall, and A. Kapadia, “Sensitive Lifelogs: A Privacy Analysis of Photos from Wearable Cameras,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 1645–1648.
- [111] K. E. Caine, “Exploring everyday privacy behaviors and misclosures,” Ph.D. dissertation, Dec. 2009, <https://smartech.gatech.edu/handle/1853/31665>.
- [112] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object Detection with Discriminatively Trained Part-Based Models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
- [113] L.-J. Li and L. Fei-Fei, “What, where and who? Classifying events by scene and object recognition,” in *2007 IEEE 11th International Conference on Computer Vision*, Oct. 2007, pp. 1–8.
- [114] S. Jana, A. Narayanan, and V. Shmatikov, “A Scanner Darkly: Protecting User Privacy from Perceptual Applications,” in *2013 IEEE Symposium on Security and Privacy (SP)*, May 2013, pp. 349–363.

- [115] A. Adams, “Multimedia Information Changes the Whole Privacy Ballgame,” in *Proceedings of the Tenth Conference on Computers, Freedom and Privacy: Challenging the Assumptions*, ser. CFP '00. New York, NY, USA: ACM, 2000, pp. 25–32.
- [116] R. C. Atkinson and R. M. Shiffrin, “Human Memory: A Proposed System and its Control Processes,” *Psychology of Learning and Motivation*, vol. 2, pp. 89–195, Jan. 1968.
- [117] A. Reeves and G. Sperling, “Attention gating in short-term visual memory,” *Psychological Review*, vol. 93, no. 2, pp. 180–206, 1986.
- [118] H. Tiitinen, P. May, K. Reinikainen, and R. Näätänen, “Attentive novelty detection in humans is governed by pre-attentive sensory memory,” *Nature*, vol. 372, no. 6501, pp. 90–92, Nov. 1994.
- [119] D. E. Broadbent, “The hidden preattentive processes,” *American Psychologist*, vol. 32, no. 2, pp. 109–118, 1977.
- [120] R. C. Atkinson and R. M. Shiffrin, “The Control of Short-Term Memory,” *Scientific American*, vol. 225, no. 2, pp. 82–91, 1971, <https://www.jstor.org/stable/24922803>.
- [121] G. A. Miller, “The magical number seven, plus or minus two: Some limits on our capacity for processing information,” *Psychological Review*, vol. 63, no. 2, pp. 81–97, 1956.
- [122] H. P. Bahrick, P. O. Bahrick, and R. P. Wittlinger, “Fifty years of memory for names and faces: A cross-sectional approach,” *Journal of Experimental Psychology: General*, vol. 104, no. 1, pp. 54–75, 1975.
- [123] E. Tulving, “How Many Memory Systems Are There?” *American Psychologist*, p. 14, 1985.
- [124] —, *Organization of Memory: Episodic and Semantic Memory 1*. Academic Press New York and London, 1972.
- [125] —, “Précis of Elements of episodic memory.” *Behavioral and Brain Sciences*, vol. 7, no. 2, pp. 223–268, 1984.
- [126] M. W. Eysenck, *Principles of Cognitive Psychology*. Psychology Press, 2001.

- [127] H. Buschke, "Cued recall in Amnesia," *Journal of Clinical Neuropsychology*, vol. 6, no. 4, pp. 433–440, Nov. 1984.
- [128] L. J. Bannon, "Forgetting as a feature, not a bug: The duality of memory and implications for ubiquitous computing," *CoDesign*, vol. 2, no. 1, pp. 3–15, Mar. 2006.
- [129] V. Mayer-Schönberger, *Delete: The Virtue of Forgetting in the Digital Age*. Princeton: Princeton University Press, 2009, oCLC: ocn319868007.
- [130] C. Sas and S. Whittaker, "Design for forgetting: Disposing of digital possessions after a breakup," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. Paris, France: ACM Press, 2013, p. 1823.
- [131] L. J. Cuddy and L. L. Jacoby, "When forgetting helps memory: An analysis of repetition effects," 1982.
- [132] S. C. Mondesire and R. P. Wiegand, "Forgetting Classification and Measurement for Decomposition-based Reinforcement Learning," 2013.
- [133] D. L. Schacter, "The seven sins of memory: Insights from psychology and cognitive neuroscience." *American Psychologist*, vol. 54, no. 3, pp. 182–203, 1999.
- [134] C. A. Morgan III, S. Southwick, G. Steffian, G. A. Hazlett, and E. F. Loftus, "Misinformation can influence memory for recently experienced, highly stressful events," *International Journal of Law and Psychiatry*, vol. 36, no. 1, pp. 11–17, Jan. 2013.
- [135] L. A. Henkel, "Photograph-induced memory errors: When photographs make people claim they have done things they have not," *Applied Cognitive Psychology*, vol. 25, no. 1, pp. 78–86, Jan. 2011.
- [136] A. S. Brown and E. J. Marsh, "Evoking false beliefs about autobiographical experience," *Psychonomic Bulletin & Review*, vol. 15, no. 1, pp. 186–190, Feb. 2008.
- [137] K. A. Wade, M. Garry, J. D. Read, and D. S. Lindsay, "A picture is worth a thousand lies: Using false photographs to create false childhood memories," *Psychonomic Bulletin & Review*, vol. 9, no. 3, pp. 597–603, Sep. 2002.

