

Towards Privacy-Compliant Mobile Computing

A dissertation submitted towards the degree
Doctor of Engineering
of the Faculty of Mathematics and Computer Science of
Saarland University

by

Paarijaat Aditya

Saarbrücken, 2018

Date of the colloquium: 03/12/2018

Dean: Univ.-Prof. Dr. Sebastian Hack

Reporters: Prof. Dr. Peter Druschel

Dr. Deepak Garg

Mary G. Baker, Ph.D.

Chairman of the Examination board: Prof. Dr. Holger Hermanns

Scientific Assistant: Dr. Amaury Pouly

Dedicated to my parents,
Archana Aditya
and
Vinay Aditya

Abstract

Sophisticated mobile computing, sensing and recording devices like smartphones, smartwatches, and wearable cameras are carried by their users virtually around the clock, blurring the distinction between the online and offline worlds. While these devices enable transformative new applications and services, they also introduce entirely new threats to users' privacy because they can capture a complete record of the user's location, online and offline activities, and social encounters, including an audiovisual record. Such a record of users' personal information is highly sensitive and is subject to numerous privacy risks. In this thesis, we have investigated and built systems to mitigate two such privacy risks: 1) privacy risks due to ubiquitous digital capture, where bystanders may inadvertently be captured in photos and videos recorded by other nearby users, 2) privacy risks to users' personal information introduced by a popular class of apps called 'mobile social apps'. In this thesis, we present two systems, called I-Pic and EnCore, built to mitigate these two privacy risks.

Both systems aim to put the users back in control of what personal information is being collected and shared, while still enabling innovative new applications. We built working prototypes of both systems and evaluated them through actual user deployments. Overall we demonstrate that it is possible to achieve privacy-compliant digital capture and it is possible to build privacy-compliant mobile social apps, while preserving their intended functionality and ease-of-use. Furthermore, we also explore how the two solutions can be merged into a powerful combination, one which could enable novel workflows for specifying privacy preferences in image capture that do not currently exist.

Zusammenfassung

Die heutigen Geräte zur mobilen Kommunikation, und Messdatenerfassung und -aufzeichnung, wie Smartphones, Smartwatches und Sport-Kameras werden in der Regel von ihren Besitzern rund um die Uhr getragen, so daß der Unterschied zwischen Online- und Offline-Zeiten zunehmend verschimmt. Diese Geräte ermöglichen zwar völlig neue Applikationen und Dienste, gefährden aber gleichzeitig die Privatsphäre ihrer Nutzer, weil sie den Standort, die gesamten On-und Offline Aktivitäten, sowie die soziale Beziehungen protokollieren, bis hin zu audio-visuellen Aufzeichnungen. Solche persönlichen Nutzerdaten sind extrem schützenswert und sind verschiedenen Risiken in Bezug auf die Privatsphäre ausgesetzt. In dieser These haben wir Systeme untersucht und gebaut, die zwei dieser Risiken für die Privatsphäre minimieren: 1) Risiko der Privatsphäre wegen omnipräsenter digitaler Aufzeichnungen Dritter, bei denen Unbeteiligte unbeabsichtigt (oder gegen ihren Wunsch) in Fotos und Videos festgehalten werden 2) Risiko für die persönlichen Informationen der Nutzer welche durch die bekannte Kategorie der sozialen Applikationen herbeigeführt werden. In dieser These stellen wir zwei Systeme, namens I-Pic und EnCore vor, welche die zwei Privatsphäre-Risiken minimieren.

Beide System wollen dem Benutzer die Kontrolle zurückgeben, zu entscheiden welche seiner persönlichen Daten gesammelt und geteilt werden, während weiterhin neue innovative Applikationen ermöglicht werden. Wir haben für beide Systeme funktionsfähige Prototypen gebaut und diese mit echten Nutzerdaten evaluiert. Wir können generell zeigen dass es möglich ist, digitale Aufzeichnung zu machen, und soziale Applikationen zu bauen, welche nicht die Privatsphäre verletzen, ohne dabei die beabsichtige Funktionalität zu verlieren oder die Bedienbarkeit zu mindern. Des weiteren erforschen wir, wie diese zwei Systeme zu einem leistungsfähigeren Ansatz zusammengeführt werden können, welcher neuartige Workflows ermöglicht, um Einstellungen zur Privatsphäre für digitale Aufzeichnungen vorzunehmen, die es heute noch nicht gibt.

Acknowledgments

It is hard for me to put myself in my parents' shoes who, over the years, provided me a loving and nurturing environment to grow in, who encouraged me to push my own boundaries, and then in an instant gave me their blessings when I decided to go away for many years on a journey of self discovery. This thesis is dedicated to my parents who gave me the wings to fly high in search of my own horizons. I hope to return back to my roots one day.

I have always looked up to my elder Brother, Animesh, who was my inspiration to pursue the doctorate degree. He has always been there for me whenever I needed his brotherly and professional guidance, and he continues to look out for me, behind scenes, even if I get lost in my own struggles. I am forever grateful for his presence and his encouragement on every step of the way.

My friends in Saarbruecken made the small city truly a memorable place to be. They not only helped me develop the softer skills of my personality but also made me appreciate the importance of living in the moment. A special mention to Mayank and Ines who went out of their way to support me in my journey and who have for me become the very definition of the word 'friends'.

To my colleagues, faculty, IT, and the administrative staff members of MPI-SWS who helped me grow as a researcher through their constructive criticism, encouragement, and their help in navigating many obstacles of my technical quests. A special mention to Rijurekha and Viktor, who I thoroughly enjoyed working with on both the projects. I am also grateful to my thesis reviewers, Deepak and Mary, for their time and their valuable feedback on my thesis.

I cannot begin to thank my advisor, Prof. Peter Druschel, for giving me a chance to pursue the doctorate under his guidance. I am extremely grateful for his unwavering guidance and his tremendous patience during my long journey. His passion for research has truly been an inspiration for me.

And finally, last but by no means least, my wife, Nandita, whose unconditional support, love, and affection has been the bedrock of my mental and physical well being. Not only has she been putting up with patience the idiosyncrasies of a Ph.D. student, she has become the source of my daily inspiration through her kind and humble nature. I am filled with gratitude towards her for literally flying into my life and starting with me the next phase of our life journey together.

Contents

1	Introduction	8
1.1	Contributions	10
2	Background	13
2.1	Garbled Circuits & Oblivious Transfers	13
2.1.1	Oblivious Transfer (OT)	13
2.1.2	Yao’s Garbled Circuits Protocol	14
2.1.3	Reducing the number of OTs	19
2.2	Head detection in I-Pic	20
2.2.1	Related work: advances in object detection in images	21
2.2.2	Head detector description	23
3	I-Pic: A Platform for Privacy-Compliant Image Capture	25
3.1	Introduction	25
3.2	I-Pic Related work	27
3.3	Online Survey	28
3.4	I-Pic Architecture	32
3.4.1	I-Pic overview	32
3.4.2	Threat model	34
3.5	I-Pic Design	34
3.5.1	Image processing	36
3.5.2	Cryptographic Protocols	37
3.5.3	Secure matching protocol	39
3.6	I-Pic Evaluation	42
3.6.1	Deployments	43
3.6.2	I-Pic decision tree	43
3.6.3	I-Pic overall performance	47
3.6.4	Vision pipeline analysis	49
3.6.5	Secure Feature Comparison	55
3.6.6	Runtime and Energy Consumption	56

3.7	I-Pic Summary	59
4	Exploring I-Pic’s performance limits	60
4.1	Exploring head detection	60
4.1.1	Post-processing head detector output	61
4.1.2	Performance of the head detector on the I-Pic dataset	61
4.2	Exploring I-Pic’s runtime performance on a new Mobile SoC	69
4.3	Conclusion	70
5	EnCore: Private, Context-based Communication for Mobile Social Apps	72
5.1	Introduction	72
5.2	EnCore Related Work	74
5.3	EnCore: Capabilities and Requirements	76
5.3.1	Detecting nearby users and resources	76
5.3.2	Event-based communication/sharing	77
5.4	EnCore Design	78
5.4.1	EnCore security properties	79
5.4.2	Encounters	80
5.4.3	Events	81
5.4.4	Communication	82
5.4.5	Security guarantees	83
5.5	Using Events with Context	83
5.5.1	Browsing the timeline	84
5.5.2	Creating events	86
5.5.3	Posting information	86
5.5.4	Receiving information	87
5.5.5	Controlling linkability	87
5.5.6	Implementation of conduits and router	87
5.6	Evaluation	88
5.7	Discussion	92
5.7.1	Qualitative user feedback	92
5.7.2	Risks and challenges	93
5.8	EnCore Summary	94
6	Discussion and Future work	96
6.1	Leveraging EnCore for I-Pic	97
6.1.1	New workflows enabled by integrating EnCore with I-Pic	98
6.1.2	Privacy concerns	100
6.2	Policy enforcement on the viewer	101

6.3	Extending I-Pic to Video and Audio	101
6.4	Using a trusted photographer’s agent	103
6.5	Using ARM TrustZone to extend I-Pic’s threat model	103
6.6	Future work	104
6.6.1	Obscuring mechanisms	104
6.6.2	Requests to override user preferences	106
7	Conclusion	107
	Appendices	109
A	I-Pic User Survey	110

Chapter 1

Introduction

Sophisticated mobile computing, sensing, and recording devices are now commonplace. Smart phones and smart watches have already achieved significant adoption, and novel devices, like Snapchat Spectacles, Microsoft HoloLens, Vuzix Blade, are imminent. These devices are carried by their users virtually round the clock, blurring the distinction between the online & offline world, and enabling transformative new applications. For instance, mobile apps can provide location- and activity- sensitive information, which can be overlaid onto a user's field of view using smart glasses. Mobile devices can also maintain a detailed record of a user's life, recording everything a user does, sees, hears, who he meets, and to enable communication related to a shared experience or an event.

However, these applications and services also introduce a range of new threats to users' privacy. While a user carries a mobile phone, it can capture a complete record of a user's location, online and offline activities, and social encounters, including an audiovisual record. While such a record is very useful to a user for their own reference, it is also highly sensitive and inherently private. Unlike information users post on online social networks, most users would likely not want to share such a comprehensive record with anyone. Such a record of users' personal information is subject to numerous privacy risks [1], and real world instances of misuse of users' personal data by organizations for profit have greatly aggravated these concerns [2].

Even in cases where a user does not use mobile applications or devices herself, her privacy might still be at risk; e.g. when a user is captured in photos taken by nearby strangers that are later shared on online social networks, revealing the whereabouts of everyone photographed. Such unwanted image capture is perceived to be such a serious privacy threat that it led to Google Glass [3] being banned at numerous venues. These privacy concerns were also likely one of the factors that led to Google Glass's discontinuation [4]. This example clearly highlights that for future mobile technology to be broadly accepted, privacy concerns cannot be treated as an afterthought. Rather,

privacy should be a first-order concern, built into the design of mobile apps and hardware.

At first glance it may appear that loss of privacy is inherent in ubiquitous context-sensitive mobile applications. However, in this thesis, we show that it is possible to achieve privacy by design for these mobile apps without losing intended functionality. In particular, we investigate whether it is possible to reconcile privacy with functionality in two specific contexts: 1) Spontaneous image capture using our smartphones while protecting the privacy of bystanders captured in these images, and 2) Building mobile social applications while preserving the privacy of users' personal information these apps operate upon.

1. I-Pic: A Platform for Privacy-Compliant Image Capture: (Chapter 3): The first part of this thesis focuses on the privacy risks introduced by ubiquitous digital capture facilitated by smartphone cameras, smart glasses, and life-logging cameras. Bystanders may be photographed (either intentionally or inadvertently) without their consent, which poses a significant risk to their privacy and security. To mitigate this risk, we built and deployed I-Pic, a trusted software platform that integrates digital capture with user-defined privacy. In I-Pic, users choose a level of privacy (e.g., image capture allowed or not) based on social context (e.g., in public vs. with friends vs. at workplace). The privacy choices of nearby users are advertised via BLE (Bluetooth Low-Energy), and I-Pic-compliant capture platforms generate edited media to conform to the privacy choices of the captured subjects. I-Pic uses a state-of-the-art deep neural network for face recognition, and combines it with secure multi-party computations, to ensure that users' visual features and privacy choices aren't revealed publicly, regardless of whether these users are the subject of an image capture. We evaluate the I-Pic prototype in realistic social scenarios, and demonstrate that the technical impediments for privacy-compliant imaging can be reasonably overcome using current hardware platforms, without giving up the spontaneity, ubiquity, and flexibility of image capture.

2. EnCore: Private, Context-based Communication for Mobile Social Apps: (Chapter 5): The second part of this thesis focuses on mitigating privacy risks introduced by mobile social apps. These apps consider users' location, activity, and nearby devices to provide context-aware services, e.g., sharing captured images with nearby users, detecting the presence of friends in close vicinity, sharing news and gossip with nearby people, and helping people find missed connections. Most of the currently deployed mobile social apps rely on a trusted cloud service to match and relay information, requiring users to reveal their whereabouts (potentially including a continuous trace of their location), the perils of which have been extensively noted [5, 6, 7, 8, 9]. To mitigate this problem, we built and deployed EnCore, a mobile platform that does not require a trusted provider, but instead builds on secure encounters

between pairs of devices as a foundation for privacy-preserving communication for mobile social apps. An encounter occurs whenever two devices discover each other within Bluetooth radio range and generate a unique encounter ID. EnCore groups these encounters in named communication abstractions called *events*, and enables encrypted communication and sharing among event participants, all the while relying on existing network, storage, and online social network services. Furthermore, the encounter formation protocol used by EnCore, i.e. SDDR [10], ensures that users cannot be tracked using their Bluetooth identifiers. We evaluated EnCore via multiple user deployments, and based on the favorable user feedback we received, we demonstrate that EnCore can support a wide range of event-based communication primitives for mobile social apps, with strong security and privacy guarantees.

1.1 Contributions

Both I-Pic and EnCore provide platforms for building mobile apps in a privacy-compliant manner that puts users in control of what personal information is collected, and how it is shared. The primary contribution of this thesis is to demonstrate, through actual deployments of applications built using these platforms, that one can preserve most of the functionality of spontaneous image capture and mobile social applications without giving up privacy.

The specific contributions of the **I-Pic** project (Chapter 3) are:

- We analyze an important privacy challenge, image capture privacy, which has no satisfactory solution till now.
- We demonstrate how one can achieve privacy preserving image capture in a convenient & flexible manner (as compared to previous work), and show that it is even feasible on current mobile devices.
- We report on the results of a user study conducted to understand people’s sentiments and preferences towards privacy in digital capture. The aim of the user study was to understand, to what extent people are willing to respect other people’s privacy, how often do their privacy preferences change, and in which situations. The findings from this study set the design requirements for I-Pic.
- We develop a technical architecture that addresses these design requirements and that provides a more flexible solution than existing related work.
- We identify specific technical components to implement our design, and we individually optimize the implementation of these components to make them power efficient and scalable.

- We implement a working prototype of I-Pic that can run on a resource constrained Android mobile device. We evaluate the I-Pic prototype in realistic social scenarios with wide range of lighting conditions, and where bystanders appear in natural poses both in the foreground and background of an image.
- We also present an exploration of the performance limits of I-Pic by evaluating how I-Pic can benefit from both recent advances in machine learning techniques and powerful new hardware likely to be available in future mobile devices. In particular we show (in Chapter 4) that newer, more accurate object detection techniques could be integrated in I-Pic without compromising the overall energy efficiency of the device.

The specific contributions of the **EnCore** project (Chapter 5) are:

- We present the design of EnCore, a communication platform that provides powerful new capabilities to mobile social apps, with strong security and privacy guarantees, without requiring a trusted provider. We also present an implementation of EnCore on Android devices.
- We demonstrate EnCore’s capabilities through *Context*, an Android application that provides communication, sharing, collaboration, and organization based on events. The application was shaped by user feedback from a series of test bed deployments.
- We report on a series of live deployments of *Context* and EnCore, with 35 users at MPI-SWS.

Finally, we also explore how to merge the capabilities of I-Pic and EnCore into a powerful combination, one which could enable completely new ways of specifying and communicating privacy preferences that do not exist currently. Specifically (in Chapter 6), we envision a system that extends I-Pic with EnCore’s encounter-based communication, and describe how this combination could be used to enable novel ways of communication between photographers and bystanders. For e.g., these workflows provide a convenient way for photographers to reach out to captured bystanders (and for bystanders to reach out to photographers) to possibly seek their consent before publishing a photograph (or to revoke their consent for a photograph). Such functionality could be useful for compliance with governmental privacy regulations, such as, GDPR [11] and AB375 [12].

Overall, we believe that even though a single platform alone may not be able to provide an ideal end-to-end privacy-preserving infrastructure, technical innovations that mitigate specific risk vectors will not only provide a strong basis for a broader societal

conversation about the value of user privacy, but will also be needed for future mobile and wearable technology to be broadly accepted by users.

The rest of the document is structured as follows. In Chapter 2 we describe some of the existing cryptographic and computer vision building blocks that have been used in the implementation of I-Pic. In Chapter 3 we present a detailed description of the I-Pic project and its related work. In Chapter 4 we present an exploration of how I-Pic can benefit from recent advances in both hardware and software. In Chapter 5 we present a detailed description of the EnCore project and its related work. Finally, in Chapter 6, we present a discussion of how EnCore and I-Pic can be used in conjunction and what this combination means for different parties involved. We also describe alternate design points and extensions for I-Pic, and also describe some future work directions for I-Pic.

Chapter 2

Background

Here we describe some existing cryptographic and computer vision building blocks that have been used in the implementation of I-Pic. These are not contributions of this thesis, and have been included here for background only.

2.1 Garbled Circuits & Oblivious Transfers

Secure function evaluation (SFE) refers to the problem of how two parties can collaborate to correctly compute the output of a function without either party needing to reveal their inputs to the function, either to each other or to a third party. In 1986 Andrew Yao presented a solution to the problem called *garbled circuits*.

Yao's *garbled circuits protocol* (GCP) transforms any function into a function that can be evaluated securely by modeling the function as a boolean circuit, and then masking the inputs and outputs of each gate so that the party executing the function cannot discern any information about the inputs or intermediate values to the function. The protocol is secure as long as both parties are semi-honest. A semi-honest adversary is assumed to follow all required steps in a protocol, but will look for all advantageous information leaked from the execution of the protocol, such as intermediate values, control flow decisions, or values derivable from the same. Note that Yao's protocol does not guarantee that one party is not able to learn other party's input by examining the function's result (if the function being executed allows for such reverse engineering).

In the following description we will refer to the function to be evaluated as f , the two parties as $P1$ & $P2$, their inputs as i_{P1} & i_{P2} respectively, and function's output as $u = f(i_{P1}, i_{P2})$.

2.1.1 Oblivious Transfer (OT)

OT refers to methods for two parties to exchange one-out-of-several values, with the sending party blinded to what value was selected, and the receiving party blinded to all other possible values that could have, but were not, selected. While OT and SFE

are approaches to distinct (though related) problems, understanding Yao's GCP and its security properties requires understanding OT. It's a cryptographic primitive and a building block that Yao's GCP builds on. In the following we provided a brief overview of the OT problem and a simple OT protocol that is used as a building in I-Pic.

Problem definition: A general form of OT is *1-out-of-N oblivious transfer*, a two party protocol where $P1$, the sending party, has a collection of values. $P2$ is able to select one of the values from this set to receive, but is not able to learn any of the other values.