- [138] D. S. Lindsay, L. Hagen, J. D. Read, K. A. Wade, and M. Garry, "True Photographs and False Memories," *Psychological Science*, vol. 15, no. 3, pp. 149–154, Mar. 2004.
- [139] G. L. Friedman, "The trustworthy digital camera: Restoring credibility to the photographic image," *IEEE Transactions on Consumer Electronics*, vol. 39, no. 4, pp. 905–910, Nov. 1993.
- [140] T. Winkler and B. Rinner, "TrustCAM: Security and Privacy-Protection for an Embedded Smart Camera Based on Trusted Computing," in *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Aug. 2010, pp. 593–600.
- [141] —, "Securing Embedded Smart Cameras with Trusted Computing," *EURASIP J. Wirel. Commun. Netw.*, vol. 2011, pp. 8:1–8:20, Jan. 2011.
- [142] T. Winkler, Á. Erdélyi, and B. Rinner, "TrustEYE.M4: Protecting the sensor – Not the camera," in *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug. 2014, pp. 159–164.
- [143] S. Saroiu and A. Wolman, "I Am a Sensor, and I Approve This Message," in *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, ser. HotMobile '10. New York, NY, USA: ACM, 2010, pp. 37–42.
- [144] K. Bcakci and N. Baykal, "Infinite length hash chains and their applications," in *Proceedings. Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2002, pp. 57–61.
- [145] Y.-c. Hu, M. Jakobsson, and A. Perrig, "Efficient Constructions for One-Way Hash Chains," 2003.
- [146] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA Broadcast Authentication Protocol," p. 11, 2002.
- [147] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: Security Protocols for Sensor Networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, Sep. 2002.
- [148] D. Bayer, S. Haber, and W. S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," *Sequences II: Methods in Communication, Security and Computer Science*, pp. 329–334, 1993.



- [149] S. Haber and W. S. Stornetta, “How to Time-Stamp a Digital Document,” in *Advances in Cryptology-CRYPTO’90*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Aug. 1990, pp. 437–455.
- [150] P. Maniatis, T. J. Giuli, and M. Baker, “Enabling the Long-Term Archival of Signed Documents through Time Stamping,” *arXiv:cs/0106058*, Jun. 2001, <http://arxiv.org/abs/cs/0106058>.
- [151] B. Parno, J. M. McCune, and A. Perrig, *Bootstrapping Trust in Modern Computers*, ser. SpringerBriefs in Computer Science. New York, NY: Springer New York, 2011, vol. 10, <http://link.springer.com/10.1007/978-1-4614-1460-5>.
- [152] K. Dietrich, M. Pirker, T. Vejda, R. Toegl, T. Winkler, and P. Lipp, “A Practical Approach for Establishing Trust Relationships between Remote Platforms Using Trusted Computing,” in *Trustworthy Global Computing*, G. Barthe and C. Fournet, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 4912, pp. 156–168, [http://link.springer.com/10.1007/978-3-540-78663-4\\_12](http://link.springer.com/10.1007/978-3-540-78663-4_12).
- [153] M. Bellare, R. Canetti, and H. Krawczyk, “Keying Hash Functions for Message Authentication,” in *Advances in Cryptology – CRYPTO’96*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Aug. 1996, pp. 1–15.
- [154] T. Hardjono and G. Kazmierczak, “Overview of the TPM Key Management Standard,” p. 15, 2008, [http://staging.trustedcomputinggroup.org/wp-content/uploads/Kazmierczak20Greg20-20TPM\\_Key\\_Management\\_KMS2008\\_v003.pdf](http://staging.trustedcomputinggroup.org/wp-content/uploads/Kazmierczak20Greg20-20TPM_Key_Management_KMS2008_v003.pdf).
- [155] M. Bellare, “New Proofs for NMAC and HMAC: Security Without Collision-Resistance,” in *Advances in Cryptology - CRYPTO 2006*, ser. Lecture Notes in Computer Science, C. Dwork, Ed. Springer Berlin Heidelberg, 2006, pp. 602–619.
- [156] A. Naveh and E. Tromer, “PhotoProof: Cryptographic Image Authentication for Any Set of Permissible Transformations,” in *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016, pp. 255–271.
- [157] J. Lukas, J. Fridrich, and M. Goljan, “Digital camera identification from sensor pattern noise,” *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, Jun. 2006.

- [158] K. O'Hara, M. M. Tuffield, and N. Shadbolt, "Lifelogging: Privacy and empowerment with memories for life," *Identity in the Information Society*, vol. 1, no. 1, pp. 155–172, Feb. 2009.
- [159] S. Clinch, P. Metzger, and N. Davies, "Lifelogging for 'Observer' View Memories: An Infrastructure Approach," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, ser. UbiComp'14 Adjunct. New York, NY, USA: ACM, 2014, pp. 1397–1404.
- [160] A. Amir, A. Efrat, J. Myllymaki, L. Palaniappan, and K. Wampler, "Buddy tracking — efficient proximity detection among mobile friends," *Pervasive and Mobile Computing*, vol. 3, no. 5, pp. 489–511, Oct. 2007.
- [161] A. Küpper and G. Treu, "Efficient Proximity and Separation Detection Among Mobile Targets for Supporting Location-based Community Services," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 10, no. 3, pp. 1–12, Jul. 2006.
- [162] L. Šikšnys, J. R. Thomsen, S. Šaltenis, and M. L. Yiu, "Private and Flexible Proximity Detection in Mobile Social Networks," in *2010 Eleventh International Conference on Mobile Data Management*, May 2010, pp. 75–84.
- [163] P. Sapiezynski, A. Stopczynski, D. K. Wind, J. Leskovec, and S. Lehmann, "Inferring Person-to-person Proximity Using WiFi Signals," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 24:1–24:20, Jun. 2017.
- [164] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place Lab: Device Positioning Using Radio Beacons in the Wild," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, H. W. Gellersen, R. Want, and A. Schmidt, Eds. Springer Berlin Heidelberg, 2005, pp. 116–133.
- [165] J. Krumm and K. Hinckley, "The NearMe Wireless Proximity Server," in *UbiComp 2004: Ubiquitous Computing*, ser. Lecture Notes in Computer Science, N. Davies, E. D. Mynatt, and I. Siio, Eds. Springer Berlin Heidelberg, 2004, pp. 283–300.
- [166] K. A. Li, T. Y. Sohn, S. Huang, and W. G. Griswold, "Peopletones: A System for the Detection and Notification of Buddy Proximity on Mobile Phones,"

- in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '08. New York, NY, USA: ACM, 2008, pp. 160–173.
- [167] N. Eagle and A. Pentland, “Social serendipity: Mobilizing social software,” *IEEE Pervasive Computing*, vol. 4, no. 2, pp. 28–34, Jan. 2005.
- [168] F. Naya, H. Noma, R. Ohmura, and K. Kogure, “Bluetooth-based indoor proximity sensing for nursing context awareness,” in *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, Oct. 2005, pp. 212–213.
- [169] M. Gast, *Building Applications with iBeacon: Proximity and Location Services with Bluetooth Low Energy*, first edition ed. Beijing: O'Reilly, 2014, oCLC: ocn880566739.
- [170] S. Liu, Y. Jiang, and A. Striegel, “Face-to-Face Proximity Estimation Using Bluetooth On Smartphones,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 4, pp. 811–823, Apr. 2014.
- [171] C. Cattuto, W. V. den Broeck, A. Barrat, V. Colizza, J.-F. Pinton, and A. Vespignani, “Dynamics of Person-to-Person Interactions from Distributed RFID Sensor Networks,” *PLOS ONE*, vol. 5, no. 7, p. e11596, Jul. 2010.
- [172] G. Zhong, I. Goldberg, and U. Hengartner, “Louis, Lester and Pierre: Three Protocols for Location Privacy,” in *Privacy Enhancing Technologies*. Springer, Berlin, Heidelberg, Jun. 2007, pp. 62–76.
- [173] L. Siknys, J. R. Thomsen, S. Saltenis, and M. L. Yiu, “Private and Flexible Proximity Detection in Mobile Social Networks,” in *2010 Eleventh International Conference on Mobile Data Management*. Kansas City, MO, USA: IEEE, 2010, pp. 75–84.
- [174] S. Mascetti, D. Freni, C. Bettini, X. S. Wang, and S. Jajodia, “Privacy in geo-social networks: Proximity notification with untrusted service providers and curious buddies,” *The VLDB Journal*, vol. 20, no. 4, pp. 541–566, Aug. 2011.
- [175] M. Conti and C. Lal, “A Survey on Context-based Co-presence Detection Techniques,” Jul. 2018, <https://arxiv.org/abs/1808.03320v2>.

- [176] A. Chiesa and E. Tromer, “Proof-Carrying Data and Hearsay Arguments from Signature Cards,” in *Proceedings of the 1st Symposium on Innovations in Computer Science*, 2010, pp. 310–331.
- [177] H. Chabanne, R. Hugel, and J. Keuffer, “Verifiable Document Redacting,” in *Computer Security – ESORICS 2017*, S. N. Foley, D. Gollmann, and E. Sneekenes, Eds. Cham: Springer International Publishing, 2017, vol. 10492, pp. 334–351, [http://link.springer.com/10.1007/978-3-319-66402-6\\_20](http://link.springer.com/10.1007/978-3-319-66402-6_20).
- [178] R. Steinfeld, L. Bull, and Y. Zheng, “Content Extraction Signatures,” in *Information Security and Cryptology – ICISC 2001*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Dec. 2001, pp. 285–304.
- [179] D. Herrmann, “Privacy issues in the Domain Name System and techniques for self-defense,” *it - Information Technology*, vol. 57, no. 6, Jan. 2015.
- [180] T. Nakakura, Y. Sumi, and T. Nishida, “Neary: Conversation Field Detection Based on Similarity of Auditory Situation,” in *Proceedings of the 10th Workshop on Mobile Computing Systems and Applications*, ser. HotMobile ’09. New York, NY, USA: ACM, 2009, pp. 14:1–14:6.
- [181] C. Xu, S. Li, G. Liu, Y. Zhang, E. Miluzzo, Y.-F. Chen, J. Li, and B. Finner, “Crowd++: Unsupervised Speaker Count with Smartphones,” in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp ’13. New York, NY, USA: ACM, 2013, pp. 43–52.
- [182] Y. Dodis and N. Fazio, “Public Key Broadcast Encryption for Stateless Receivers,” in *Digital Rights Management*, ser. Lecture Notes in Computer Science, J. Feigenbaum, Ed. Springer Berlin Heidelberg, Nov. 2002, no. 2696, pp. 61–80, [http://link.springer.com/chapter/10.1007/978-3-540-44993-5\\_5](http://link.springer.com/chapter/10.1007/978-3-540-44993-5_5).
- [183] M. Georgiev and V. Shmatikov, “Gone in Six Characters: Short URLs Considered Harmful for Cloud Services,” *arXiv:1604.02734 [cs]*, Apr. 2016, <http://arxiv.org/abs/1604.02734>.
- [184] Li, N. Vishwamitra, B. Knijnenburg, H. Hu, and Y. P. Kelly Caine, “Blur vs. Block: Investigating the Effectiveness of Privacy-Enhancing Obfuscation for Images,” in *Proceedings of the IEEE Conference on*