More formally, a *1-out-of-N oblivious transfer protocol* takes as inputs a set of N values from $P1$, and an index i from $P2$, where $0 \leq i < |N|$. The protocol then outputs nothing to $P1$, and N_i to $P2$ in a manner that prevents $P2$ from learning N_j for all values of $j \neq i$.

A special case of the above is the *1-out-of-2 oblivious transfer* problem, where N is fixed at 2. Here $P1$ has just two values, and $P2$ is accordingly limited to $i \in \{0, 1\}$.

Example 1-out-of-2 OT Protocol: Figure 2.1 shows a *1-out-of-2 OT* protocol[13] that is used in I-Pic as a building block. The protocol is based on Diffie-Hellman Key Exchange. As long as no party deviates from the protocol, *receiver* is able to recover the desired string M_c but is not able to recover the other value, M_{1-c} . Similarly, *sender* never learns c . The protocol is included here to assist in the next section's explanation of how the full GCP works, and to provide an easy-to-understand example of OT to build from later.

2.1.2 Yao's Garbled Circuits Protocol

This section provides a description of Yao's garbled circuits protocol and how the protocol incorporates OT. For a more comprehensive description please refer to [14].

High-level Description of the Protocol

$P1$ and $P2$ wish to compute function f securely, so that their inputs to the function remain secret. The computation is initiated by first modeling f as a boolean circuit. $P1$ then "garbles" the circuit by representing the boolean values on all wires in the circuit with pseudo-random bit strings, and then keeping the mapping between the boolean values to random bit strings secret. This is done for the input and output wires of every gate in the circuit, with the exception of circuit's output gates; the values of these gates' output wires are left un-garbled.

$P1$ then replaces each bit of his input with the pseudo-random string that maps to that bit's input on the corresponding input wire into the circuit. $P1$ then sends the garbled circuit and his garbled input to $P2$.

$P2$ receives both the garbled circuit and $P1$'s garbled input. However, since all input wires into the circuit have been garbled and only $P1$ has the mapping between the

Diffie-Hellman Key Exchange

Sender

Input: (M)

Output: none

$$a \leftarrow \mathbb{Z}_p$$

$$A = g^a$$

$$B = g^b$$

$$k = H(B^a)$$

$$e \leftarrow \text{Enc}_k(M)$$

Receiver

Input: none

Output: M

$$b \leftarrow \mathbb{Z}_p$$

$$k = H(A^b)$$

$$M = \text{Dec}_k(e)$$

OT Protocol based on Diffie-Hellman

Sender

Input: (M_0, M_1)

Output: none

$$a \leftarrow \mathbb{Z}_p$$

$$A = g^a$$

$$B$$

$$k_0 = H(B^a)$$

$$k_1 = H\left(\left(\frac{B}{A}\right)^a\right)$$

$$e_0 \leftarrow \text{Enc}_{k_0}(M_0)$$

$$e_1 \leftarrow \text{Enc}_{k_1}(M_1)$$

Receiver

Input: c

Output: M_c

$$b \leftarrow \mathbb{Z}_p$$

$$\text{if } c = 0 : B = g^b$$

$$\text{if } c = 1 : B = Ag^b$$

$$k_R = H(A^b)$$

$$M_c = \text{Dec}_{k_R}(e_c)$$

Figure 2.1. A 1-out-of-2 Oblivious Transfer protocol

1. $P1$ generates a boolean circuit representation c of f that takes input i_{P1} from $P1$ and i_{P2} from $P2$.
2. $P1$ transforms c by garbling each gate's computation table, creating garbled circuit c_g .
3. $P1$ sends both c_g and the values for the input wires in c_g corresponding to i_{P1} to $P2$.
4. $P2$ uses *1-out-of-2 OTs* to receive from $P1$ the garbled values for i_{P2} in c_g .
5. $P2$ calculates c_g with the garbled versions of i_{P1} and i_{P2} and outputs the result.

Figure 2.2. Yao's Garbled Circuits Protocol

garbled values of these wires and the boolean values these garbled values represent, $P2$ does not know what values to input into the circuit to match her input bits. In other words, for each input wire into the circuit, $P2$ can select one of two random strings to input (corresponding to 0 or 1), but does not know which of these correspond to her desired input bit.

In order to learn which pseudo-random string to select for each of $P2$'s input wires, $P2$ engages in a *1-out-of-2 OT* with $P1$ for each bit of $P2$'s input. For each round of the OT, $P2$ submits the bit she wishes to learn, receives the corresponding string. Note that the properties of OT prevent $P1$ from learning about $P2$'s input in this process. Once $P2$ has received all of the strings corresponding to her input into the circuit, she holds everything needed to compute the output of the circuit: her garbled inputs, $P1$'s garbled inputs, and the garbled circuit itself. Further, she has obtained these values without $P1$ learning her inputs, nor $P2$ learning $P1$'s inputs.

$P2$ then begins to compute the circuit by entering the pseudo-random strings that correspond to each bit of her and $P1$'s input into the corresponding input wires and using the resulting garbled output string as an input to the next gate. $P2$ may try to learn information about $P1$'s inputs by watching the execution of the circuit. The protocol prevents $P2$ from doing so due to the manner that each computation table for each gate was constructed.

Recall that the computation table for every gate in the circuit was constructed so that each pair of inputs produces an output that represents the correct boolean result, but which appears pseudo-random to $P2$. In other words, instead of mapping from $\{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$, all gates in the circuit become a function mapping two random looking strings to another uniformly distributed pseudo-random string,

or $f(\{0, 1\}^{|k|}, \{0, 1\}^{|k|}) \rightarrow \{0, 1\}^{|k|}$, where $|k|$ is the size of the pseudo-random strings. Since $P2$ never learns the mapping between strings used in the table and their underlying boolean values, $P2$ learns nothing by watching the outputs of each gate.

Recall that the values returned by the output gates in the circuit are not obscured. This results in $P2$ learning the value of $f(i_{P1}, i_{P2})$ once the computation has finished. $P2$ then completes the protocol by sharing this computed value with $P1$.

Detailed Description of the Protocol

Here we provide further details of steps 2 to 5 of Yao's protocol presented in Figure 2.2. Details presented in this section may be omitted in case the reader is only interested in gaining a high-level understanding of the protocol.

Step 2: Garbling Truth Tables: Once $P1$ has constructed a boolean circuit representation c of f , the next step is to garble the truth table for each gate in c , generating a garbled version of the circuit, c_g (i.e. $c \rightarrow c_g$).

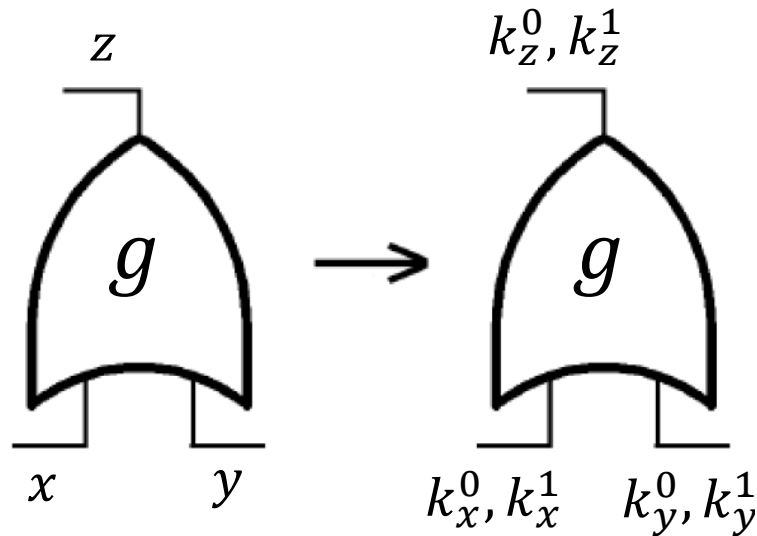


Figure 2.3. Garbling a single gate

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

(a) Original Values

x	y	z	garbled value
k_x^0	k_y^0	k_z^0	$\text{Enc}_{k_x^0, k_y^0}(k_z^0)$
k_x^0	k_y^1	k_z^1	$\text{Enc}_{k_x^0, k_y^1}(k_z^1)$
k_x^1	k_y^0	k_z^1	$\text{Enc}_{k_x^1, k_y^0}(k_z^1)$
k_x^1	k_y^1	k_z^1	$\text{Enc}_{k_x^1, k_y^1}(k_z^1)$

(b) Garbled Values

Figure 2.4. Computation table for g

Lets us consider a single logical OR gate, g , represented in Figure 2.3, where x, y denote the input wires and z denotes the output wire. Initially $P1$ generates the values for this gate as normal, resulting in the truth table in Figure 2.4(a).

Next, $P1$ associates two 80-bit random cryptographic keys k_x^0, k_x^1 with each wire x of the gate (k_x^0 encodes a 0-bit and k_x^1 encodes a 1-bit on wire x). The original truth table, Figure 2.4(a), can now be written in terms of these cryptographic keys producing the first three columns of the table shown in Figure 2.4(b). Next, $P1$ computes the following ciphertexts to produce the values for the ‘garbled value’ column:

$$\text{Enc}_{k_x^i, k_y^j}(k_z^{g(i,j)}), \text{ for all inputs } i, j \in \{0, 1\}$$

The resulting four ciphertexts taken in random order constitute a garbled gate. The collection of all garbled gates forms the garbled circuit c_g . Note, $g(i, j)$ denotes the boolean output of gate g (obtained from its original truth table) for boolean inputs i, j .

This encryption plays two important roles in the protocol. First, since the output of each encryption operation is pseudo-random, it removes any correlation between the underlying truth values in the table and the resulting garbled values. Even though the OR gate produces three identical boolean values, the garbled values are all uniformly distributed, revealing nothing about the underlying value being encrypted.

Second, as shown in ‘garbled values’ column in Figure 2.4(b), the output keys (values in z column) are encrypted using the input keys (values in the x, y columns). Doing so prevents $P2$, the circuit evaluator, from manipulating the circuit structure and using inputs other than those provided by $P1$.

The only gates in the circuit that do not need to be garbled are the output gates, or gates with wires that do not serve as input wires to another gate. The output values from these gates can remain unobscured since they are outputting the final result of the circuit, a value which $P2$ is allowed to learn.

Step 3: Sending Garbled Values to P2: Once $P1$ has finished generating the garbled circuit, he then needs to garble his input to the function, creating a mapping of i_{P1} to its garbled equivalents. $P1$ begins this process by replacing the first bit of his input with the corresponding key for that input wire in the circuit. For example, $P1$ ’s first bit was input into the wire x , and the value of i_{P1}^0 was 1, $P1$ would select k_x^1 to be the first value in his input to the garbled circuit. $P1$ then repeats this procedure for the remaining bits in his input, creating $P1$ ’s garbled input. $P1$ then sends the garbled circuit c_g and his garbled input to $P2$.

Step 4: Receiving P2’s Input Values through OT: $P2$ receives c_g and $P1$ ’s garbled input, but still needs the garbled representation of her own input to compute the circuit. Recall that $P1$ has the garbled values for all of $P2$ ’s input wires, but has no knowledge

of what values correspond to $P2$'s true input. $P2$, inversely, knows the bits of her own input, but not the corresponding keys for her input wires in c_g .

$P2$ maps the first bit of her input to its corresponding garbled value by engaging in a *1-out-of-2 OT* with $P1$, where $P1$'s inputs are k_y^0, k_y^1 (assuming $P2$'s first input bit is mapped on to wire y in the circuit), and $P2$'s input is 0 or 1, depending on the first bit of $P2$'s input. $P2$ performs additional OTs with $P1$ for all values $0 \leq i \leq |i_{P2}|$ to achieve her full garbled input into c_g . The number of OTs performed grows linearly with $|i_{P2}|$. An optimization proposed by [15] to reduce the number of OTs to a constant value, k , which can be set to a small value ($k = 80$ used in this thesis). In Section 2.1.3 we describe this reduction.

Step 5: Computing the Garbled Circuit: Once $P2$ has both garbled inputs and the garbled circuit, she can compute the circuit as follows. For each input gate, $P2$ looks up the corresponding value from $P1$ and $P2$'s garbled input values and uses them as keys to decrypt the output value from the gate's garbled truth table. Since $P2$ does not know which output key these two input keys correspond to, $P2$ must try to decrypt each of the four output keys¹. If the protocol has been carried out correctly, only one of the four values will decrypt correctly. The other three decryption attempts will produce \perp . The newly decrypted key then becomes an input key to the next gate. $P2$ continues this process until she reaches the output wires of the circuit. Each of these wires output a single, unencrypted bit. $P2$ then reassembles the output bits and has the correct solution for the f encoded by c_g . $P2$ completes the protocol by sending the output of the circuit to $P1$.

2.1.3 Reducing the number of OTs

In this thesis we use an optimization proposed by [15] to reduce a large number of OTs to a smaller constant (k) number of OTs. We use $k = 80$ for our implementation.

Consider a general OT primitive, denoted as OT_ℓ^m , which realizes m (independent) oblivious transfers of ℓ -bit strings. That is, OT_ℓ^m represents the following functionality:

Inputs: S holds m pairs $(x_{j,0}, x_{j,1})$, $1 \leq j \leq m$, where each $x_{j,b}$ is an ℓ -bit string. R holds m selection bits $\mathbf{r} = (r_1, \dots, r_m)$.

Outputs: R outputs x_{j,r_j} for $1 \leq j \leq m$. S has no output.

For the use cases described later (Chapter 3) in this thesis, $m \approx 800$ and $\ell = 80$ (fixed). The protocol proposed by [15] reduces OT_ℓ^m to OT_m^k , where k is a security parameter and $m > k$ (for simplicity, we assume ℓ being equal to k). The OT_m^k primitive is implemented as k invocations of a OT_m^1 protocol. The *1-out-of-2 OT* protocol, described in Figure 2.1, is used as the OT_m^1 primitive, where the length of strings (M_0, M_1) is set to m . We describe this protocol in Figure 2.5.

¹In this thesis we use the implementation of garbled circuits described in [16], which reduces the number of decryptions per garbled gate to one.

INPUT OF S : m pairs $(x_{j,0}, x_{j,1})$ of ℓ -bit strings, $1 \leq j \leq m$.

INPUT OF R : m selection bits $\mathbf{r} = (r_1, \dots, r_m)$.

COMMON INPUT: a security parameter k .

ORACLE: a random oracle $H : [m] \times \{0, 1\}^k \rightarrow \{0, 1\}^\ell$.

CRYPTOGRAPHIC PRIMITIVE: An OT_m^k primitive.

1. S initializes a random vector $\mathbf{s} \in \{0, 1\}^k$ and R a random $m \times k$ bit matrix T .
2. The parties invoke the OT_m^k primitive, where S acts as a receiver with input \mathbf{s} and R as a sender with inputs $(\mathbf{t}^i, \mathbf{r} \oplus \mathbf{t}^i)$, $1 \leq i \leq k$.
3. Let Q denote the $m \times k$ matrix of values received by S . (Note that $\mathbf{q}^i = (s_i \cdot \mathbf{r}) \oplus \mathbf{t}^i$ and $\mathbf{q}_j = (r_j \cdot \mathbf{s}) \oplus \mathbf{t}_j$.) For $1 \leq j \leq m$, S sends $(y_{j,0}, y_{j,1})$ where $y_{j,0} = x_{j,0} \oplus H(j, \mathbf{q}_j)$ and $y_{j,1} = x_{j,1} \oplus H(j, \mathbf{q}_j \oplus \mathbf{s})$.
4. For $1 \leq j \leq m$, R outputs $z_j = y_{j,r_j} \oplus H(j, \mathbf{t}_j)$.

Figure 2.5. Reducing OT_ℓ^m to OT_m^k

Notation for the protocol: We use capital letters to denote matrices and small bold letters to denote vectors. We denote the j th row of a matrix M by \mathbf{m}_j and its i th column by \mathbf{m}^i . The notation $b \cdot \mathbf{v}$, where b is a bit and \mathbf{v} is a binary vector, should be interpreted in the natural way: it evaluates to 0 if $b = 0$ and to \mathbf{v} if $b = 1$.

2.2 Head detection in I-Pic

In this section we describe the head detector [17] used in Chapter 4 where we explore performance limits of the I-Pic prototype developed in Chapter 3.²

Head detection refers to detecting the presence and location of heads in an image. Head detection produces bounding boxes enclosing a rectangular area in an image where a head might be present. Head detectors are special cases of generic object detectors that have been specialized for detecting heads only. In this thesis, we use a head detector [17] that is based on the popular object detection framework called Faster R-CNN[19].

In the next section (Section 2.2.1) we will first describe some of the related work pertaining to recent advances in object detection techniques, followed by (Section 2.2.2) a description of the head detector [17] used in Chapter 4.

²The I-Pic prototype(developed in Chapter 3) uses an open source face detector called HeadHunter [18], which is described in Chapter 3.

2.2.1 Related work: advances in object detection in images

In this section we briefly describe the recent advances in object detection techniques for still images. The details presented in section may be omitted in case the reader is only interesting in gaining a high level understanding of the head detector described in Section 2.2.2.

Up until 2012, progress on various visual recognition tasks was largely based on the use of SIFT [20] and HOG [21] features. Using these features, the improvements in accuracy on tasks such as PASCAL VOC object detection [22] were small, and were mostly obtained by building ensemble systems and employing variants of successful methods. HeadHunter [18], the face detector used in I-Pic, was also based on ensemble systems.

In 2012, AlexNet [23] took a radically different approach, and demonstrated that accuracy of computer vision tasks could be substantially improved by leveraging Convolutional Neural Networks (CNNs). Specifically, AlexNet achieved a whopping 40% improvement over the previous best result for image classification on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset [24, 25]. Since then, CNNs (and deep learning) have become the de-facto standard for implementing computer vision tasks. In the following we describe related work specific to object detection based on deep learning techniques.

Multi stage object detection using deep learning: In 2014, R-CNN [26] explored the possibility of generalizing AlexNet’s classification results to object detection tasks. They showed that CNNs provide dramatically higher object detection performance as compared to prior systems based on HOG-like features. The R-CNN (Region-based Convolutional Network) operated in two stages: in the first stage it used an existing object proposal system, called Selective Search [27], to generate a set of bounding boxes that might contain an object in them. In the second stage, each object proposal was processed using a CNN to extract convolutional features for that object proposal. These features were then used for classifying object proposals into different object categories (using pre-trained SVMs). After this classification, the same CNN features were used, in conjunction with a linear regression model, to output a tighter bounding box for each object proposal. Using this multi-stage approach, R-CNN improved object detection accuracy by more than 30% compared to the previous best results. This improvement in accuracy, unfortunately, came at the cost of substantial increase in computational complexity. This was because R-CNN required forward passing every object proposal, around 2000 per image, through the R-CNN network. Another downside to R-CNN was that it was very time consuming to train, as it required training three different models separately.

In 2015, Fast R-CNN [28] combined R-CNN's three models into a single CNN, which resulted in 9x faster training times. Furthermore Fast R-CNN significantly reduced the repeated processing required for each object proposal by reusing intermediate CNN feature maps for subsequent processing. As a result Fast R-CNN performed 213x faster at test-time.

In 2016, Faster R-CNN [19] showed that it was not necessary to use a separate system for generating object proposals. Instead, it integrated a region proposal network (RPN) into the existing Fast R-CNN network (FRN). The Faster R-CNN network contains two branches that share the majority of their computations, branching at the end into separate RPN and FRN stages. The common portion of the network starts with a set of dummy proposals (called anchors) and the input image, to compute a CNN feature map that is used by both RPN and FRN. RPN produces a set of object proposals that are subsequently used by FRN (along with the pre-computed CNN feature map) to refine them into tighter bounding boxes, and to classify them. The resulting network operates 10x faster at test time than Fast R-CNN, provides better object detection accuracy, and requires approximately 6 times fewer object proposals than Fast R-CNN.