- Computer Vision and Pattern Recognition Workshops*, 2017, pp. 39–47, [http://openaccess.thecvf.com/content\\_cvpr\\_2017\\_workshops/w16/html/Caine\\_Blur\\_vs.html](http://openaccess.thecvf.com/content_cvpr_2017_workshops/w16/html/Caine_Blur_vs.html).
- [185] D. K. Rappe, “Homomorphic Cryptosystems and their Applications,” Tech. Rep. 001, 2006, <https://eprint.iacr.org/2006/001>.
- [186] Y. Zhang, L. Zhuo, Y. Peng, and J. Zhang, “A secure image retrieval method based on homomorphic encryption for cloud computing,” in *Digital Signal Processing (DSP), 2014 19th International Conference On*. IEEE, 2014, pp. 269–274, <http://ieeexplore.ieee.org/abstract/document/6900669/>.
- [187] N. Yukun, T. Xiaobin, C. Shi, W. Haifeng, Y. Kai, and B. Zhiyong, “A security privacy protection scheme for data collection of smart meters based on homomorphic encryption,” in *EUROCON, 2013 IEEE*. IEEE, 2013, pp. 1401–1405, <http://ieeexplore.ieee.org/abstract/document/6625161/>.
- [188] N. Islam, W. Puech, and R. Brouzet, “A homomorphic method for sharing secret images,” in *International Workshop on Digital Watermarking*. Springer, 2009, pp. 121–135, [http://link.springer.com/chapter/10.1007/978-3-642-03688-0\\_13](http://link.springer.com/chapter/10.1007/978-3-642-03688-0_13).
- [189] R. L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-key Cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.
- [190] T. Elgamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.
- [191] P. Paillier *et al.*, “Public-key cryptosystems based on composite degree residuosity classes,” in *Eurocrypt*, vol. 99. Springer, 1999, pp. 223–238, <http://link.springer.com/content/pdf/10.1007/3-540-48910-X.pdf#page=235>.
- [192] C. Gentry, *A Fully Homomorphic Encryption Scheme*. Stanford University, 2009, <http://search.proquest.com/openview/93369e65682e50979432340f1fdae44e/1?pq-origsite=gscholar&cbl=18750&diss=y>.
- [193] C. Gentry and S. Halevi, “Implementing Gentry’s Fully-Homomorphic Encryption Scheme.” in *EUROCRYPT*, vol. 6632. Springer,

- 2011, pp. 129–148, <http://link.springer.com/content/pdf/10.1007/978-3-642-20465-4.pdf#page=142>.
- [194] P. Yang, X. Gui, J. An, and F. Tian, “An Efficient Secret Key Homomorphic Encryption Used in Image Processing Service,” <https://www.hindawi.com/journals/scn/2017/7695751/>, 2017.
- [195] C. Bettini, S. Jajodia, X. S. Wang, and D. Wijesekera, “Provisions and Obligations in Policy Management and Security Applications,” in *Proceedings of the 28th International Conference on Very Large Data Bases*, ser. VLDB '02. Hong Kong, China: VLDB Endowment, 2002, pp. 502–513, <http://dl.acm.org/citation.cfm?id=1287369.1287413>.
- [196] A. Lazouski, F. Martinelli, and P. Mori, “Usage control in computer security: A survey,” *Computer Science Review*, vol. 4, no. 2, pp. 81–99, May 2010.
- [197] P. Oechslin, “Making a Faster Cryptanalytic Time-Memory Trade-Off,” in *Advances in Cryptology - CRYPTO 2003*, G. Goos, J. Hartmanis, J. van Leeuwen, and D. Boneh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, vol. 2729, pp. 617–630, [http://link.springer.com/10.1007/978-3-540-45146-4\\_36](http://link.springer.com/10.1007/978-3-540-45146-4_36).
- [198] J. J. Treurniet, C. Sarkar, R. V. Prasad, and W. de Boer, “Energy Consumption and Latency in BLE Devices under Mutual Interference: An Experimental Study,” in *2015 3rd International Conference on Future Internet of Things and Cloud*. Rome, Italy: IEEE, Aug. 2015, pp. 333–340.
- [199] Gough, Lui, “Review, Teardown: Keweisi KWS-V20 USB Tester,” <http://goughlui.com/2016/08/20/review-teardown-keweisi-kws-v20-usb-tester/>, Aug. 2016.
- [200] M. Siekkinen, M. Hienkari, J. K. Nurminen, and J. Nieminen, “How low energy is bluetooth low energy? Comparative measurements with Zig-Bee/802.15.4,” in *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. Paris, France: IEEE, Apr. 2012, pp. 232–237.
- [201] M. S. Ferdous, S. Chowdhury, and J. M. Jose, “Privacy threat model in lifelogging.” ACM Press, 2016, pp. 576–581.

- [202] B. A. Price, A. Stuart, G. Calikli, C. McCormick, V. Mehta, L. Hutton, A. K. Bandara, M. Levine, and B. Nuseibeh, "Logging you, Logging me: A Replicable Study of Privacy and Sharing Behaviour in Groups of Visual Lifeloggers," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–18, Jun. 2017.
- [203] M. Fan, A. T. Adams, and K. N. Truong, "Public Restroom Detection on Mobile Phone via Active Probing," in *Proceedings of the 2014 ACM International Symposium on Wearable Computers*, ser. ISWC '14. New York, NY, USA: ACM, 2014, pp. 27–34.
- [204] S. Moncrieff, S. Venkatesh, and G. West, "Dynamic Privacy Assessment in a Smart House Environment Using Multimodal Sensing," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 5, no. 2, pp. 10:1–10:29, Nov. 2008.
- [205] G. W. Fitzmaurice, H. Ishii, and W. A. S. Buxton, "Bricks: Laying the Foundations for Graspable User Interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '95. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 442–449.
- [206] H. Ishii and B. Ullmer, "Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '97. New York, NY, USA: ACM, 1997, pp. 234–241.
- [207] K. P. Fishkin, A. Gujar, B. L. Harrison, T. P. Moran, and R. Want, "Embodied User Interfaces for Really Direct Manipulation," *Commun. ACM*, vol. 43, no. 9, pp. 74–80, Sep. 2000.
- [208] J. Rekimoto and E. Sciammarella, "ToolStone: Effective Use of the Physical Manipulation Vocabularies of Input Devices," in *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '00. New York, NY, USA: ACM, 2000, pp. 109–117.
- [209] J. Sheridan, B. W. Short, K. Van Laerhoven, N. Villar, and G. Kortuem, "Exploring cube affordance: Towards a classification of non-verbal dynamics of physical interfaces for wearable computing," vol. 2003. IEE, 2003, pp. 113–118.

- [210] K. Van Laerhoven, N. Villar, A. Schmidt, G. Kortuem, and H. Gellersen, "Using an Autonomous Cube for Basic Navigation and Input," in *Proceedings of the 5th International Conference on Multimodal Interfaces*, ser. ICMI '03. New York, NY, USA: ACM, 2003, pp. 203–210.
- [211] E. van den Hoven, J. Frens, D. Aliakseyeu, J.-B. Martens, K. Overbeeke, and P. Peters, "Design Research & Tangible Interaction," in *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, ser. TEI '07. New York, NY, USA: ACM, 2007, pp. 109–115.
- [212] D. A. Norman, *The Design of Everyday Things*, revised and expanded edition ed. New York, New York: Basic Books, 2013.
- [213] Y. Rogers, H. Sharp, and J. Preece, *Interaction Design: Beyond Human - Computer Interaction*. John Wiley & Sons, Jun. 2011.
- [214] A. Kapadia, T. Henderson, J. J. Fielding, and D. Kotz, "Virtual Walls: Protecting Digital Privacy in Pervasive Environments," in *Pervasive Computing*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, May 2007, pp. 162–179.
- [215] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, "Smart-its friends: A technique for users to easily establish connections between smart artefacts," in *Ubicomp 2001: Ubiquitous Computing*. Springer, 2001, pp. 116–122.
- [216] S. Ahern, D. Eckles, N. S. Good, S. King, M. Naaman, and R. Nair, "Overexposed?: Privacy Patterns and Considerations in Online and Mobile Photo Sharing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '07. New York, NY, USA: ACM, 2007, pp. 357–366.
- [217] D. Christin, P. S. López, A. Reinhardt, M. Hollick, and M. Kauer, "Privacy Bubbles: User-Centered Privacy Control for Mobile Content Sharing Applications," in *Information Security Theory and Practice. Security, Privacy and Trust in Computing Systems and Ambient Intelligent Ecosystems*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Jun. 2012, pp. 71–86.
- [218] L. A. Goodman, "Snowball Sampling," *The Annals of Mathematical Statistics*, vol. 32, no. 1, pp. 148–170, 1961, <https://www.jstor.org/stable/2237615>.



- [219] M. B. Miles and A. M. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook*. SAGE, Jan. 1994.
- [220] M. E. Sobel, "Asymptotic Confidence Intervals for Indirect Effects in Structural Equation Models," *Sociological Methodology*, vol. 13, pp. 290–312, 1982.
- [221] P. W. Jordan, B. Thomas, I. L. McClelland, and B. Weerdmeester, *Usability Evaluation In Industry*. CRC Press, Jun. 1996.
- [222] S. Borsci, S. Federici, and M. Lauriola, "On the dimensionality of the System Usability Scale: A test of alternative measurement models," *Cognitive Processing*, vol. 10, no. 3, pp. 193–197, Aug. 2009.
- [223] J. R. Lewis and J. Sauro, "The Factor Structure of the System Usability Scale," in *Human Centered Design*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Jul. 2009, pp. 94–103.
- [224] D. A. Norman, *Emotional Design: Why We Love (or Hate) Everyday Things*. Hachette UK, Mar. 2007.
- [225] D. Grijincu, M. A. Nacenta, and P. O. Kristensson, "User-defined Interface Gestures: Dataset and Analysis," in *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '14. New York, NY, USA: ACM, 2014, pp. 25–34.
- [226] A. C. Long, Jr., J. A. Landay, and L. A. Rowe, "Implications for a Gesture Design Tool," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '99. New York, NY, USA: ACM, 1999, pp. 40–47.
- [227] M. A. Nacenta, Y. Kamber, Y. Qiang, and P. O. Kristensson, "Memorability of Pre-designed and User-defined Gesture Sets," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '13. New York, NY, USA: ACM, 2013, pp. 1099–1108.
- [228] T.-K. Yu, L.-C. Lu, and T.-F. Liu, "Exploring factors that influence knowledge sharing behavior via weblogs," *Computers in Human Behavior*, vol. 26, no. 1, pp. 32–41, Jan. 2010.
- [229] D. A. Norman, "Affordance, Conventions, and Design," *interactions*, vol. 6, no. 3, pp. 38–43, May 1999.