R-FCN (Region-based Fully Convolutional Network) [29] further improved on Faster R-CNN's runtime performance, by pushing the object proposal specific repeated computations to the last layer possible. Instead of cropping feature maps from the layer where the RPN stage branched off, R-FCN extracted object proposal specific feature vectors from the last feature layer prior to prediction. As a result, R-FCN achieved upto 20x faster test-time runtime, while maintaining accuracy comparable to Faster R-CNN.

Single stage object detection using deep learning: Faster R-CNN [19] and R-FCN [29] provide good accuracy for object detection but are still considered too computationally intensive for embedded systems and too slow for real-time applications. For example, Faster R-CNN achieves a maximum of 7 frames per second on desktop grade GPUs [30]. Single shot detectors, on the other hand, such as SSD [31] and YOLO [32], explore the possibility of trading off accuracy for speed (and lower computational complexity). Both systems, SSD and YOLO, eliminate the bounding box proposals and the subsequent feature re-sampling stage altogether, and instead use a single feed-forward convolutional network to directly predict classes & anchor offsets. Both approaches divide an image into a fixed sized grid. For each grid cell they predict a fixed number of bounding boxes and scores that indicate the presence of objects in those boxes. Such a pipeline, on the one hand, provides significant speeds ups, both YOLO and SSD can operate at 45 frames per second. On the other hand, both approaches impose strong spatial constraints on bounding box predictions, which limits the number of nearby objects they can predict. Despite these limitations, these systems are promising alternatives for achieving real time detection on mobile devices.

Instance segmentation: All the above approaches identify a bounding box around the object, and do not identify the exact object boundaries. More recent research efforts, such as Mask R-CNN [33] and DeepMask [34], have also focused on instance segmentation, which refers to the correct detection of all objects in an image while also producing the exact shape of an object. Knowing which pixels fall within the exact shape of an object could be particularly useful for cases where these pixels need to be post-processed, e.g., for blurring out an object from an image by distorting the pixels falling within the object.

2.2.2 Head detector description

The head detector we used [17] is based on the Faster R-CNN [19] framework. The head detector was trained using 7,372 head images extracted from 10,103 images available in the PASCAL VOC 2010 trainval set [35]. Images that did not have people in them were retained as a source of negatives. The training was modified from the original Faster R-CNN configuration to make it more suitable for head detection. Specifically, to account for small heads, small scale dummy anchor boxes were added during the training. The resulting network was specialized for detecting heads that appeared both in foreground & background, and could also detect heads when they were turned more than 90 degrees in profile, including heads from behind. The training was carried out using the Matlab extension of the Caffe framework [36]. The trained detector is made available as two Caffe model files, corresponding to the two stages of the detector, and Matlab code that executes the two models in succession and performs some of the post processing steps.

The first stage of the detector, called the region proposal network (RPN), takes as input the image to be processed and produces approximately 30,000 proposal bounding boxes and corresponding scores. The score indicates the probability of a head being present in a proposal box. These boxes are then processed by Non-maxima suppression (NMS) using the algorithm described in R-CNN [26]. This is followed by eliminating boxes based on their scores and retaining only the top-K. NMS is a post-processing algorithm responsible for reducing multiple detection boxes that belong to the same object down to a single box. NMS rejects a box if it has an intersection-over-union (IoU) overlap larger than a certain threshold with a higher scoring box. Higher values of NMS threshold tend to produce multiple boxes for a single object. For the first stage, the NMS threshold was set to 0.7 and the value of K, for the top-K step, was 2000.

The second stage of the detector, called the Fast R-CNN network (FRN), takes as input the 2000 object proposals and a feature map output by the last convolutional layer of the RPN. The FRN stage produces a tighter bounding box for each object proposal along with a probability score. These boxes are post-processed using NMS and the top-

K steps, with thresholds of 0.6 and 300, respectively. The boxes, along with their score values, remaining after this post-processing step is the output of the head detector.

The output of the head detector may still contain multiple boxes detected for a single head. Further lowering the NMS threshold (below 0.6) for the FRN stage reduces the number of boxes produced per head, but these produced boxes often do not overlap with the actual head properly. In Chapter 4 we describe some additional post-processing steps, not part of the head detector, that were applied to the output of the head detector to reduce duplicate boxes.

Chapter 3

I-Pic: A Platform for Privacy-Compliant Image Capture

3.1 Introduction

Smart phones and wearable devices like smart glasses have audiovisual recording equipment that can be operated near continuously. These devices enable a wide range of novel applications and services, including location-, situation-, and activity-aware services, augmented reality based services, and lifelogging.

At the same time, these devices pose serious new risk to users' privacy and security. Bystanders may be recorded (intentionally and or inadvertently) without their consent. Objects (e.g. defense installations), activities (e.g. screening procedures at an airport), and information (contents of a whiteboard or monitor, paper on table) may be recorded illegally or in violation of corporate policy.

Traditional ways to dealing with these types of threats are inadequate. Devices that integrate recording with other capabilities are now ubiquitous, and it is not obvious to a bystander if a smart glass, for instance, is presently recording or not. It would be cumbersome and socially awkward for bystanders to voice their preferences to anyone wearing a potential recording device. Similarly, it may be awkward or impossible to ask visitors to relinquish or switch/stow their phones and wearables while in a space that may have some recording restrictions (e.g., an airport).

Instead recording devices should be able to sense the privacy preferences of nearby users and organizations automatically, and enforce these preferences in their platform software. Moreover, the techniques used to disseminate privacy preferences must not by themselves reduce users' privacy by enabling tracking or identification of the user. Developing the principles, methods, and protocol underlying this capability, as well as prototype systems, is the goal of this project.

Towards that end we built I-Pic, a platform for policy-compliant image capture, whereby captured images are automatically edited according to the privacy choices of individuals photographed. I-Pic’s design was motivated by a user study, described in Section 3.3, which found that:

Capture policies should be individualized: Privacy concerns vary between individuals. Even in the same situation, different subjects have different preferences. This finding motivated I-Pic to preclude options that impose blanket or venue specific policies [37, 38, 39].

Policies should be situational: Study subjects stated consent to be photographed at certain times, places, events, or by certain photographers, but would make different choices in other circumstances. This motivated I-Pic to not impose a static policy per individual [40], and to avoid solutions that require prior arrangements between specific subjects and photographers (whitelisting or blacklisting).

Compliance by courtesy is sufficient: An overwhelming majority of our subjects stated that they would choose to comply with the privacy preferences of friends and strangers, especially if doing so didn’t interfere with the spontaneity of image capture. I-Pic provides such a platform but is not meant to stop determined users from taking pictures against the wishes of others; indeed, these users could simply use a non-I-Pic compliant device.

Consider a strawman system where mobile devices broadcast their owner’s privacy preferences via Bluetooth. Without additional information, a camera would have to edit the image according to the most restrictive policy received, even if the corresponding person does not appear in the image at all! To be practical, policies must be accompanied by a visual signature so that a camera can associate a person captured in an image with a policy.

However, Bluetooth transmissions can cross walls, which would create a serious privacy problem if visual signatures were broadcast in the clear: Next-door neighbors could identify persons whom they have never seen or photographed! To avoid this problem, I-Pic relies on secure multiparty computation (MPC) to ensure that a capture device learns only a person’s privacy choice, and only if that person was captured; otherwise, neither side learns anything.

User studies and privacy requirements inform the architectural components of I-Pic: Users advertise their presence over BLE(Bluetooth Low-Energy): these broadcasts are received by I-Pic-compliant capture platforms. When an image is taken, the platform determines if any of the captured people match the visual signatures of nearby users using MPC. If there is a match, the platform learns the policy and edits the image

accordingly, e.g., by occluding the person’s face. To maintain the responsiveness of image capture, unedited images are shown to the photographer immediately, but cannot be shared until the image is processed in the background.

Next we describe I-Pic’s related work in Section 3.2, followed by results of our online survey in Section 3.3. After that we describe the main technical design of I-Pic in Sections 3.4 and 3.5, along with existing work in face recognition and cryptography we build on. Finally, we present results of an experimental evaluation in Section 3.6.

3.2 I-Pic Related work

Privacy in the presence of recording devices: Hoyle et al. [41] seek to understand users’ concerns about continuous recording using wearable cameras, by studying a large user population of avid life-loggers. Denning et al. [42] conduct a large scale user survey to understand bystanders’ privacy concerns in public places like coffee shops and possible ways to mitigate them. Our online survey additionally shows that privacy concerns are very personal and dependent on the situation.

Roesner et al. [38] present a system that shares a venue’s privacy preferences with wearable devices in an unobtrusive way. The idea is to convey privacy expectations associated with places like gyms and washrooms with broadcast messages or visual signs. The wearable devices in the venue pick up these messages or visual cues and obey the specified privacy protocol. Unlike I-Pic, this system has no way to associate a privacy policy with an object or person that appears in an audiovisual recording.

Visual markers to convey privacy policies to nearby wearable recording devices are also used in [39]. [43] explores the expression of bystanders’ privacy intent using gestures. Unlike I-Pic, these approaches require either physical tagging of objects and locations, or explicit user actions (i.e., gestures) to convey privacy choices. Moreover, I-Pic enables user-defined, personalized, context-dependent privacy choices.

In the work by Bo et al. [40], individuals wear clothes with a printed barcode, which encodes the wearer’s public key. When an image of an individual showing face and barcode is uploaded to an image server, the server garbles the face pixels, using the public key encoded in the barcode. Only the individual who owns the associated private key can later extract the actual face image. I-Pic, on the other hand, does not require its users to wear any visual markers, it does not require users to trust an image server with their private images, and can support context-dependent privacy policies.

In [44, 45, 46], the authors address privacy concerns in untrusted perceptual and augmented reality applications, by partially processing media stream within the trusted platform, thus denying apps access to the raw media streams. An augmented reality app, for instance, might be provided only with the position of relevant objects within a video stream sufficient for the app to overlay its own information, but not the full video.

I-Pic also relies on the trusted platform, but focuses on enforcing individual’s privacy policies regarding image capture by nearby devices.

Zero-Effort Payments [47], similar to I-Pic, uses face recognition and proximate device detection using BLE to identify a user in an image, but their goal instead is to create a mobile payment system. Unlike I-Pic, which is tuned to identify even small faces in diverse range of photographic contexts, their system is meant to visually identify a user, with human assistance, when she is in close proximity to the cashier. Furthermore, they acknowledge concerns of user privacy in such a monitored environment and propose the use of signage indicating that a face recognition system is deployed in the area. Such a privacy solution is only viable in select scenarios, and lacks the flexibility provided by I-Pic.

Visual fingerprints: Performance on human identification and re-identification tasks has greatly improved over the last decade. Most notably, face recognition on large databases in realistic settings is even approaching human performance [48]. Besides the identity, a person can also be described and identified by a set of attributes [49, 50]. I-Pic uses a state of the art face recognition algorithm based on neural networks, but can benefit from using semantic attributes describing a face, including features from other body parts in addition to the face.

Cryptographic primitives: There is complementary work to protect the privacy of biometric data [51, 52] by projecting or encrypting representations. It is possible that these approaches could be used in I-Pic to further reduce trust in the Cloud service by obscuring users’ visual signatures.

InnerCircle [53] describes a secure multi-party protocol for location privacy, which computes in a single round whether the distance between two encrypted coordinates is within some radius r . This computation is similar to I-Pic’s secure dot product and thresholding computation. However, the protocol’s efficiency degrades exponentially with the number of bits of precision of the distance. Since our threshold comparison involves dot products of large feature vectors, we use garbled circuits for the threshold comparison instead.

3.3 Online Survey

I-Pic’s design was informed by an online survey designed to provide a broader perspective on personal expectations and desires for privacy. The survey, and experiments with I-Pic, were conducted with user consent under an IRB approval from the University of Maryland. The survey included an optional section on user demographic, including gender, age, and ethnicity.

We publicized the survey on mailing lists and online social networks on November 10th, 2015. The survey is available online at <http://goo.gl/forms/6tGG0YmFFG> (and

reproduced in Appendix A), and the results here present a snapshot of all responses collected on December 4th, 2015. As of this date, there were 227 responses, with 208 responders also answering the demographic questions. Respondents represented 32 countries. The age distribution is shown in Table 3.1.

Age group	Fraction of participants
less than 20 years	9.2%
20 - 30 years	56.6%
30 - 40 years	25.1%
40 - 50	4.8%
more than 50 years	3.9%
Unspecified	0.4%

Table 3.1. Age groups of survey participants

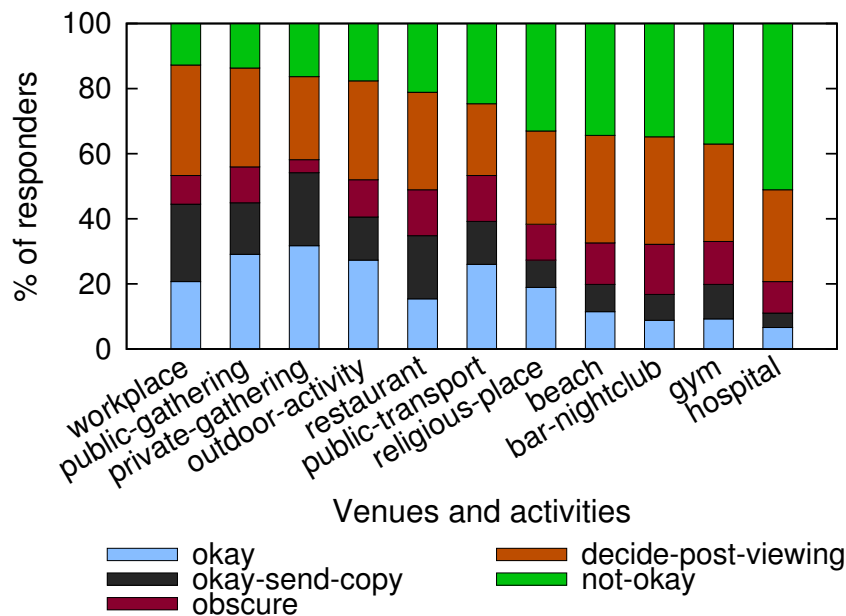


Figure 3.1. Variety in privacy preferences: based on physical situations

Questions in the survey envisioned different venues and activities and presented participants with different privacy options: (a) agree to be captured in any photograph, (b) agree, but would like a copy of the image, (c) please obscure my appearance in any image, (d) can decide my preference only after viewing the photo, or (e) do not wish to be captured in any photograph. Participants were asked to choose the privacy action they considered most appropriate for each scenario (Figure 3.1). To help visualize a common scenario and to provide perspective for others, participants were shown an image of people on a platform waiting to board a train, some with faces clearly visible. The survey also gauged individual’s level of comfort depending on their relationship to the photographer or the other subjects in the photograph (Figure 3.2). Finally, we

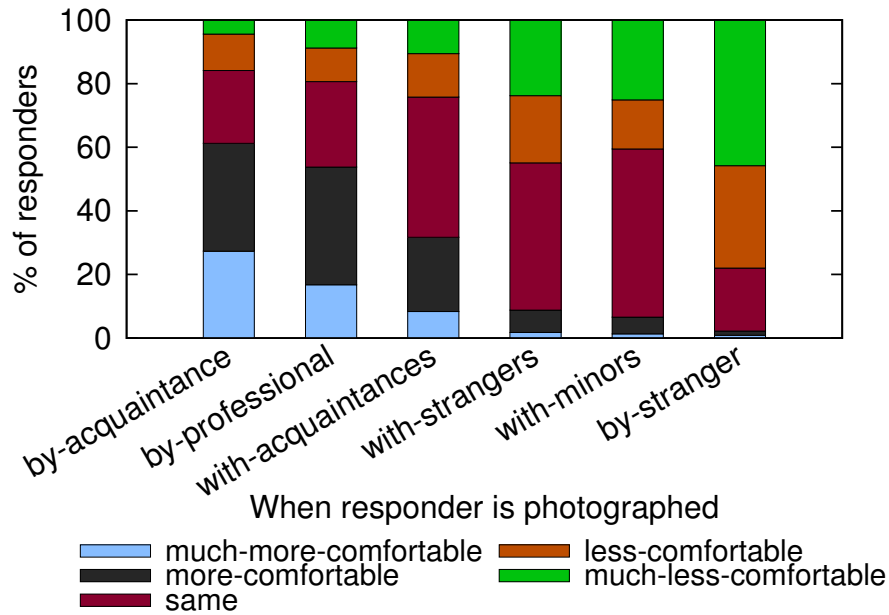


Figure 3.2. Variety in privacy preferences: based on social situations

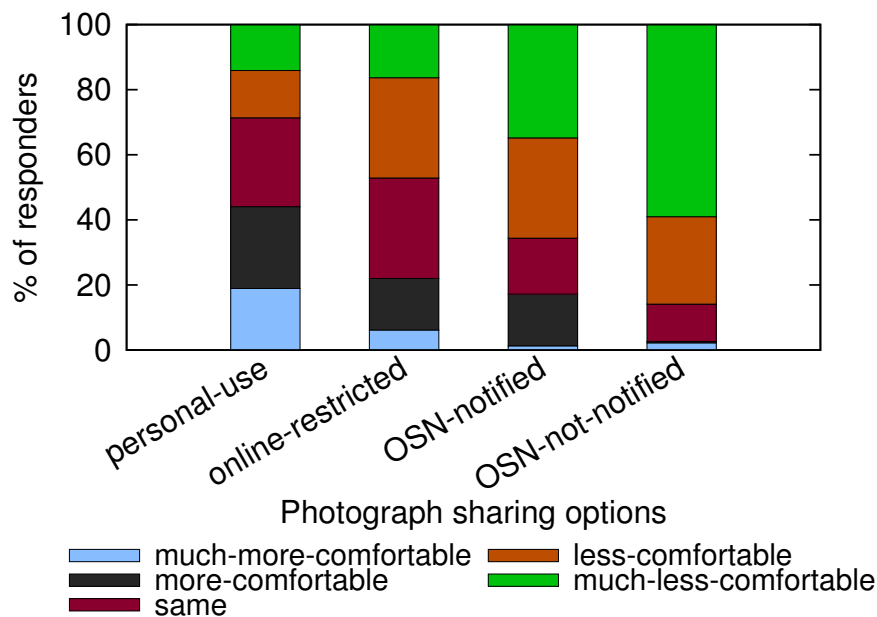


Figure 3.3. Variety in privacy preferences: based on use of images (OSN = Online Social Networks)

asked how potential uses of an image influence responders' level of comfort with being captured (Figure 3.3).

In Figure 3.1, the x-axis is sorted by the percentage of responders who chose the most private action of "do not wish to be captured", increasing from left to right. Our results show a mix of privacy concerns for different scenarios. In Figures 3.2 and 3.3, the x-axis is sorted by the percentage of responders who were much less comfortable with photography, increasing from left to right. Once again, for these social situations

or image usage scenarios, the privacy concerns of responders is not uniform. *These results demonstrate the necessity of diversity in privacy policy, and argue against venue based policies that cannot be customized for individuals [38, 54].*

Unsurprisingly, privacy preferences are not unanimous for any scenario; there are, however, trends. Responders tend to be more restrictive in venues such as beaches, gyms and hospitals (in Figure 3.1); with strangers in a social situation (in Figure 3.2); and when images can potentially be shared online (in Figure 3.3(c)). These trends can be useful as they suggest default policies appropriate for different situations.