- [230] G. Dhillon, T. Oliveira, S. Susarapu, and M. Caldeira, “When Convenience Trumps Security: Defining Objectives for Security and Usability of Systems,” in *Information Security and Privacy Research*, ser. IFIP Advances in Information and Communication Technology. Springer, Berlin, Heidelberg, Jun. 2012, pp. 352–363.
- [231] L. Di Geronimo, M. Bertarini, J. Badertscher, M. Husmann, and M. C. Norrie, “Exploiting Mid-air Gestures to Share Data Among Devices,” in *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI ’17. New York, NY, USA: ACM, 2017, pp. 35:1–35:11.
- [232] A. Liptak, “Amazon’s Alexa started ordering people dollhouses after hearing its name on TV,” <https://www.theverge.com/2017/1/7/14200210/amazon-alexa-tech-news-anchor-order-dollhouse>, Jan. 2017.
- [233] S. Wolfson, “Amazon’s Alexa recorded private conversation and sent it to random contact,” *The Guardian*, May 2018, <https://www.theguardian.com/technology/2018/may/24/amazon-alexa-recorded-conversation>.
- [234] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan, “uWave: Accelerometer-based Personalized Gesture Recognition and Its Applications,” p. 9.
- [235] S. N. Patel, J. S. Pierce, and G. D. Abowd, “A Gesture-based Authentication Scheme for Untrusted Public Terminals,” in *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST ’04. New York, NY, USA: ACM, 2004, pp. 157–160.
- [236] “Physical assault by McDonald’s for wearing Digital Eye Glass,” <http://eyetap.blogspot.com/2012/07/physical-assault-by-mcdonalds-for.html>.
- [237] Unknown, “I Love Social Media: Google Glass Assault and Robbery at Molotov’s Bar, Haight St. February 22, 2014 (warning: Profanity),” <http://ilovesocialmediainc.blogspot.com/2014/03/google-glass-assault-and-robbery-at.html>, Mar. 2014.
- [238] J. Shu, R. Zheng, and P. Hui, “Your Privacy Is in Your Hand: Interactive Visual Privacy Control with Tags and Gestures,” in *Communication Systems and Networks*, ser. Lecture Notes in Computer Science, N. Sastry and S. Chakraborty, Eds. Springer International Publishing, 2017, pp. 24–43.

- [239] M. Koelle, S. Ananthanarayan, S. Czupalla, W. Heuten, and S. Boll, “Your Smart Glasses’ Camera Bothers Me!: Exploring Opt-in and Opt-out Gestures for Privacy Mediation,” in *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*, ser. NordiCHI ’18. New York, NY, USA: ACM, 2018, pp. 473–481.
- [240] F. Paci, A. Squicciarini, and N. Zannone, “Survey on Access Control for Community-Centered Collaborative Systems,” *ACM Comput. Surv.*, vol. 51, no. 1, pp. 6:1–6:38, Jan. 2018.
- [241] W. Tolone, G.-J. Ahn, T. Pai, and S.-P. Hong, “Access Control in Collaborative Systems,” *ACM Comput. Surv.*, vol. 37, no. 1, pp. 29–41, Mar. 2005.
- [242] A. K. Malik, A. Anjum, B. Raza, and M. Gupta, Eds., *Innovative Solutions for Access Control Management*, ser. Advances in Information Security, Privacy, and Ethics. IGI Global, 2016, <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-5225-0448-1>.
- [243] M. S. Ferdous, S. Chowdhury, and J. M. Jose, “Privacy Threat Model in Lifelogging,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, ser. UbiComp ’16. New York, NY, USA: ACM, 2016, pp. 576–581.
- [244] C. Efstratiou, I. Leontiadis, M. Picone, K. K. Rachuri, C. Mascolo, and J. Crowcroft, “Sense and Sensibility in a Pervasive World,” in *Pervasive Computing*, ser. Lecture Notes in Computer Science, J. Kay, P. Lukowicz, H. Tokuda, P. Olivier, and A. Krüger, Eds. Springer Berlin Heidelberg, 2012, pp. 406–424.
- [245] R. Rawassizadeh and A. M. Tjoa, “Securing Shareable Life-logs.” IEEE, Aug. 2010, pp. 1105–1110.
- [246] J. Y. Tsai, P. G. Kelley, L. F. Cranor, and N. Sadeh, “Location-Sharing Technologies: Privacy Risks and Controls,” p. 26.
- [247] V. Sacramento, M. Endler, and F. N. Nascimento, “A Privacy Service for Context-aware Mobile Computing,” in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM’05)*, Sep. 2005, pp. 182–193.
- [248] Y. G. Le Gall, A. J. Lee, and A. Kapadia, “PlexC: A policy language for exposure control,” in *Proceedings of the 17th ACM Symposium on Access*