Number of privacy preferences	Fraction of participants
1	12.7%
2	27.8%
3	32.2%
4	19.4%
5	7.9%

Table 3.2. Variety in privacy preferences for same person

Table 3.2 shows the percentage of responders versus the number of different privacy choices for each responder. The table shows that individuals prefer different privacy choices depending on the given situation. *This finding illustrates the utility of context-specific policies, and demonstrates the shortcomings of individualized hardcoded policies, e.g., bar-codes on clothing [40].*

The survey asked whether responders cared about *by-stander* privacy when respondents themselves capture images. An overwhelming majority (96.47%) answered in the affirmative, motivating a system such as I-Pic. About a quarter (28%) agreed if the overhead of the solution was low; another quarter (26%) agreed if the aesthetics of images remain good.

Respondent Selection Bias The survey was voluntary and anonymous. The URL for the survey was advertised on mailing lists and social networks used by the authors and their friends, leading to a bias in how respondents learned about the survey. However, we believe that the results presented here still have merit as they represent views across different age groups and ethnicities. The results overwhelmingly support the thesis that users often desire privacy from digital capture in social situations, and further that “one-size-fits-all” solutions to image privacy are not effective. Moreover, as photographers, the responders overwhelmingly consider bystander privacy to be important. These observations inform I-Pic’s architecture, described next.

3.4 I-Pic Architecture

3.4.1 I-Pic overview

Figure 3.4 shows I-Pic’s major components and their interaction. The two types of principals in the system are *bystanders* or users who may be photographed, and *photographers* who capture images. Both are assumed to operate an I-Pic-compliant *platform*. Associated with each principal is a cloud-based *agent* to which the principals offload compute-intensive tasks. The photographer is associated with a *Capture Agent*; each bystander is associated with a *Bystander Agent*. We note that agents are logical constructs; functions provided by the agent can be implemented within mobile devices should I-Pic be used without wide-area connectivity.

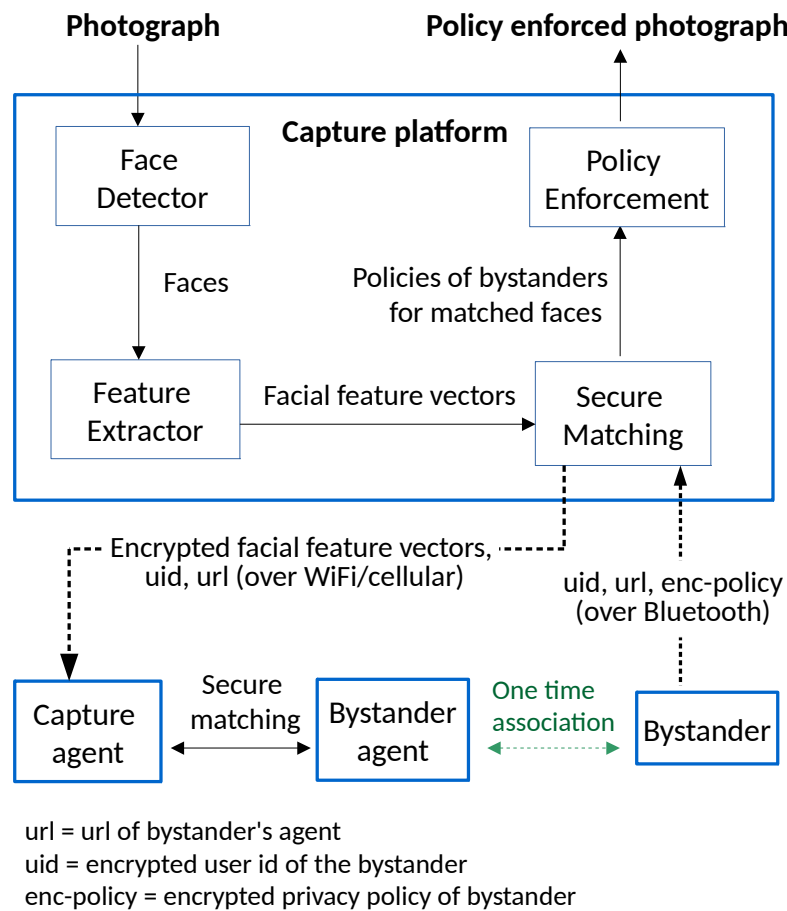


Figure 3.4. I-Pic major components

I-Pic requires a one-time *Association* protocol between users and their agent. Users *periodically broadcast* their presence using BLE. Once an image is captured, the *Face Detection*, *Feature Extraction*, and *Secure Matching* protocols are executed. If a user is identified, the capture platform uses the *Policy Enforcement* protocol to modify the photograph as requested. We describe these sub-protocols next.

Association: Users select an agent as a proxy and provide it with photographs, which are used to train an SVM classifier for face recognition. A user trusts her agent not to leak her visual signature. The association protocol also exchanges a master key between agent and user’s device, which is used to generate session keys in the future.

Next, users initialize their privacy profile, which is locally stored on their device, by choosing relevant contexts based on location (e.g. office, home, gym, bar/restaurant, public spaces) and time (work hours, off-work hours), and by choosing an appropriate action for each context (agree to appear with face, blur face).

Periodic Broadcast: Users periodically broadcast an encrypted policy that specifies how to treat the user’s picture if she appears in a photograph. This broadcast also includes sufficient information to identify the user’s agent. The policy is encrypted with a session key generated using the current time (divided into 15-minutes epochs) and the master key exchanged with the user’s agent.

Capture platforms receive and cache policies. Once a photograph is captured, if a user is identified, then the associated policy can be decrypted.

Secure Matching: Upon image capture, the platform detects and tries to recognize faces. These components leverage existing prior work in face detection [18] and facial feature extraction [55], detailed in Section 3.5.1.

The capture platform encrypts the extracted features and uploads them to its agent, along with the network identifiers of all bystander agents that it has received as broadcast recently. The *Capture Agent* and the *Bystander Agent* compare extracted features and a bystander’s classifier weight vector by implementing a secure dot-product protocol [56] followed by a secure threshold comparison protocol based on garbled circuits [57]. If the threshold passes, then the session key used to encrypt user’s policy is revealed to the capture platform.

Policy Enforcement: When granted a session key for a user, the capture platform decrypts the corresponding user’s privacy policy and performs the action requested. Our current implementation only supports face obfuscation, which we implement using the OpenCV library. More sophisticated techniques exist. For instance, it is possible to morph a face into another face [58] instead of blurring it. Furthermore, it is also possible to remove an entire body from an image and extrapolate the background so that the removal is not obvious [59]. While such advanced image processing techniques are not the subject of this thesis, I-Pic can take advantage of them.

If a captured face cannot be matched against any bystander, but all advertised policies have been evaluated, I-Pic defaults to blurring the face. This protects the privacy of bystanders who either do not own a smart device or are not I-Pic users.

Similarly, all unmatched faces are blurred if the identification protocol does not complete for some policies, likely due to lack of network connectivity. The platform

maintains an encrypted copy of the original image, which can be used to release an unblurred face in the original image as the protocol completes in the future.

3.4.2 Threat model

I-Pic’s cryptographic protocols ensure that a non-compliant capture device cannot learn the feature vectors of a bystander who does not appear in a captured image. For privacy policies of bystanders to be correctly applied, the capture platform on users’ devices is assumed to implement the I-Pic protocol correctly. Third-party applications installed on users’ devices are untrusted.

Users of capture devices may be able to bypass I-Pic by “rooting” their device; a different implementation could integrate I-Pic into the device firmware or implement the protocol on a trusted hardware platform, thus raising the bar for bypassing I-Pic’s privacy protection. We dismissed this approach, because uncooperative photographers could in any case use a non-I-Pic compliant camera. Our goal instead is to enable cooperative photographers to respect bystander’s privacy wishes in an unobtrusive manner, without introducing new attack vectors. We believe that most users welcome the ability to automatically comply with bystander’s wishes, as it enables them to take pictures freely, without worrying whether they might offend others. This was also observed in our online survey(Section 3.3), where 96% of the participants indicated that they cared about bystanders’ privacy.

The *Bystander Agent* must be trusted by the bystander not to leak her visual signature. The *Capture Agent*, on the other hand, does not have access to either the users’ visual signature stored on the *Bystander Agent* or the features vectors extracted by the capture device. However, *Bystander Agent* and *Capture Agent* are assumed not to collude, else they could jointly extract the feature vectors of people captured in an image. *Capture Agent* is additionally expected to construct the garbled circuit used for secure threshold comparison (described in section 3.5.2) accurately.

Cloud agents learn when an I-Pic compliant device captures an image, and the *Capture Agent* learns the IP address of that camera device (Technically, both could be spoofed since the request may use an identifier without capturing an image, and the source IP address in a request could be that of a forwarding relay). I-Pic protocols are designed to ensure that the cloud agents do not learn if a user appears in an image, or the user’s current context or policy. The following Section 3.5 describes the I-Pic protocols in detail.

3.5 I-Pic Design

Next, we describe the design of I-Pic in more detail. Figure 3.5 shows the I-Pic workflow in normal operation.

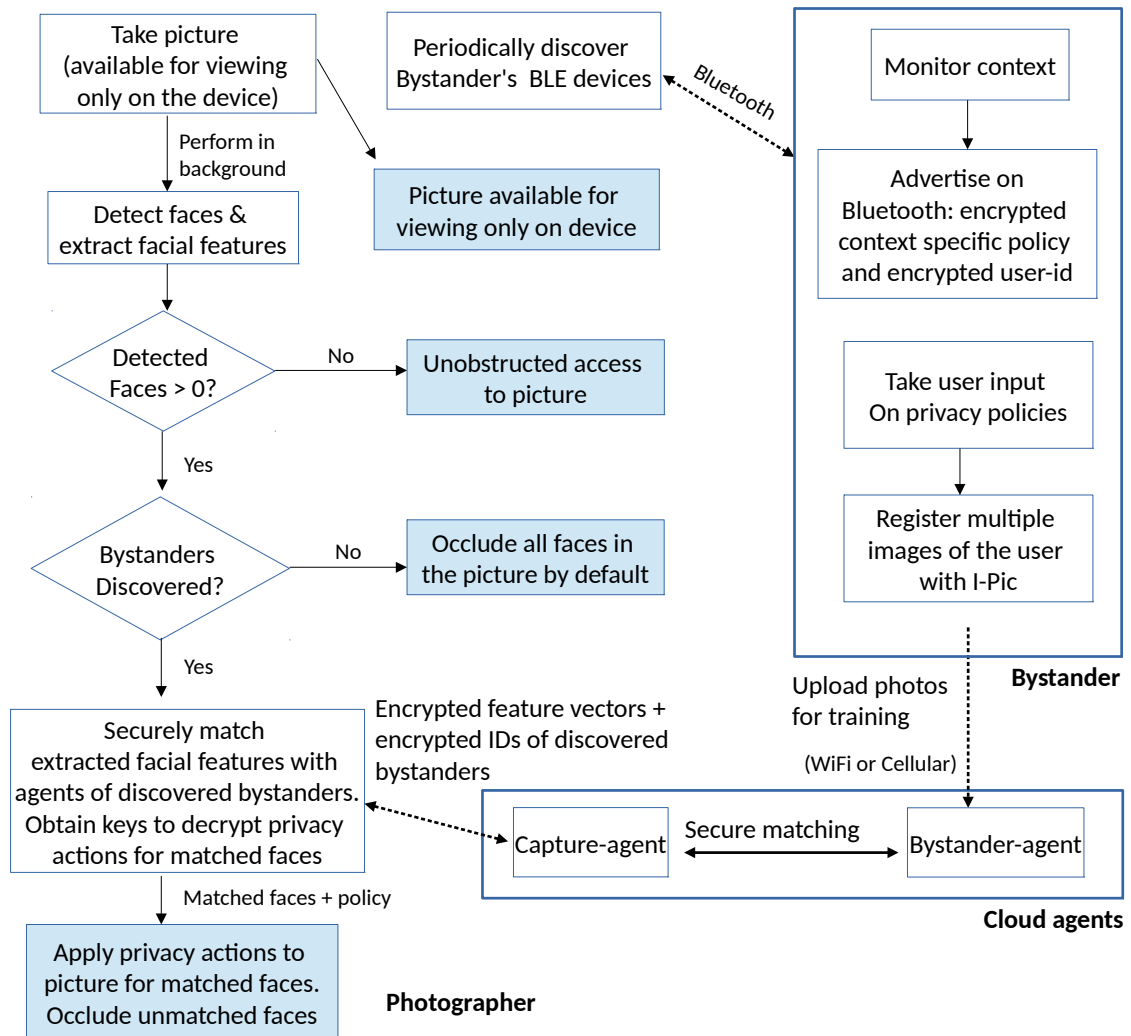


Figure 3.5. I-Pic workflow

I-Pic compliant devices broadcast their encrypted $(userid, policy)$ pairs periodically. I-Pic compliant capture devices additionally discover other Bluetooth devices periodically and add any received pairs to a local cache of nearby users. The entries are flushed from the cache when a device's broadcast has not been received for 10 minutes.

When an image is captured, I-Pic intercepts the raw image data. The captured image is available for viewing immediately but cannot be shared until the image is processed. A background task runs the vision pipeline described below in Section 3.5.1 to detect faces and extract feature vectors for each. Next, for each feature vector extracted from the image, the background task performs the secure matching protocol described below in Section 3.5.3 to determine if it matches with the registered classifiers of any of the bystanders in the cache, and decrypts the policies of any matching bystanders.

Finally, the I-Pic background task edits the image according to the policies of the users captured in the image. By default, any face detected in the image that did not match the signature of a bystander is occluded. This conservative choice errs on the side of privacy in case of a bystanders who does not carry a mobile device or does not use I-Pic, whose BLE broadcast was not received, or whose visual signature did not match due to a false negative of the face recognition.

3.5.1 Image processing

The goal of I-Pic’s image processing is to identify people captured in the image, extract visual signatures for each person, and match these signatures with those advertised by nearby bystanders.

Detecting and recognizing people in images is an active area of research in computer vision. The current I-Pic prototype relies on face recognition as a well-understood and natural technique for detecting and recognizing people. More general techniques for people detection and recognition based on full-body visual signatures can be integrated into I-Pic in the future.

In the following, we briefly describe I-Pic’s face detection, feature extraction, and face recognition pipeline.

Face detection: I-Pic must detect faces with high recall, ensuring that bystanders’ faces are detected with high probability regardless of size, focus, pose, angle, lighting, or partial occlusion. Unlike the primary subjects of an image, bystanders are not posing for the camera, may be in the background, poorly lit, or out of focus, which makes their detection challenging.

We use the open source HeadHunter [18] face detector. HeadHunter achieves face detection recall of $\sim 95\%$ on standard image datasets like the Annotated Faces in the Wild (AFW) [60]. For I-Pic, we ported HeadHunter to a mobile tablet with a GPU, as described in Section 3.6. As we will show in Section 3.6.4, HeadHunter is superior to other face detectors available for mobile platforms.

Feature extraction: We use the state of the art person recognition method from [55]. Unlike typical face recognition systems that can recognize only the frontal faces, [55] person recognition system has been trained to generalize across head pose by utilizing hairstyle and context information. Due to this generalization, it outperforms other cutting-edge face recognition systems in a social media photo setting [61], where individuals often do not pose for the camera. Since I-Pic aims at identifying bystanders, this person recognition system is highly relevant.

The person recognition system [55] is based on a convolutional neural network (AlexNet [23]) pretrained on the ImageNet [62] classification task, and fine-tuned for the person identification task on People In Photo Albums (PIPA [61]), a large database

of people in social media photos. While [55] uses five different body regions (face, head, upper/full body, and scene) to maximize the performance, we only extract features from the face region, and denote this cue as FNet.

Given a face, the original FNet extracts a 4096-dimensional feature vector. To ensure the efficiency of the secure matching algorithm, which is inversely proportional to the number of dimensions, we reduce this feature vector to 128 dimensions. We found that using the neural network itself for dimensionality reduction results in a smaller drop in overall recognition accuracy than using Principal Component Analysis. Specifically, we insert a 128-dimensional fully connected layer before the last layer in the AlexNet, randomly initialize the weights, and tune it using Stochastic Gradient Descent. Our FNet features are extracted from this 128-dimensional layer after forward passing Headhunter face detections through the network. All the training and feature extraction in neural networks are done using the open source deep learning framework Caffe [36].

Face recognition: When a user registers, I-Pic extracts FNet features from the set of portraits he or she provides. Per-user SVM classifiers are then trained on the FNet features, where positive examples consist of the portraits provided by the corresponding user, and negative examples from the other users and $\sim 12\text{K}$ celebrity faces in the Labeled Faces in the Wild dataset (LFW) [63]. On average, there are ~ 15 positive examples per user, captured with different viewpoints and facial expressions. Users may subsequently provide additional images for training, for instance, if they start to wear glasses or grow a beard. The liblinear [64] package has been used to train the SVMs.

In normal operation, HeadHunter detects faces in captured images, and the corresponding FNet features are extracted. I-Pic compares the feature vector of each detected face against the trained SVM classifiers of each bystander using a dot product computation. If the dot product is above a certain threshold, the classifier indicates a match. To ensure privacy, I-Pic computes the dot product and threshold comparison as part of a secure multi-party computation between the photographer’s capture agent and each bystander’s agent.

Before we describe the secure matching protocol, we briefly review the underlying cryptographic protocols.

3.5.2 Cryptographic Protocols

I-Pic composes two standard protocols to achieve secure matching: secure dot product and garbled circuits.

Secure dot product: The secure dot product protocol allows two parties, each with a private vector, to compute the vector dot product without divulging the vectors. We use the protocol described in [56], which is based on the Paillier homomorphic encryption

scheme [65]. We use the notation $\llbracket a \rrbracket_{pk}$ to represent the encryption of a number a using a public key pk . The Paillier encryption scheme is additively homomorphic, i.e., given $\llbracket a \rrbracket_{pk}$ and $\llbracket b \rrbracket_{pk}$, it is possible to compute $\llbracket a + b \rrbracket_{pk} = \llbracket a \rrbracket_{pk} \llbracket b \rrbracket_{pk}$. It follows that given $\llbracket a \rrbracket_{pk}$ and an integer c , one can compute $\llbracket ca \rrbracket_{pk} = (\llbracket a \rrbracket_{pk})^c$. These two primitives can be combined to compute the dot product securely as follows: Given two vectors $v_a = [v_{a_1}, v_{a_2}, \dots, v_{a_m}]$ and $v_b = [v_{b_1}, v_{b_2}, \dots, v_{b_m}]$, the dot product $v_a \cdot v_b = \sum_{j=1}^m (v_{a_j} v_{b_j})$. Given the Paillier encryption scheme, one can compute the encrypted dot product of an encrypted vector $\llbracket v_a \rrbracket_{pk}$ and a cleartext vector v_b as

$$\llbracket v_a \cdot v_b \rrbracket_{pk} = \left[\sum_{j=1}^m (v_{a_j} v_{b_j}) \right]_{pk} = \prod_{j=1}^m (\llbracket v_{a_j} \rrbracket_{pk})^{v_{b_j}}$$

A straightforward application of this protocol in I-Pic, however, faces two problems: First, the capture device learns the dot products, which would enable a ‘rogue’ capture device to learn the classifier weight vector of each bystander. By computing dot products using a series of standard basis vectors (vectors that have a value of one in one dimension and zero in all others), the dot product values reveal the dimensions of a bystander’s weight vector. To prevent this attack, we use garbled circuits [57], described below (and in detail in Section 2.1), to compute whether the dot product exceeds a threshold \mathcal{E} without revealing the dot product itself.

Second, a capture device typically needs to compare several feature vectors, corresponding to multiple faces that appear in a photo, to the classifier weight vector of a bystander. For n feature vectors with m dimensions, the secure dot product computations require nm encryptions (and n decryptions). We can optimize this computation as follows.

Optimized $n \times 1$ secure dot product: I-Pic reduces the number of encryptions from nm to m using ideas from [66]. Consider a matrix V of n vectors with m dimensions each, corresponding to n faces in a photograph, where $V_{i,j}$ is the j th element in the i th vector. Let $c_j = [V_{1,j}, V_{2,j}, \dots, V_{n,j}]$ be the j th column of V . The photographer computes an encryption of c_j as $\llbracket c_j \rrbracket_{pk} = \llbracket (V_{1,j} \parallel V_{2,j} \parallel \dots \parallel V_{n,j}) \rrbracket_{pk}$, where \parallel denotes concatenation. This involves only one encryption to produce the ciphertext for n values. The photographer sends $\llbracket c_1 \rrbracket_{pk}, \dots, \llbracket c_m \rrbracket_{pk}$, the encrypted user ids (*uid*) of the discovered bystanders, and pk to the *Bystander Agent*. For each bystander, the *Bystander Agent* computes $\llbracket v_{b_j} c_j \rrbracket_{pk} = (\llbracket c_j \rrbracket_{pk})^{v_{b_j}}$ for $1 \leq j \leq m$, where v_b is the classifier weight vector of a bystander. Multiplying these encrypted values, the *Bystander Agent* obtains a packed encryption of the dot products, $\llbracket P_1 \parallel \dots \parallel P_n \rrbracket_{pk} = \llbracket V_1 \cdot v_b \parallel V_2 \cdot v_b \parallel \dots \parallel V_n \cdot v_b \rrbracket_{pk} = \llbracket v_{b_1} c_1 \rrbracket_{pk} \llbracket v_{b_2} c_2 \rrbracket_{pk} \dots \llbracket v_{b_m} c_m \rrbracket_{pk}$ and sends it back

to the photographer, who decrypts (using sk) and unpacks the values to recover the individual dot products.

Garbled circuits for secure threshold computation: Garbled circuits allow two parties holding inputs x and y , respectively, to evaluate an arbitrary function $f(x,y)$ without disclosing their inputs. The basic idea is that one party (the garbled circuit generator—the *Capture Agent* in our setting), prepares an “encrypted” version of a boolean circuit computing f ; the second party (the circuit evaluator—the *Bystander Agent* in our case) then obviously computes the output of the circuit. The combination of *secure dot product* and *garbled circuits* can provide the property that the bystander’s session key is revealed to the capture device if, and only if, there is a match between an extracted feature vector and the classifier weight vector of a bystander. The capture device can then decrypt the bystander’s policy.

3.5.3 Secure matching protocol

Figure 3.6 shows a high level diagram of I-Pic’s secure matching using the two cloud agents.

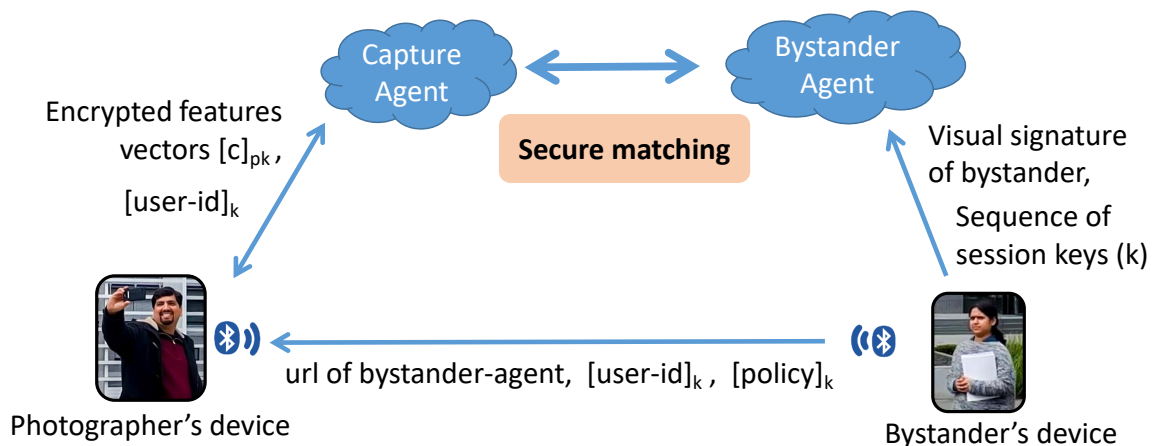


Figure 3.6. High level diagram of I-Pic secure matching with the two cloud agents

An example message exchange of the secure matching protocol for one image with n detected faces and b bystanders is shown in Figure 3.7. The photographer’s device computes the m encrypted column vectors according to the “optimized $n \times 1$ ” secure dot product protocol, which requires m encryptions. The device sends these vectors to the *Bystander Agent* (via the *Capture Agent*) along with the encrypted user ids of the b bystanders (Message 2 and 3 in Figure 3.7).

The I-Pic *Bystander Agent*¹ now looks up the classifier weight vectors of the b bystanders. For each bystander, it computes the encrypted packed dot products,

¹To simplify exposition, the description here assumes a single *Bystander Agent* service. The capture device would have to execute the protocol for each *Bystander Agent* in case more than one is discovered.

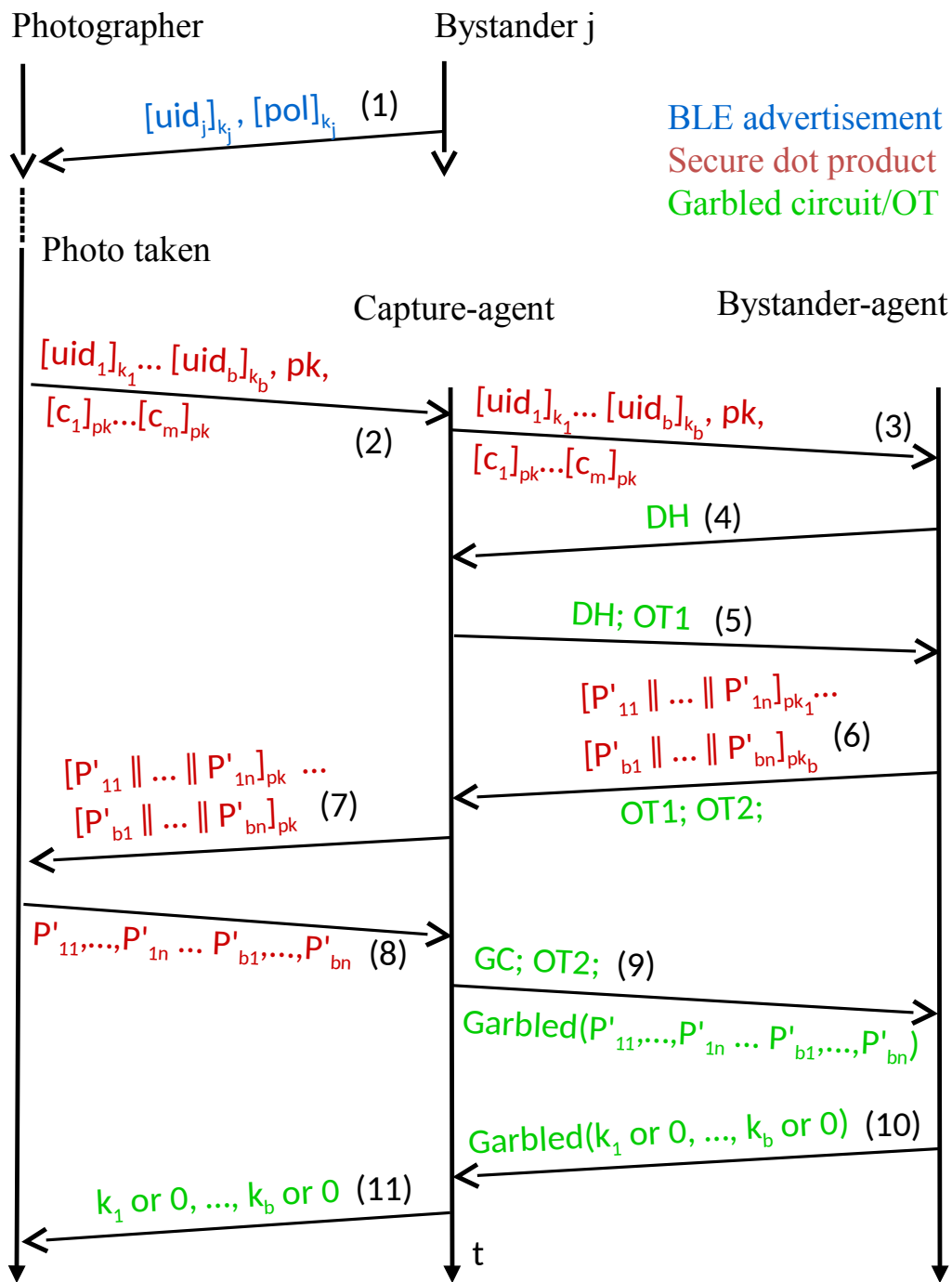


Figure 3.7. I-Pic secure matching protocol for one image with n faces (each facial feature vector has m dimensions). The photographer receives an advertisement from one of b bystanders (blue). The secure dot product computation requires one round trip (red). The garbled circuit (GC) requires a DH key exchange and two rounds of oblivious transfers (OT) (green).

$\llbracket P_{i,1} \parallel P_{i,2} \parallel \dots \parallel P_{i,n} \rrbracket_{pk}, 1 \leq i \leq b$, of the bystander feature vector and the n image feature vectors.

Secure thresholding The *Bystander Agent* computes *obscured* encrypted packed dot products, $\llbracket P'_{i,1} \parallel P'_{i,2} \parallel \dots \parallel P'_{i,n} \rrbracket, 1 \leq i \leq b$, by adding a different random value $R_{i,j}$ to each dot product $P_{i,j}$, for $1 \leq i \leq b, 1 \leq j \leq n$. This is performed by multiplying each of the b packed encrypted values containing n dot products each, $\llbracket P_{i,1} \parallel P_{i,2} \parallel \dots \parallel P_{i,n} \rrbracket_{pk}$, with $\llbracket R_{i,1} \parallel R_{i,2} \parallel \dots \parallel R_{i,n} \rrbracket_{pk}$ for $1 \leq i \leq b$. These *obscured* encrypted packed dot products are sent to the photographer's device via the *Capture Agent* (Message 6 and 7).

The photographer's device decrypts the b packed encrypted values containing n *obscured* dot products each, which requires b decryption operations. The device forwards these *obscured* dot products to the *Capture Agent* (Message 8), which then constructs a garbled circuit that takes as input n *obscured* dot products $P'_{i,j} = P_{i,j} + R_{i,j}$, n random values $R_{i,j}$, a session key K_i , and the threshold \mathcal{E} (all provided by the *Bystander Agent*), for $1 \leq i \leq b, 1 \leq j \leq n$. The circuit computes

$$f(P'_{i,j}, \mathcal{E}, R_{i,j}, K_i) = \begin{cases} K_i & \text{if } P'_{i,j} > \mathcal{E} + R_{i,j} \\ 0 & \text{Otherwise} \end{cases}$$

that is, the circuit reveals a bystander's session key iff the dot product of the bystander's classifier weight vector and an image feature vector exceed the threshold.

Delivering the *Bystander Agent*'s inputs to the garbled circuit requires a Diffie-Hellman key exchange (DH) and two rounds of oblivious transfers (NPOT [67] and OTEXT [15]), which are partly piggy-backed on the secure dot product protocol messages, and shown in Figure 3.7 (Messages 4, 5, 6 and 9). The *Capture Agent* now sends the circuit to the *Bystander Agent*, along with the garbled values of the obfuscated inputs $P'_{i,j}$, and the garbled values of *Bystander Agent*'s inputs as part of the OTEXT oblivious transfer (Message 9).

The steps of OTEXT algorithm are described in Figure 2.5. In Figure 3.7 the steps labeled 'OT2' (steps 6 and 9) refer to completing the overall OT_ℓ^m described in Figure 2.5. The steps labeled 'OT1' (steps 5 and 6 in Figure 3.7) correspond to completing the OT_m^k primitive (step in 2 in Figure 2.5). The OT_m^k primitive is implemented as k invocations of the *1-out-of-2 OT* protocol, described in Figure 2.1. These k invocations are clubbed into steps 4, 5, and 6 in Figure 3.7.

The *Bystander Agent* executes the circuit b times with the appropriate inputs, and returns the garbled results to the *Capture Agent* (Message 10). After ungarbling the results, the *Capture Agent* returns the session keys for the matched bystanders to the photographer's device (Message 11).

As composed, the matching protocol has the desired property that a photographer learns a bystander’s current session key if and only if a feature vector in the image matches that bystander’s classifier weight vector. Garbled circuits also ensure that the *Bystander Agent* does not learn whether there was a match between the encrypted facial feature vectors and a bystander. Additionally, no principal learns the vectors held by the other principals nor the magnitude of the dot products.

Note that the *Capture Agent* is trusted to construct the garbled circuit correctly. This requirement could be relaxed if one is willing to run additional checks [68] at some additional computational and runtime overhead.

3.6 I-Pic Evaluation

We have prototyped I-Pic on Android version 4.4.2.² In our deployment, we used a Google Project Tango Tablet [69] as the photographer’s capture device and Galaxy Nexus³ phones as bystander devices. The Nexus phones advertised their presence once every 640ms over BLE.

We ported HeadHunter [18] to Android for face detection. HeadHunter is optimized for execution on CUDA-enabled GPUs [70]; the Tango Tablet allows us to access CUDA cores. The camera output on the tablet (available as a JPEG file) is first histogram equalized [71] and then resized to 640x360 before being input to HeadHunter. HeadHunter outputs bounding boxes corresponding to detected faces.

To extract feature vectors from facial images, we used an Android port of the Caffe framework [72] and ran it with our FNet neural network. The extracted vectors were normalised such that each feature value was in the range $[0, 1]$. We ported existing Java secure dot product and garbled circuit implementations [73] to C++ on Android to optimize for runtime and energy consumption. The various agents were implemented as HTTP servers.

We begin with a description of I-Pic deployments in various settings; these deployments were also approved by the University of Maryland IRB. While we gained intuition about our vision pipeline using standard face recognition datasets (and the pipeline’s performance compares well with the state-of-the-art on them), all results presented here evaluate I-Pic on images captured “in the wild”, reflecting spontaneous image capture in different social situations with a range of lighting conditions, camera angles, distances, and poses.

²The code for the I-Pic prototype is available at <https://ipic.mpi-sws.org/code/ipic.html>

³Galaxy Nexus has Bluetooth hardware capable of BLE advertising, but the functionality is not available via standard API calls. We patched the kernel to enable BLE advertising.

3.6.1 Deployments

To evaluate I-Pic, we registered fifteen volunteers from our institutions using the registration procedure detailed in Section 3.5.1. Each volunteer received a Galaxy Nexus device for BLE advertisement, which they carried on their person. Registered users could choose to either *show* or *blur* their face when photographed; this setting could be changed at their discretion.

The photographs in our results were captured over three days (see Table 3.3), and were taken using the Tango tablet and a DSLR camera. We used the DSLR setup (Sony A7, 35mm f/2.8 lens, 1/80 fixed exposure time with Sony HVL-F32M flash) to simulate better tablet cameras with higher resolution and faster apertures expected in future tablets. The photographs captured by the DSLR camera were manually fed into the I-Pic processing pipeline.

We annotated all photographs manually with ground truth face rectangles using the open source annotation tool Sloth [74]. For each face, we manually added other information, such as the identity of registered users, pose, and lighting condition.

Date	Capture device	Number of photographs	Number of ground-truth faces
Nov 20	Tango tablet	81	277
Nov 27	Tango tablet	176	553
Dec 02	DSLR	130	843
	All	387	1673

Table 3.3. Experimental dataset

3.6.2 I-Pic decision tree

In I-Pic, faces in photographs end up being edited (e.g., blurred) or remain unchanged, correctly or incorrectly, depending on decisions made by different subsystems. Figure 3.8 shows the possible paths through I-Pic, culminating in leaf nodes colored green if I-Pic preserves user privacy and red if it does not. Note that it is possible for I-Pic to make a mistake, e.g., not recognize a face, and for the corresponding path to still lead to a green leaf node, e.g., because the user policy stated not to obscure their face. Finally, some leaf nodes are grey, corresponding to privacy irrelevant mistakes where non-faces were detected as faces and possibly blurred.

Understanding this decision tree, and in particular, analyzing where privacy-relevant errors can accrue, will enable us to parameterize and evaluate our vision pipeline in the context of I-Pic’s overall goal.