- Control Models and Technologies - SACMAT '12.* Newark, New Jersey, USA: ACM Press, 2012, p. 219.
- [249] R. Hull, B. Kumar, D. Lieuwen, P. F. Patel-Schneider, A. Sahuguet, S. Varadarajan, and A. Vyas, “Enabling context-aware and privacy-conscious user data sharing,” in *IEEE International Conference on Mobile Data Management, 2004. Proceedings. 2004*, Jan. 2004, pp. 187–198.
- [250] L. Kagal, T. Finin, and A. Joshi, “A policy language for a pervasive computing environment,” in *Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop On.* IEEE, 2003, pp. 63–74, [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1206958](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1206958).
- [251] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, “The Ponder Policy Specification Language,” in *Policies for Distributed Systems and Networks*, ser. Lecture Notes in Computer Science, M. Sloman, E. C. Lupu, and J. Lobo, Eds. Springer Berlin Heidelberg, 2001, pp. 18–38.
- [252] A. Behrooz and A. Devlic, “A Context-Aware Privacy Policy Language for Controlling Access to Context Information of Mobile Users,” in *Security and Privacy in Mobile Information and Communication Systems*, R. Prasad, K. Farkas, A. U. Schmidt, A. Liroy, G. Russello, and F. L. Luccio, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 94, pp. 25–39, [http://link.springer.com/10.1007/978-3-642-30244-2\\_3](http://link.springer.com/10.1007/978-3-642-30244-2_3).
- [253] A. Corradi, R. Montanari, and D. Tibaldi, “Context-based access control for ubiquitous service provisioning,” in *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.*, Sep. 2004, pp. 444–451 vol.1.
- [254] J. Ahn, B.-M. Chang, and K.-G. Doh, “A Policy Description Language for Context-Based Access Control and Adaptation in Ubiquitous Environment,” in *Emerging Directions in Embedded and Ubiquitous Computing*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, X. Zhou, O. Sokolsky, L. Yan, E.-S. Jung, Z. Shao, Y. Mu, D. C. Lee, D. Y. Kim, Y.-S. Jeong, and C.-Z. Xu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 4097, pp. 650–659, [http://link.springer.com/10.1007/11807964\\_66](http://link.springer.com/10.1007/11807964_66).

- [255] H. Choi, S. Chakraborty, Z. Charbiwala, and M. Srivastava, "Sensorsafe: A framework for privacy-preserving management of personal sensory information," *Secure Data Management*, pp. 85–100, 2011, <http://link.springer.com/content/pdf/10.1007/978-3-642-23556-6.pdf#page=93>.
- [256] M. J. Covington, W. Long, S. Srinivasan, A. K. Dev, M. Ahamad, and G. D. Abowd, "Securing Context-aware Applications Using Environment Roles," in *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies*, ser. SACMAT '01. New York, NY, USA: ACM, 2001, pp. 10–20.
- [257] J. I. Hong and J. A. Landay, "An Architecture for Privacy-sensitive Ubiquitous Computing," in *Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '04. New York, NY, USA: ACM, 2004, pp. 177–189.
- [258] M. J. Covington, P. Fogla, and a. M. Ahamad, "A context-aware security architecture for emerging applications," in *18th Annual Computer Security Applications Conference, 2002. Proceedings.*, Dec. 2002, pp. 249–258.
- [259] A. K. Dey, G. D. Abowd, and D. Salber, "A Context-Based Infrastructure for Smart Environments," in *Managing Interactions in Smart Environments*, P. Nixon, G. Lacey, and S. Dobson, Eds. London: Springer London, 2000, pp. 114–128, [http://link.springer.com/10.1007/978-1-4471-0743-9\\_11](http://link.springer.com/10.1007/978-1-4471-0743-9_11).
- [260] J. Wiese, P. G. Kelley, L. F. Cranor, L. Dabbish, J. I. Hong, and J. Zimmerman, "Are You Close with Me? Are You Nearby?: Investigating Social Groups, Closeness, and Willingness to Share," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, ser. UbiComp '11. New York, NY, USA: ACM, 2011, pp. 197–206.
- [261] R. Rawassizadeh, "Towards sharing life-log information with society," *Behaviour & Information Technology*, vol. 31, no. 11, pp. 1057–1067, Nov. 2012.
- [262] S. Consolvo, I. E. Smith, T. Matthews, A. LaMarca, J. Tabert, and P. Powledge, "Location Disclosure to Social Relations: Why, when, & What People Want to Share," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '05. New York, NY, USA: ACM, 2005, pp. 81–90.

- [263] T. Olsson, H. Soronen, and K. Väänänen-Vainio-Mattila, “User Needs and Design Guidelines for Mobile Services for Sharing Digital Life Memories,” in *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '08. New York, NY, USA: ACM, 2008, pp. 273–282.
- [264] D. Carrie and E. Gates, “Access Control Requirements for Web 2.0 Security and Privacy,” in *Proc. of Workshop on Web 2.0 Security & Privacy (W2SP 2007)*, 2007.
- [265] B. Carminati, E. Ferrari, and A. Perego, “Rule-Based Access Control for Social Networks,” in *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, R. Meersman, Z. Tari, and P. Herrero, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 4278, pp. 1734–1744, [http://link.springer.com/10.1007/11915072\\_80](http://link.springer.com/10.1007/11915072_80).
- [266] M.-R. Ra, R. Govindan, and A. Ortega, “P3: Toward Privacy-Preserving Photo Sharing,” p. 14.
- [267] P. Ilia, I. Polakis, E. Athanasopoulos, F. Maggi, and S. Ioannidis, “Face/Off: Preventing Privacy Leakage From Photos in Social Networks,” in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. New York, NY, USA: ACM, 2015, pp. 781–792.
- [268] B. A. Bouna, R. Chbeir, A. Gabillon, and P. Capolsini, “A Flexible Image-Based Access Control Model for Social Networks,” in *Security and Privacy Preserving in Social Networks*, R. Chbeir and B. Al Bouna, Eds. Vienna: Springer Vienna, 2013, pp. 337–364, [http://link.springer.com/10.1007/978-3-7091-0894-9\\_11](http://link.springer.com/10.1007/978-3-7091-0894-9_11).
- [269] P. Ilia, B. Carminati, E. Ferrari, P. Fragopoulou, and S. Ioannidis, “SAMPAC: Socially-Aware Collaborative Multi-Party Access Control,” in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, ser. CODASPY '17. New York, NY, USA: ACM, 2017, pp. 71–82.
- [270] H. Hu, G. Ahn, and J. Jorgensen, “Multiparty Access Control for Online Social Networks: Model and Mechanisms,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1614–1627, Jul. 2013.