The decision tree has three stages: (1) face detection, (2) face recognition and (3) policy application. Stages 1 and 2 are computational and depend solely on the accuracy of the vision pipeline. The diagram separates these from Stage 3, which is contingent

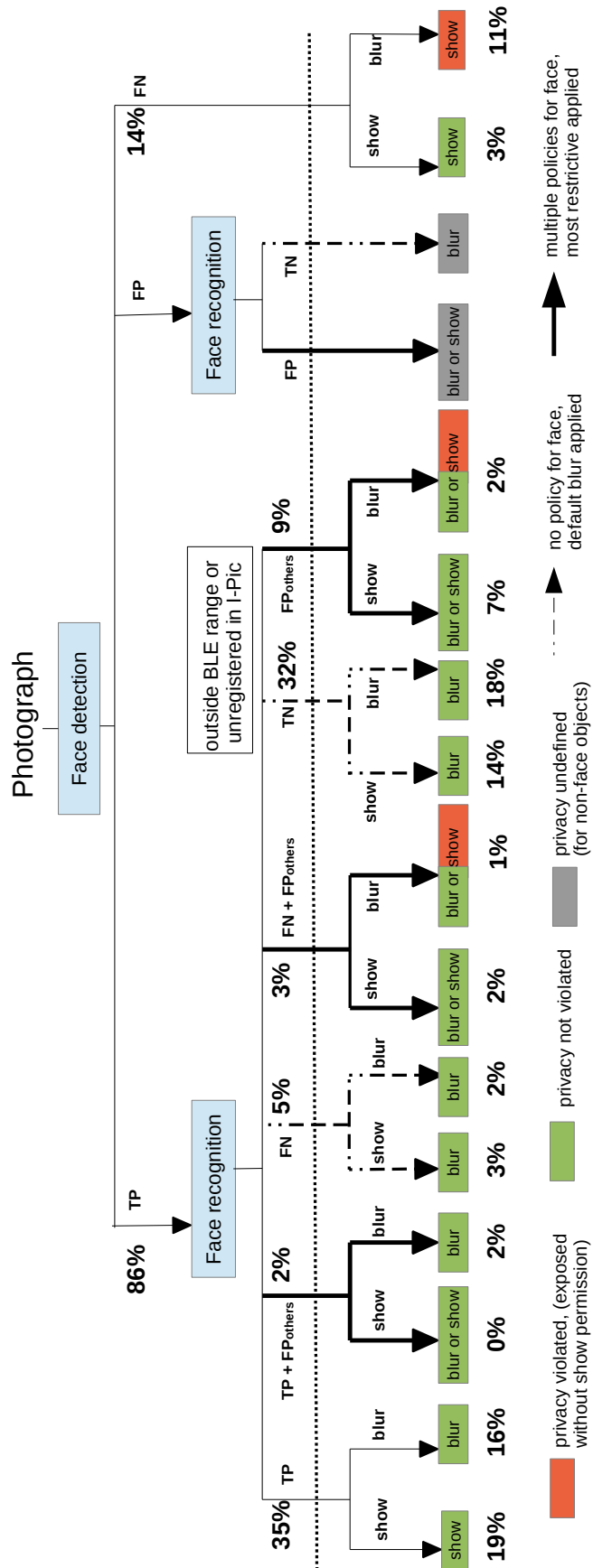


Figure 3.8. I-Pic decision tree

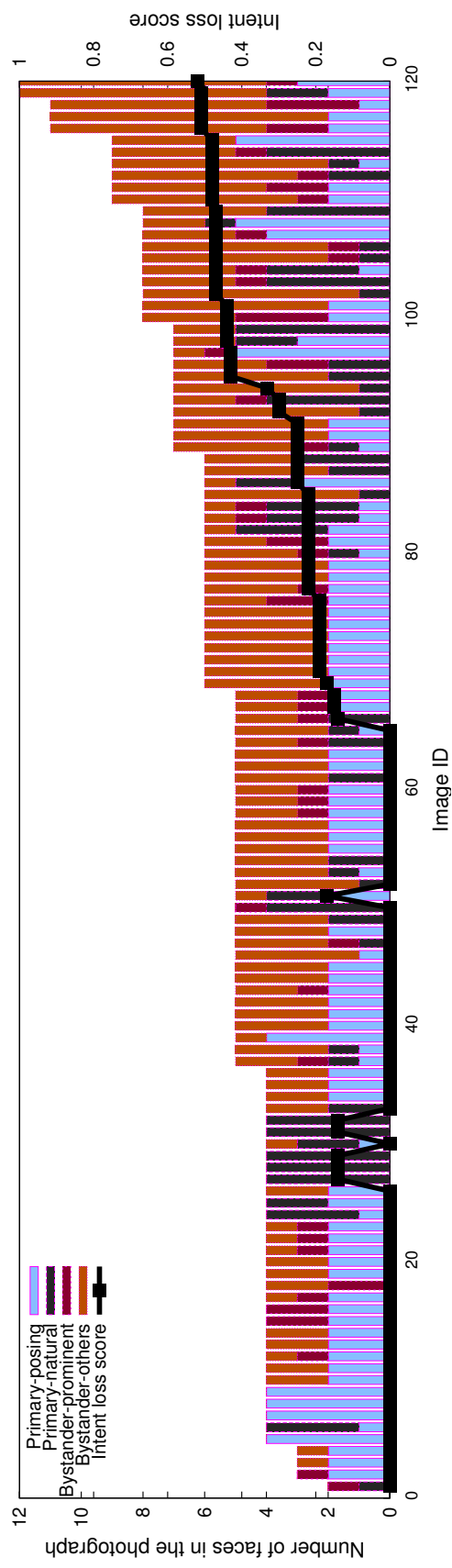


Figure 3.9. I-Pic Intent score of images

on user choices. For instance, if users choose more permissive policies, then errors from previous stages will less likely result in privacy violations, and vice-versa.

Face Detection: Stage 1 may result in three outcomes: True Positive (*TP*), where I-Pic detects a face marked in ground truth; False Positive (*FP*), where I-Pic detects a non-face object as a face; or False Negative (*FN*), where I-Pic does not detect a face marked by ground truth. All *TP* and *FP* detections are passed to the face recognition engine in the next stage.

The *FN* faces bypass the I-Pic pipeline and remain unchanged, and can potentially lead to a privacy violation (red leaf node). To minimize these cases, we bias the face detection engine towards higher recall (lower *FN*) at the expense of lower precision (higher *FP*). This means that a non-face object occasionally gets blurred in an image, in exchange for increased privacy.

Face Recognition: For a *TP* face detection output, there are six possible choices for recognition in the I-Pic pipeline: (1) True Positive (*TP*), where the detected face is matched only with the individual identified in ground truth; (2) True Positive along with False Positives (*TP**), where the face is matched with the ground truth individual, but also with others⁴; (3) False Negative (*FN*), where the face is not matched with the ground truth person; (4) False Negative along with False Positives (*FN**): I-Pic does not match with the ground truth, but instead matches with one or more other registered individuals; (5) True Negative (*TN*), where I-Pic correctly does not match the face to any registered individual; and (6) False Positive(s) (*FP**), where I-Pic incorrectly matches the face to one or more registered users.

Two leaf nodes have privacy violations for face recognition. *FP* is responsible for both paths, while one of them also requires a *FN*. Thus lower *FP* or high precision has higher priority for recognition, and adequate balance with low *FN* or high recall is also necessary. These requirements guide the parameterization of the I-Pic face recognition engine.

Misdetected faces (*FP* in detection) are also fed into the recognition protocol, and may lead to (1) True Negatives (*TN*) whereby I-Pic does not recognize the “face” as a registered user, or (2) False Positives (*FP**) where I-Pic mistakenly matches the “face” to one or more registered users.

Policy: Each detected face leads to an action, as shown by the leaves of the tree. If the recognition engine outputs a single user, then the action corresponding to that users’ policy is undertaken. However, in cases of multiple matches, e.g., due to *TP**, *FN** or *FP**, the most restrictive policy chosen by any “recognized” user is applied. For all unrecognized users, I-Pic blurs faces by default.

⁴We allow multiple matches; any registered face that exceeds a similarity threshold is considered a match.

We will detail an experiment with 687 faces in 120 images to examine I-Pic’s privacy violations in Section 3.6.3. The percentages below the leaves in Figure 3.8 show the fraction of faces that mapped to each path in the decision tree, in this experiment. As can be seen from the percentage values, the privacy preferences of 14% of 687 captured faces were violated, primarily due to errors early in the vision pipeline (face detection). In the next sections, we will present detailed evaluations of the vision pipeline, whose accuracy primarily determines I-Pic’s performance.

3.6.3 I-Pic overall performance

We begin with an evaluation of I-Pic’s overall performance in terms of its primary goals, which are to (i) respect bystanders’ privacy, and to (ii) preserve the photographer’s intent to the extent allowed by subjects’ privacy choices.

Toward this end, we took a sample of 120 images with 687 faces marked in the ground-truth. We additionally marked each face according to its role in the image, as shown in Table 3.4, along with the frequency of faces with a given role.

Name	Role in photograph	Number of occurrences
PP	primary subject posing	185
PN	primary subject natural	115
BP	prominent bystander	56
BO	other bystanders	331

Table 3.4. Roles of faces captured in images

Many of the captured faces correspond to unregistered individuals. Since we don’t know the privacy preferences of these individuals, we assigned them policies manually, so that we can process each image as if each captured person were registered with a policy. We assigned the *show-face* policy to the 185 PP faces, since it would be inconsistent for a person who poses for a photograph to refuse to have their face shown. For the remaining 502 faces, we randomly choose one of *show-face* or *blur-face* policies.

The percentage values given at the leaves in Figure 3.8 show what fraction of these 687 faces had what outcome when run through the I-Pic system. As we can see, privacy was violated in 14% of the cases, while the remaining 86% had no privacy violation.

We also assign a privacy loss score in each case of violation. These scores provide a subjective measure of the severity of the privacy violation depending on the role of the face in the image, with higher scores indicating a more severe violation. The privacy loss scores are given in Table 3.5, with the last column indicating how many of each type of violation occurred in the 687 faces.

Privacy loss score	penalization scenario	occurrences
3	PN privacy violated	15 (2.18%)
2	BP privacy violated	12 (1.75%)
1	BO privacy violated	70 (10.19%)
0	no privacy violated	590 (85.88%)

Table 3.5. Privacy loss scores

About 2% of cases had the most severe privacy violation, which is to show a primary subject not posing for the camera against their wishes. Also about 2% of cases had a clearly visible bystander shown against their wishes, and around 10% were less severe cases, where a not prominently depicted bystander was not blurred. We conclude that, overall, I-Pic observes subjects' policies in most cases (86%). Moreover, violations that did occur were mostly in the moderate or mild category.

The second aspect of I-Pic's overall performance is its ability to preserve the photographer's intent, to the extent allowed by the subject's policies. Similar to the privacy loss score, we can define a subjective intent loss score, which penalizes blurring a posing primary subject (score 3), blurring a non-posing primary subject with a *show-face* policy (score 2), and bystanders with *show-face* policies (score 1) in decreasing order of severity. The ordering is based on a subjective judgment of intent loss severity when a face is unnecessarily blurred, based on the face's role in the image. We note that our assignment of an intent penalty for the bystander case is conservative, as it is unclear whether a photographer should have expectations about capturing bystanders.

Figure 3.9 shows the intent loss scores for the 120 images, normalized by the maximum intent loss that could occur in a given image. The images are sorted by increasing number of faces from left to right. The bars represent the image composition in terms of roles of the faces depicted in it. I-Pic preserves the photographer's intent, as measured by our score, perfectly in 55 (45.8%) of the images, with the intent loss increasing for pictures with more faces. The vast majority of intent loss cases are caused by a failure to recognize the face of a bystander with a permissive policy, combined with I-Pic's default policy to blur.

Being focused on privacy, I-Pic biases its choices towards privacy, including the default policy and the rule to apply the most restrictive policy in case of multiple matches. As a result, losses in the vision pipeline come at the expense of intent rather than privacy. In the following subsections, we investigate circumstances that lead to imperfections in the vision pipeline, which are causal for the losses in privacy and intent reported here.

3.6.4 Vision pipeline analysis

The I-Pic decision tree demonstrates how (and how many) privacy violations occur as a result of errors in the vision pipeline. An obvious case is when a face is not detected, and thus not blurred in post-process. We have identified and manually labeled images with factors that affect detection and analysis, as we explain next. This analysis is done with our full image dataset of 387 images, where 1673 faces have been manually marked with ground truth (Table 3.3).

Factors affecting detection and recognition

The factors labeled in the ground truth (lighting, pose, and size) greatly affect whether a face is detected or not. We determine size based on the number of pixels in the image the face occupies; “small” faces (**s-Sm**) have a bounding box with at least one dimension less than 100 pixels⁵; all other faces are “large” (**s-Lg**). Pose is one of “frontal, profile, tilted head” (**p-Std**); “facing up, down” (**p-Avert**); “back turned, obstructed view” (**p-Occ**). Lighting is one of “Bright, even lighting” (**l-Good**); “Low even lighting” (**l-Low**); “Backlit, Shadow, Strong directional” (**l-Poor**).

Figure 3.10 decomposes face detection recall along these factors, for our image dataset (Table 3.3). The figure includes example images corresponding to different conditions for visual reference. The recall values for detection can be as high as 95% to as low as 32%, based on lighting, face size, and how occluded a face is in a photograph.

The leftmost bar with recall around 32% represents all combinations of factors combined with a partly occluded pose (**p-Occ**). 20% of the faces in our dataset are in this category. Together with the faces that suffer from low or poor illumination and an averted pose (four leftmost bars), they have recall below 50%. Faces in this category are probably not clearly recognizable even for humans without contextual information.

Face characteristic	Recognition recall
l-Good-p-Std-s-Lg	85.22%
l-Good-p-Avert-s-Lg	82.79%
l-Low-p-Std-s-Lg	78.62%
l-Good-p-Avert-s-Sm	67.38%
l-Good-p-Std-s-Sm	66.29%
p-Occ or l-Poor	20.49%

Table 3.6. Face recognition recall vs. different illumination conditions, face poses and face sizes

Table 3.6 shows the face recognition recall for a subset of illumination, pose and size characteristics. Recognition recall is only meaningful for individuals who are registered

⁵The Tango camera produces 2688 x 1520 pixels images and Sony A7 produces 4240 x 2832 pixels images

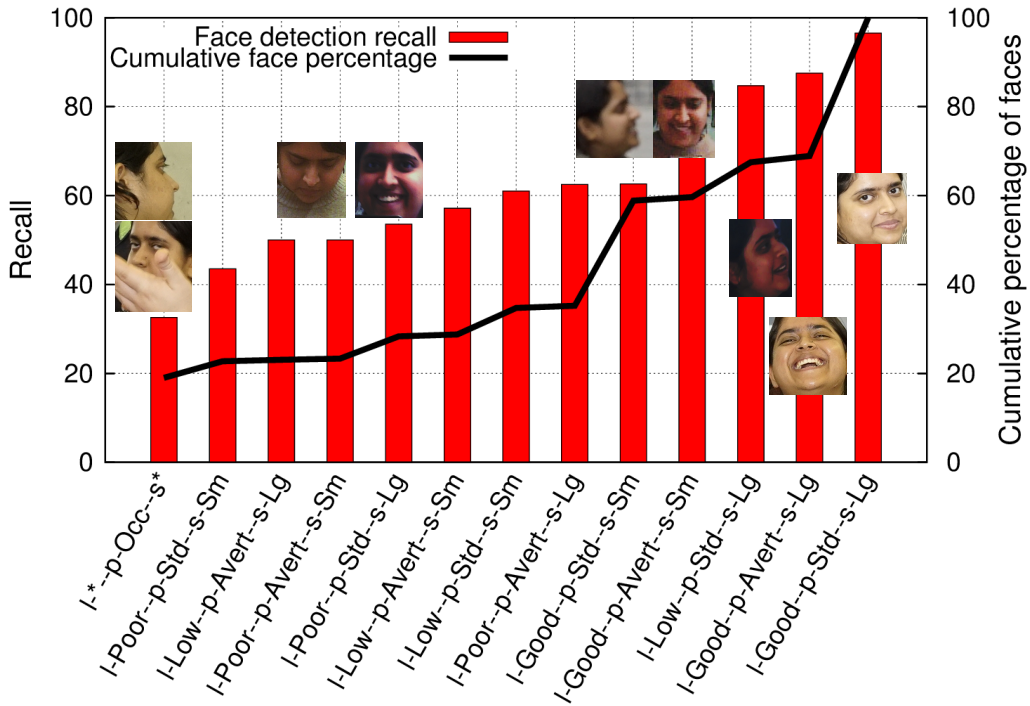


Figure 3.10. Face detection accuracy vs. illumination conditions, face poses and face sizes

in the I-Pic system. Our 15 registered individuals occurred with the subset of conditions given in Table 3.6, while only unregistered individuals occurred in other conditions.

p-Occ and **I-Poor** lead to poor recognition recall. This effect is intuitive, as occlusion or directional lighting distorts the facial features, making it harder to match with registered face models. Additionally, **s-Sm** performs worse than **s-Lg**. Our FNet neural network scales the input image to 227 x 227 pixels before feature extraction. Since **s-Sm** faces are less than 100 pixels in either width or height, this upscaling potentially affects the face recognition accuracy for small faces.

Precision for face detection or recognition do not show any marked correlation under different illumination, pose or size. In summary, good detection recall (>60%) and excellent recognition recall of nearly 80% occurs when pose is frontal or averted, illumination is good or low, or the size is large. This category includes about 65% of the faces in our images, and represents cases where subjects are clearly recognizable and privacy is most important.

Mapping back to events

The previous section identified different factors affecting I-Pic's face detection and recognition. But in what scenarios can one expect favorable conditions? In this section, we describe the scenarios in which we have evaluated I-Pic, and catalog photographs

and faces from each scenario according to our factors. We note that photographers were *not* aware of these factors when the photographs were taken.

Context name	characteristics	illumination
<i>Campus</i> (180 faces)	Individuals posing outdoors, with some bystanders present	Natural light
<i>Social</i> (237 faces)	Afternoon tea session with 40 people in an indoor atrium	Combination natural and fluorescent light
<i>Office</i> (129 faces)	Daily exchanges in offices and corridors	Fluorescent light
<i>Party</i> (424 faces)	Crowded party in small indoor venue	Back and directional lighting from lamps

Table 3.7. Four different social contexts

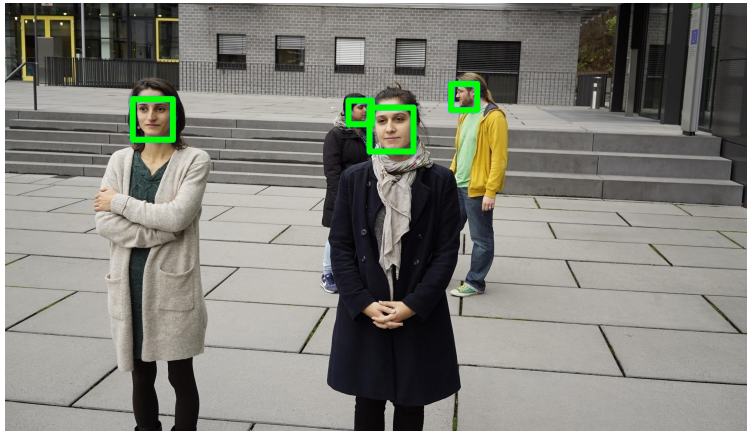


Figure 3.11. Campus Context

Table 3.7 lists four events where we obtained about 64% of our captured images. These images contain 970 manually annotated faces; the table lists the number of faces for each context. Figures 3.11, 3.12, 3.13, 3.14 shows representative images from each event; Figures 3.15, 3.16, 3.17, show the illumination, poses and size distribution for these 970 faces.

Figure 3.18 plots the recall ($\frac{TP}{TP+FN}$) and precision ($\frac{TP}{TP+FP}$) for both detection and recognition for the four events. The plot also includes data for *All*, corresponding to all 1673 faces in our evaluation, including those taken outside the four events.

Both detection and recognition recalls depend on contexts: *Campus* photographs taken outdoors with favorable poses have high recall for both detection and recognition. In contrast, challenging lighting and occluded faces in the indoor *Party* context lead to low recall.

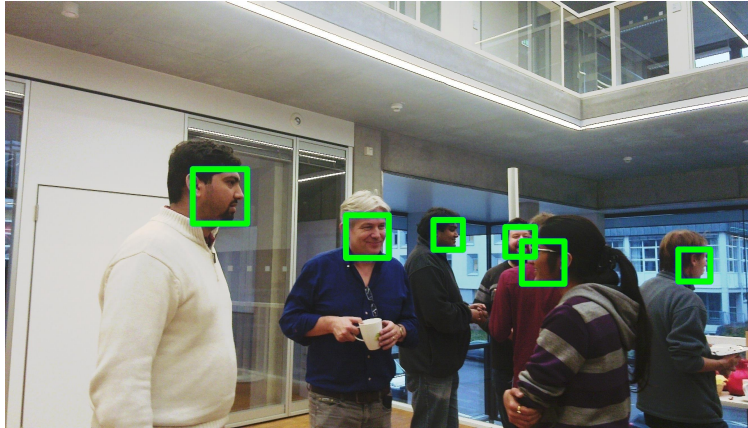


Figure 3.12. Social Context

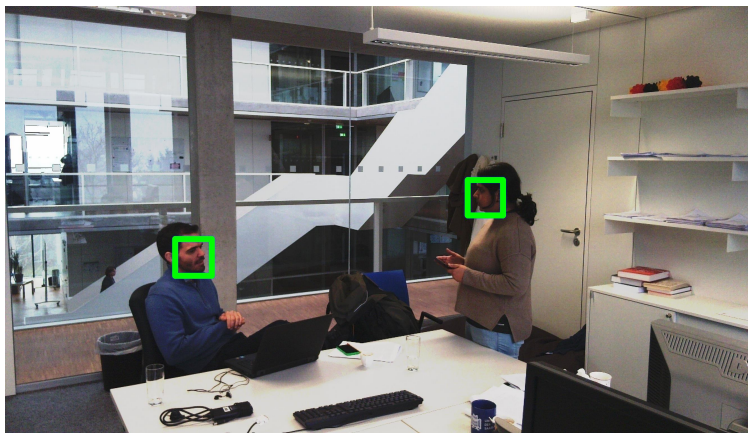


Figure 3.13. Office Context

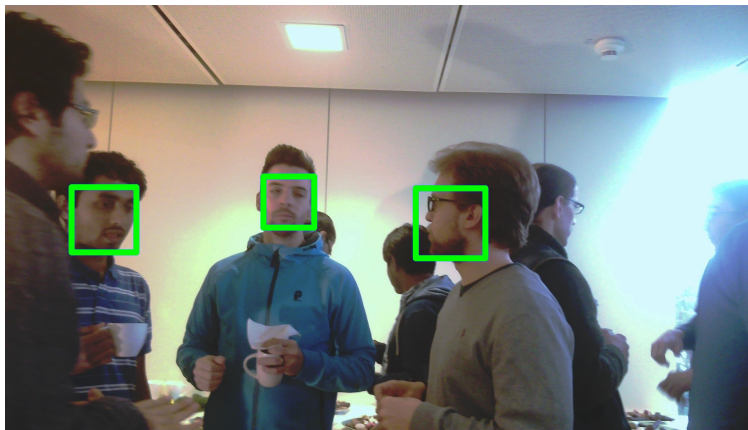


Figure 3.14. Party Context

Face recognition precision is high, independent of the social context. However, face detection precision varies with context. Manual inspection of the images revealed that busy scenes with many people have more false positives in face detection. Here, body parts like ears or hands, or striped clothing, accidentally match the face detection

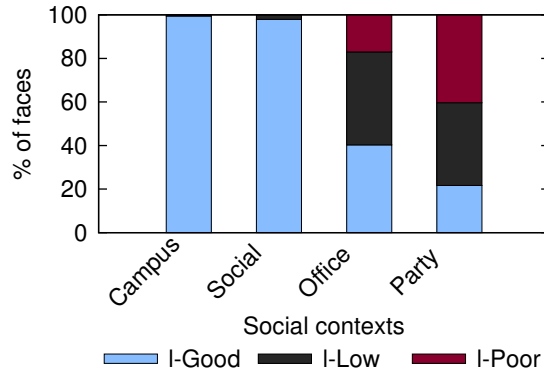


Figure 3.15. Variety in illumination conditions in different contexts

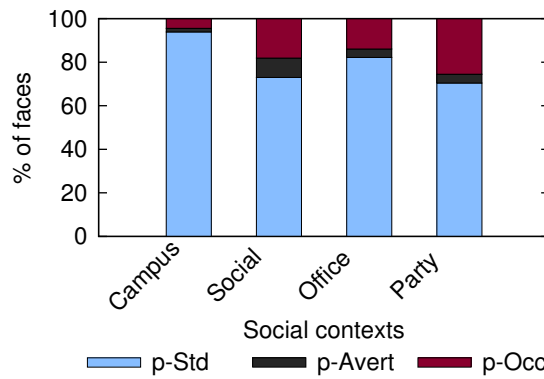


Figure 3.16. Variety in faces poses in different contexts

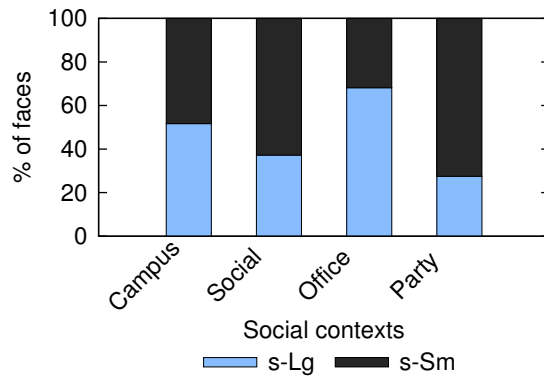


Figure 3.17. Variety in faces sizes in different contexts

template of HeadHunter. This shows up as lower precision in the *Social* context, which has crowded scenes.

As discussed in Section 3.6.2, I-Pic is biased towards higher recall for face detection and higher precision for face recognition, to maximize the privacy scores of the system. Figure 3.18 shows the effects of these choices on the vision pipeline performance.

In summary, the *Campus*, *Social*, and *Office* contexts have recall in the 70-80% range for both face detection and recognition. The challenging scenarios like *Party*

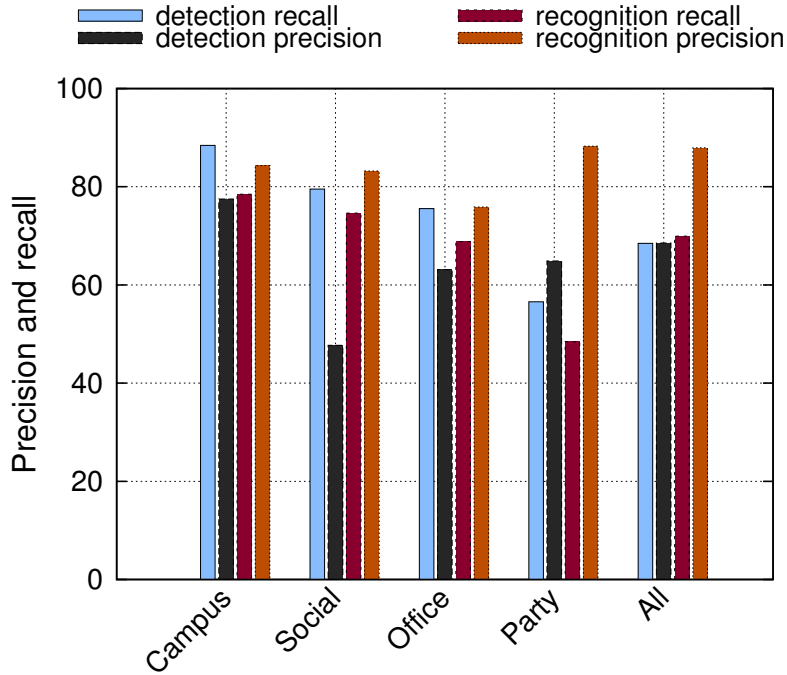


Figure 3.18. I-Pic vision pipeline performance

provide an opportunity for future vision research. Our image dataset, captured with mobile cameras, will be very useful to design new vision algorithms, which I-Pic can incorporate in the future.

Comparison to existing face detectors

We have used the open source HeadHunter [18] for face detection. A natural question to consider is how well existing, widely used face detectors, such as those bundled with Android or OpenCV, compare. Table 3.8 shows the precision and recall for different face detection libraries on our dataset. HeadHunter vastly outperforms the competition, justifying its use within I-Pic. Note that low detection recall, in particular, leads to false negatives in I-Pic, which can lead to privacy violations. In Chapter 4 (Section 4.1) we also evaluate the performance of a recent state-of-the-art head detector.

Library	Precision	Recall
Android	38.65	5.49
Snapdragon	94.28	5.91
OpenCV	31.27	49.91
HeadHunter	68.47	68.55

Table 3.8. Comparison of face detection libraries

3.6.5 Secure Feature Comparison

Next, we present microbenchmarks evaluating the processing and bandwidth requirements of the secure vector matching protocol with varying numbers of faces and bystanders. During these experiments, both the cloud agents are running on the same machine and are on the same 802.11 WiFi network as the I-Pic devices. In each run of the experiment, we generated feature vectors randomly.

Consider Figures 3.19 and 3.20, which show the protocol’s total runtime latency and its breakdown. Latency includes computations on the device, on the cloud agents, and the network transit time between the device and the *Capture Agent*.

The number of input vectors that have to be encrypted and transmitted increases with the number of faces in the photograph, resulting in an expected linear increase in runtime in Figure 3.19. Figure 3.20 shows that a major contribution to this runtime is the client side encryption of feature vectors for the secure dot product part of the protocol (Step 2 in Figure 3.7). Due to the “ $n \times 1$ dot product” optimization, described in Section 3.5.2, the client side runtime does not increase significantly with the number of faces.

From separate measurements (not shown in Figure 3.20) we know that these client side encryption operations show a 2x reduction in runtime on mobile platforms supporting a 64bit ARMv8-A instruction set.⁶

Increasing the number of bystanders for a fixed number of faces increases the runtime linearly, but importantly, it does not significantly increase the client-side runtime (Figure 3.20). This is a desirable property as the photographer’s overhead does not significantly depend on the number of bystanders in the vicinity.

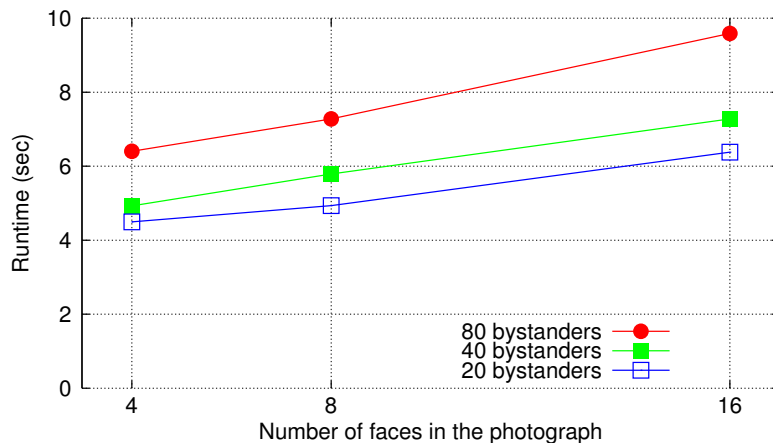


Figure 3.19. Total Runtime of the secure matching protocol

⁶Measurements are not shown here because the Tango tablet does not support the ARMv8-A instruction set.

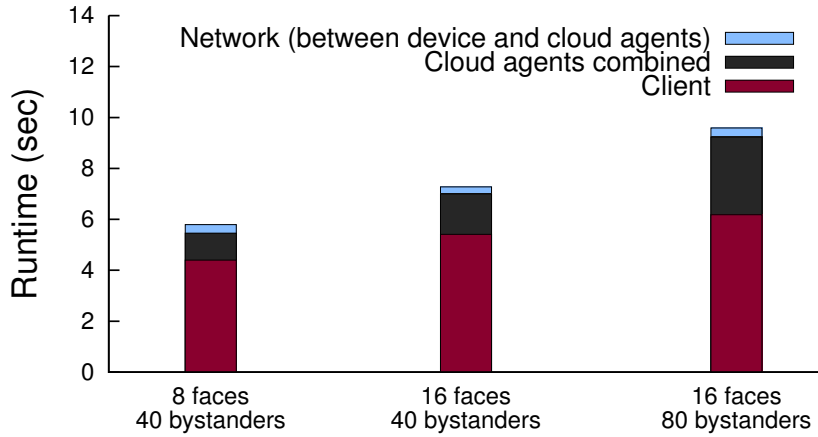


Figure 3.20. Runtime breakdown of secure matching protocol

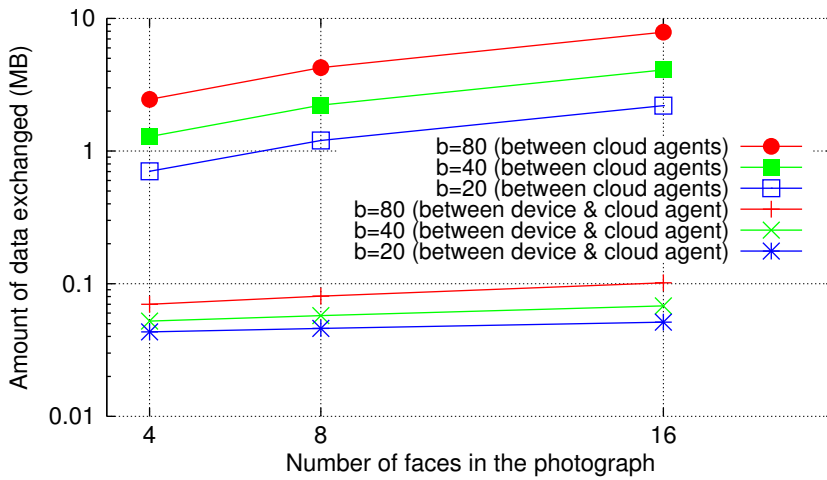


Figure 3.21. Total data exchanged for the secure matching protocol for different number of bystanders

Figure 3.21 shows the data transmitted between the device and *Capture Agent*, and between the cloud agents. We observe that data transmitted between the device and *Capture Agent* is less than 100KB and it does not increase significantly with the number of faces or bystanders. This figure also shows the effects of adding the garbled circuit. The garbled circuit affects the data exchanged (and the latency) between the cloud agents, which increases both with the number of bystanders and the number of faces. Garbled circuits are evaluated by the *Bystander Agent* for each bystander and the number of inputs to each garbled circuit depends on the number of faces.

Overall the results show that the secure matching protocol can be efficiently executed. Moreover, computation can be offloaded to a significant extent from the client devices to the cloud agents.

3.6.6 Runtime and Energy Consumption

Figure 3.22 plots the overall time taken for I-Pic to process different photographs, along with times spent in different vision and secure matching tasks. In each case,

the capture platform received and processed between 3 and 10 BLE advertisements, with varying number of faces in the photograph as plotted along the x -axis. The times for secure matching includes network communication and all cryptographic functions. Face detection dominates, often requiring 25 seconds per photograph. Recall that the processing takes place asynchronously in the background, and does not interfere with the users' experience while capturing and reviewing images.

While the face detection cost in particular is high in our prototype (70–80% of total processing time), we believe it is encouraging that best-of-breed face detection is feasible on mobile devices available today. Advances in mobile hardware capabilities, driven in part by emerging virtual reality applications, will benefit HeadHunter and other stages of the I-Pic pipeline in the near future. Moreover, face detection is already being offered as a standard feature on mobile platforms, and future implementations (possibly hardware supported) with better accuracy could directly benefit I-Pic.

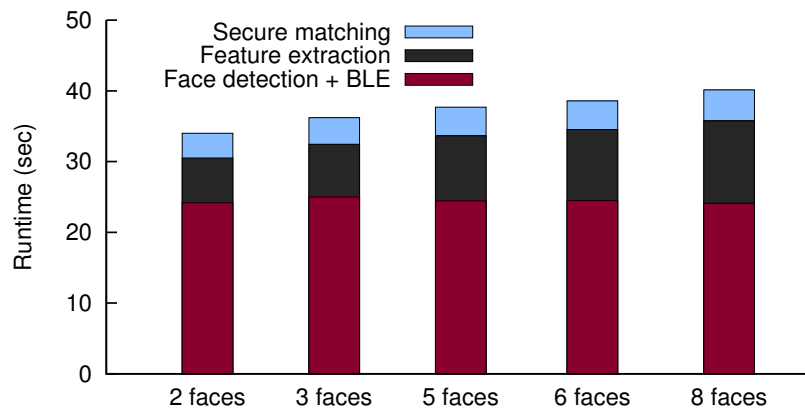


Figure 3.22. Overall and task level runtimes of I-Pic prototype. 10 bystanders were discovered in each case.

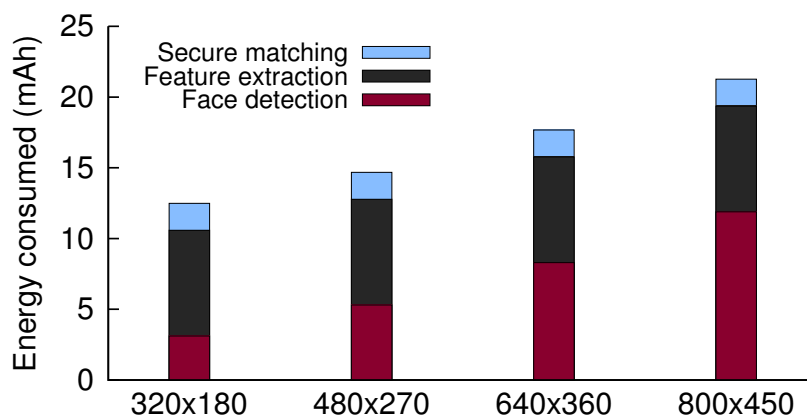


Figure 3.23. Energy consumption of I-Pic prototype for different image resolutions, 30 faces.

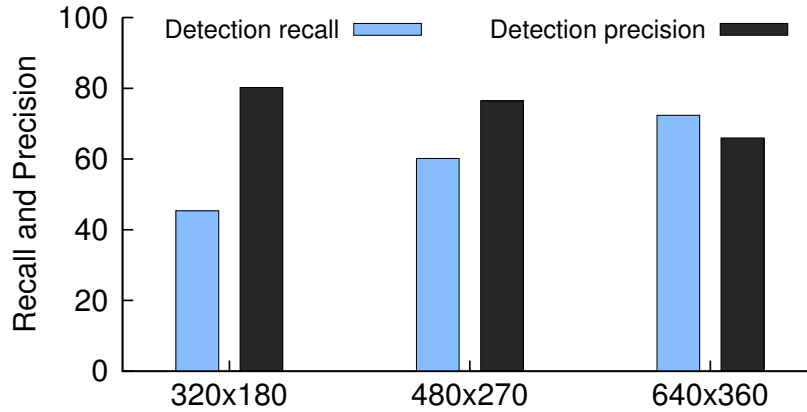


Figure 3.24. Face detection accuracy of I-Pic prototype for different image resolutions

We measured the energy consumption of the various subcomponents of I-Pic using the Monsoon Power Monitor [75]. We attached the power monitor to a Nvidia Shield Tablet K1 [76]⁷ and processed an image with 30 faces in it. Figure 3.23 shows the energy consumption for different resolutions of the input image. The face detector uses the GPU, whereas the feature extraction is CPU bound. Energy consumption of face detection is independent of the number of faces in an image, whereas it is linear in the number of faces for feature extraction. The secure matching algorithm was run with the 30 faces extracted from the image along with 40 simulated bystanders⁸.

Image resolution (pixels)	Number of images processed (containing 30 faces each)
320x180	408
480x270	347
640x360	288
800x450	239

Table 3.9. I-Pic's projected capacity on a 5100 mAh battery

Using these measurements, Table 3.9 shows I-Pic's projected capacity on the Nvidia Shield tablet, which has a 5100 mAh battery. More than 288 images and 8640 faces can be processed on a single charge. Figure 3.24 compares the face detection accuracy versus the resolution of input images, and serves to highlight the tradeoff between accuracy and energy consumption of the prototype. Reducing the resolution to 480x270 pixels enables the prototype to process 20% more images, but comes at a high (12%) drop in face detection recall. On the other hand increasing the resolution to 800x450

⁷We used the Shield tablet for the power measurements because the Monsoon power monitor is unable to power the Tango tablet. The latter requires a 7.5 volts power supply whereas the Monsoon power monitor can only supply a maximum of 4.5 volts.

⁸BLE scanning for 5 seconds consumes 0.12 mAh of energy, which is accounted for in Figure 3.23 but not shown separately.

only gives diminishing returns for face detection recall when compared to the increased energy consumption that accompanies it.