- [271] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino, “2D and 3D face recognition: A survey,” *Pattern Recognition Letters*, vol. 28, no. 14, pp. 1885–1906, Oct. 2007.
- [272] J. Steil, M. Koelle, W. Heuten, S. Boll, and A. Bulling, “PrivacEye: Privacy-Preserving First-Person Vision Using Image Features and Eye Movement Analysis,” *arXiv:1801.04457 [cs]*, Jan. 2018, <http://arxiv.org/abs/1801.04457>.
- [273] S. Alcalde Bagüés, A. Zeidler, C. Fernandez Valdivielso, and I. R. Matias, “Towards Personal Privacy Control,” in *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*, ser. Lecture Notes in Computer Science, R. Meersman, Z. Tari, and P. Herrero, Eds. Springer Berlin Heidelberg, 2007, pp. 886–895.
- [274] S. A. Bagüés, A. Zeidler, F. Valdivielso, and I. R. Matias, “Sentry@Home - Leveraging the Smart Home for Privacy in Pervasive Computing,” *International Journal of Smart Home*, vol. 1, no. 2, p. 18, 2007.
- [275] “PRML: Privacy Rights Markup Language Specification, Zero Knowledge Systems,” 2001.
- [276] J. Iyilade and J. Vassileva, “P2U: A Privacy Policy Specification Language for Secondary Data Sharing and Usage,” in *2014 IEEE Security and Privacy Workshops*, May 2014, pp. 18–22.
- [277] J. Lobo, R. Bhatia, and S. Naqvi, “A Policy Description Language,” p. 8.
- [278] M. Blount, J. Davis, M. Ebling, W. Jerome, B. Leiba, X. Liu, and A. Misra, “Privacy Engine for Context-Aware Enterprise Application Services,” in *2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, vol. 2, Dec. 2008, pp. 94–100.
- [279] A. Kapadia, D. Kotz, and N. Triandopoulos, “Opportunistic sensing: Security challenges for the new paradigm,” in *2009 First International Communication Systems and Networks and Workshops*. Bangalore, India: IEEE, Jan. 2009, pp. 1–10.
- [280] A. C. Yao, “Protocols for Secure Computations,” p. 5.
- [281] B. Y. Lim, “Improving Trust in Context-aware Applications with Intelligibility,” in *Proceedings of the 12th ACM International Conference Adjunct Papers*

- on Ubiquitous Computing - Adjunct*, ser. UbiComp '10 Adjunct. New York, NY, USA: ACM, 2010, pp. 477–480.
- [282] J. Vermeulen, G. Vanderhulst, K. Luyten, and K. Coninx, “PervasiveCrystal: Asking and Answering Why and Why Not Questions about Pervasive Computing Applications,” in *2010 Sixth International Conference on Intelligent Environments*, Jul. 2010, pp. 271–276.
- [283] R. F. Kizilcec, “How Much Information?: Effects of Transparency on Trust in an Algorithmic Interface,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: ACM, 2016, pp. 2390–2395.
- [284] B. Y. Lim, A. K. Dey, and D. Avrahami, “Why and why not explanations improve the intelligibility of context-aware intelligent systems,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 2119–2128, <http://dl.acm.org/citation.cfm?id=1519023>.
- [285] B. Y. Lim and A. K. Dey, “Investigating Intelligibility for Uncertain Context-aware Applications,” in *Proceedings of the 13th International Conference on Ubiquitous Computing*, ser. UbiComp '11. New York, NY, USA: ACM, 2011, pp. 415–424.
- [286] L. Franceschi-Bicchierai, “Hackers Could Break Into Your Monitor To Spy on You and Manipulate Your Pixels,” [https://www.vice.com/en\\_us/article/jpgdzb/hackers-could-break-into-your-monitor-to-spy-on-you-and-manipulate-your-pixels](https://www.vice.com/en_us/article/jpgdzb/hackers-could-break-into-your-monitor-to-spy-on-you-and-manipulate-your-pixels), Aug. 2016.
- [287] “1 billion computer monitors vulnerable to undetectable firmware attacks,” <https://boingboing.net/2016/08/06/computer-monitors-vulnerable-t.html>.
- [288] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog Computing and Its Role in the Internet of Things,” p. 3.
- [289] S. Yi, C. Li, and Q. Li, “A Survey of Fog Computing: Concepts, Applications and Issues,” in *Proceedings of the 2015 Workshop on Mobile Big Data - Mobidata '15*. Hangzhou, China: ACM Press, 2015, pp. 37–42.



- 
- [290] N. Damianou, A. K. Bandara, M. Sloman, and E. C. Lupu, "A Survey of Policy Specification Approaches," p. 37.
- [291] S. Lederer, A. K. Dey, and J. Mankoff, "A Conceptual Model and a Metaphor of Everyday Privacy in Ubiquitous Computing Environments," p. 9.