3.7 I-Pic Summary

I-Pic allows users to respect each others' individual and situational privacy preferences, without giving up the spontaneity, ubiquity, and flexibility of digital capture. The I-Pic design and prototype demonstrate that the technical impediments for privacy-compliant imaging can be reasonably overcome using current hardware platforms. I-Pic leverages cutting-edge face detection and recognition technology, which is often perceived as a threat to privacy, to instead increase user's privacy regarding digital capture. Future advances in mobile platform hardware and computer vision will directly benefit I-Pic to further improve the efficiency and accuracy of its privacy enforcement. In Chapter 4, we explore a state of the art head detection technique based on deep neural networks to improve I-Pic's privacy performance and also port I-Pic to a newer platform, a powerful development board containing the Nvidia Jetson TX2 Mobile SoC, to explore gains in energy efficiency possible using newer hardware.

Chapter 4

Exploring I-Pic’s performance limits

In this chapter we explore how I-Pic can benefit from recent advances in both software and hardware technology. With the initial I-Pic prototype presented in Chapter 3, the primary aim was to build a complete end-to-end system that worked on a mobile device, and to evaluate it in realistic social scenarios. In this chapter we explore how much better I-Pic can perform if we take advantage of state-of-the-art object detection techniques and powerful new hardware likely to be available in future mobile devices.

Specifically, we experiment with a state-of-the-art head detector [17] built on top of Faster R-CNN [19], a popular deep learning framework used to detect objects in still images. We also explore the feasibility of running this head detector on a mobile platform, using a powerful development board containing the new Nvidia Jetson TX2 Mobile SoC [77]. Overall we found that we can significantly improve I-Pic’s ability to protect a user’s privacy while consuming 33% less energy than our original prototype.

4.1 Exploring head detection

As described in Section 3.6.3, I-Pic’s privacy guarantees depend on the accuracy of its visual recognition sub-systems, which include the face detection and face recognition components. This dependency is highlighted in Figure 3.8, which shows all possible paths through I-Pic, culminating in leaf nodes colored green if I-Pic preserves user privacy and red if it does not. The figure shows that in 11% of cases I-Pic violates users’ privacy due to false negatives in our face detector, i.e., the detector fails to detect faces actually present in photographs. This loss in accuracy is because HeadHunter [18]), the face detector used in our prototype implementation, is unable to detect faces that are small, or faces that are partially occluded, or faces that are turned up or down, or rotated by more than 90 degrees in profile. Furthermore, HeadHunter frequently identifies hands and ears as faces, which further reduces the overall accuracy of I-Pic’s vision pipeline. Finally, HeadHunter does not use deep learning, which is now the de facto standard for building object detectors that offer better accuracy than

HeadHunter, such as Faster R-CNN[19]. Overcoming these limitations while exploring the limits of I-Pic’s accuracy was our primary motivation to experiment with a state-of-the-art head detector [17].

The head detector we used [17] was trained by our computer vision collaborators, and is based on the Faster R-CNN [19] framework. Additional details of this head detector along with some of the recent related work in object detection techniques are described in Section 2.2. In the following sections we will, first, briefly describe the additional post-processing steps that were performed on the output of the head detector (Section 4.1.1), and then we evaluate how well the head detector performs (Section 4.1.2) on the dataset of images generated in I-Pic’s earlier deployment.

4.1.1 Post-processing head detector output

The trained head detector was made available to us as two Caffe [36] model files, corresponding to the two stages(described in Section 2.2.2) of the detector, and Matlab code that executed the two models in succession to produce the output of the head detector.

The output of the head detector are bounding boxes in an image where a head might be present. Each bounding box also has an associated score that indicates the likelihood of a head being present in that box. We retained bounding boxes with score ≥ 85 and size $\geq 30 \times 30$ pixels. We will refer to these retained boxes as *output-boxes*.

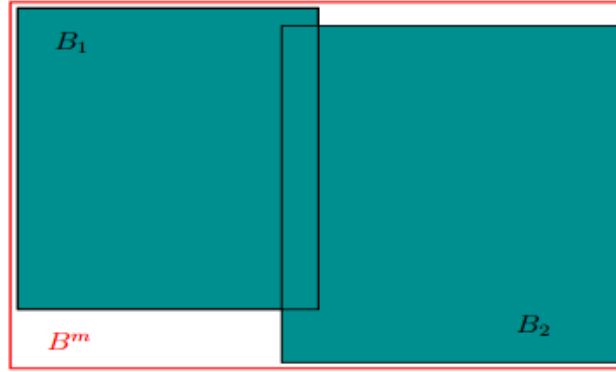
Box merging: We additionally post-process *output-boxes* by merging nearby boxes into a single larger bounding box, as show in Figure 4.1(a). We introduced this step to reduce multiple *output-boxes* detected for a single head. We merge a candidate box B_1 with a box B_2 into a larger box B^m , if 1) the relative area of intersection, $\frac{Area(B_1 \cap B_2)}{Area(B_1)}$, is ≥ 0.50 , and 2) the *Affinity* [78] between the two boxes, $\frac{Area(B_1 \cup B_2)}{Area(B^m)}$, is ≥ 0.95 .

Figures 4.1(b) and 4.1(c) show the effect of box merging on an image taken from I-Pic’s deployment. Figure 4.1(b) shows the multiple *output-boxes* (blue) without any merging applied. Figure 4.1(c) shows the output after box merging, where the merged box is shown in green. The ground truth annotation is shown in a yellow box in both figures.

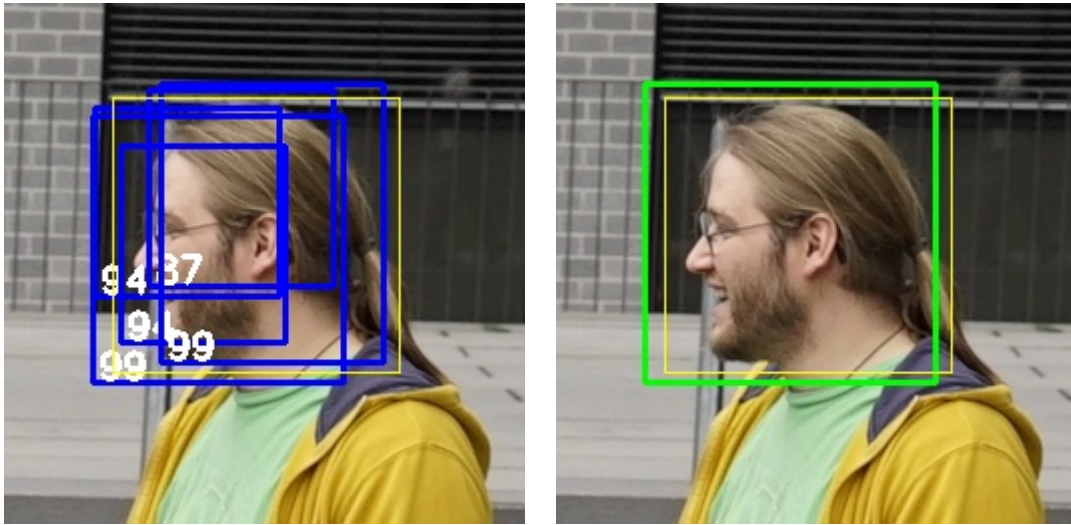
4.1.2 Performance of the head detector on the I-Pic dataset

In this section, we present a detailed evaluation of the head detector on the set of photographs collected in I-Pic’s initial deployment (Section 3.6.1).

As a first step, we manually annotated all photographs, replacing each face rectangle, previously annotated, with a full head rectangle. We additionally annotated each head with information, such as the identity of the person, pose, lighting condition, and the role it plays in an image. The head detector was executed on all the photographs using an Nvidia Tesla K80 GPU, followed by box merging to produce a set of *DetectedHeads*.



(a) Box merging



(b) Before merging

(c) After merging

Figure 4.1. Effect of box merging. Output-boxes (along with scores) are shown in blue. The box produced after merging output-boxes is shown in green. Yellow box shows the annotated ground truth.

For a particular ground truth box, the highest scoring *DetectedHead* with a relative area of intersection, $\frac{Area(DetectedHead \cap GroundTruthBox)}{Area(DetectedHead)} \geq 0.60$ was treated as a true positive (TP). All unmatched *DetectedHeads* and ground truth boxes were treated as false positives (FP) and false negatives (FN) respectively.

Head detector recall & precision: Table 4.1 shows the performance of the head detector compared to HeadHunter and other face detection libraries. There is a significant increase in recall, $\frac{TP}{(TP+FN)}$, as compared to HeadHunter, which shows that the head detector is able to detect many more heads than HeadHunter. This is a highly desirable property to have in order to improve I-Pic's ability to provide privacy. Figure 4.2 shows all the false negatives (FN) of the head detector. We find that the head detector sometimes misses out on heads that have significant occlusions, or heads that are very small in size, or heads that are backlit. Somewhat surprisingly,

but very rarely, it fails to detect frontal heads. This is a limitation of the detector that our vision collaborators also acknowledge, and will require re-training the head detector. Nevertheless, the performance of the head detector is significantly better than HeadHunter.



Figure 4.2. False negatives of the head detector.

We also manually inspected the false positives in *DetectedHeads* and found that very few non-head objects were falsely identified as heads. About 86% of all the false positive *DetectedHeads* corresponded to actual heads. 75% of these were the backs

of heads which were not marked in our ground truth dataset, and the rest were either profile or front heads corresponding to duplicate detection boxes still remaining after box merging.

Library	Precision	Recall
Android	38	5
Snapdragon	94	6
OpenCV	31	49
HeadHunter	68	68
Head detector	76	92

Table 4.1. Comparison of the head detector with other face detection libraries

I-Pic’s processing pipeline with head detector: I-Pic’s updated processing pipeline with the head detector included is shown in Figure 4.3. Unlike I-Pic’s earlier pipeline (presented in Figure 3.8), the updated pipeline processes each photograph using, both, HeadHunter (to detect faces) and the head detector (to detect heads). The output of this step is classified in three categories and is processed accordingly.

Matched faces: These are all faces (detected by HeadHunter) that are also detected as heads by the head detector. They serve as input to the face recognition step. Depending on the privacy preferences of the bystanders and the accuracy of face recognition, some of the *matched faces* are selected for obfuscation. *DetectedHeads* corresponding to these *matched faces* are obfuscated. We obfuscate the head instead of the face because obfuscating the head removes more identifying information about a person than just obfuscating a face. Therefore to provide better privacy and since the corresponding matched head bounding box is readily available, we choose to obfuscate the head.

Unmatched heads: These are heads detected by the head detector (*DetectedHeads*) that are not detected as faces by HeadHunter. These heads are obfuscated by default. By doing so, we err on the side of privacy in case of a bystander who either does not carry a mobile device, or does not use I-Pic, or whose BLE broadcasts were not received, or whose face was missed due to a false negative in face detection. The default policy of obfuscation is also in accordance with the one we had used for our initial I-Pic prototype, described previously in Section 3.5.

Unmatched faces: These are faces detected by HeadHunter that are not detected as heads by the head detector. These faces are not processed further. Through manual inspection we found that a large majority of these *Unmatched faces* were non-face objects, such as hands and ears. Therefore we chose not to obfuscate these by default.

To study the performance of this updated pipeline, in terms of its ability to protect users’ privacy and photographer’s intent, we conducted an experiment, similar to one described in Section 3.6.3, using a sample of 1545 ground truth faces/heads from 380 images. Table 4.2 and Figures 4.3 and 4.4 show the result of this experiment.

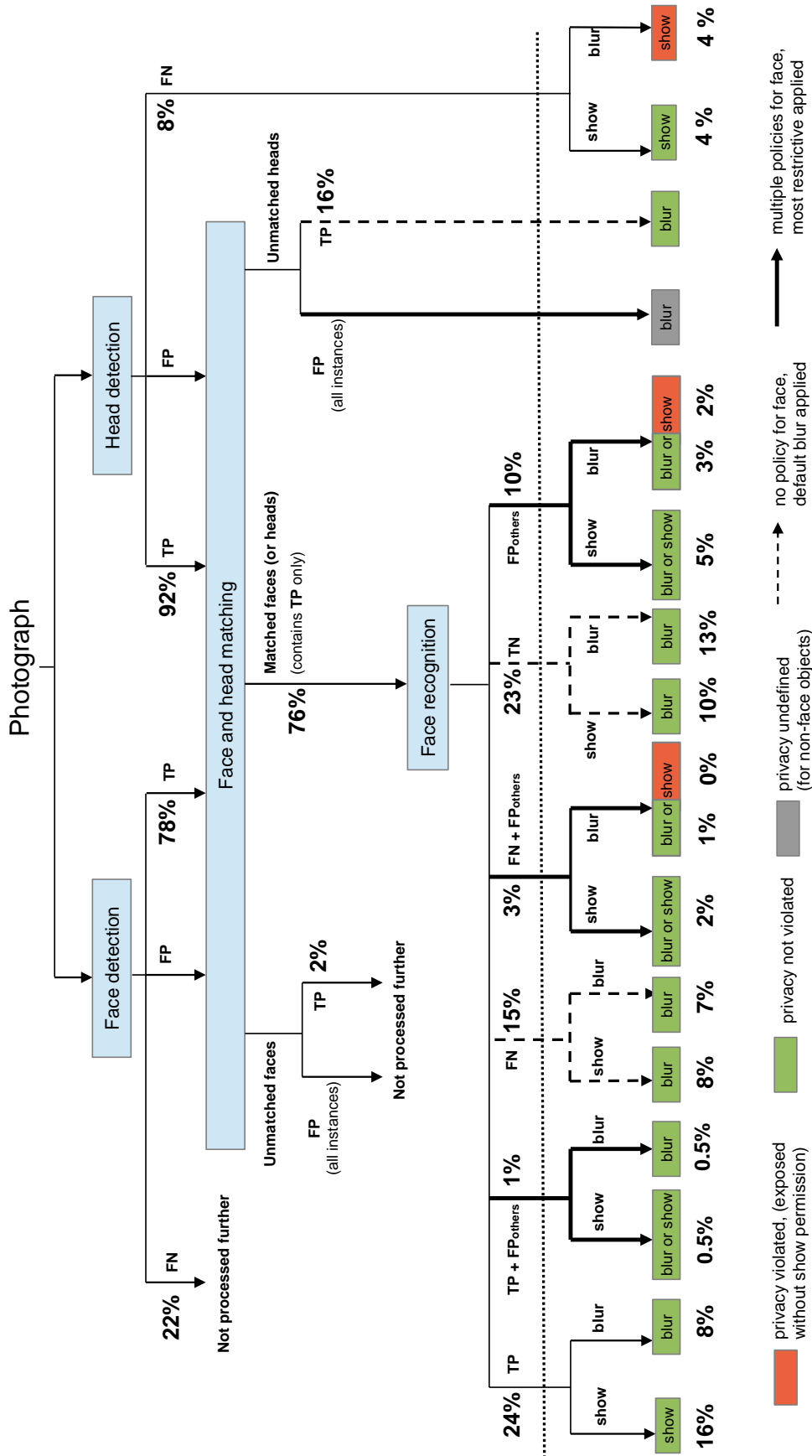


Figure 4.3. I-Pic decision tree with head and face detection

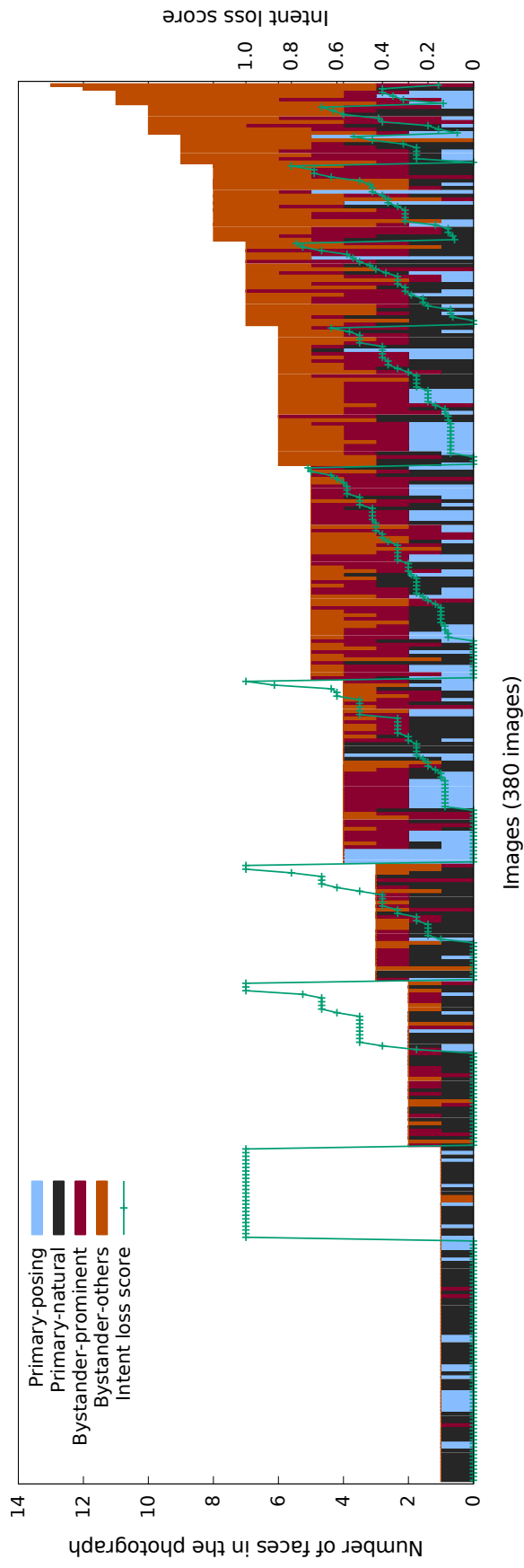


Figure 4.4. I-Pic Intent score of images with face recognition on matched faces

Role in photograph	Number of occurrences	Privacy violations	Intent violations
Primary subject posing	221 (14.3%)	0 (0%)	48 (3.1%)
Primary subject natural	428 (27.7%)	6 (0.4%)	111 (7.2%)
Prominent bystander	425 (27.5%)	24 (1.5%)	160 (10.3%)
Other bystanders	471 (30.5%)	65 (4.2%)	159 (10.3%)
Total	1545	95 (6.1%)	478 (30.9%)

Table 4.2. Number of instances of privacy & intent violations, categorized by their role in photographs

Protecting privacy: From Figure 4.3 we find that including the head detector significantly improves I-Pic’s ability to protect users privacy. This is shown by the percentage of heads for which privacy is violated, which has been reduced from 14% of faces previously (in Figure 3.8) to 6% of faces. This significant improvement is a direct result of the head detector’s ability to detect nearly all heads in an image (high recall). Furthermore we found (through manual inspection) that all non-face objects (hands, ears, etc.) detected by HeadHunter as faces were discarded as *unmatched faces*. This is because the head detector’s output (*DetectedHeads*) contains very few non-face objects, therefore the intersection of HeadHunter’s output and *DetectedHeads* retains actuals heads only, and filters away non-head objects as *unmatched faces*. This ability to automatically discard non-face false positives is highly desirable, since these objects don’t match any of the bystanders during face recognition, and are unnecessarily obfuscated.

Intent preservation: Figure 4.4 shows how using the new head detector impacts I-Pic’s ability to preserve the photographer’s intent. We find that 36% of all images show no intent violations, as compared to 46% images previously (in Figure 3.9). This decrease in performance is primarily because all *unmatched heads* are obfuscated by default, decreasing I-Pic’s overall intent score performance. This also highlights the inherent trade off between privacy and photographer’s intent. In I-Pic we have deliberately chosen to err on the side of privacy. Moreover, from Table 4.2, we see that the majority of intent violation instances occurred for bystanders in the background, who generally appear smaller in size than primary subjects in an image. This is encouraging because, even though intent violations are not desirable in general, overall it is less severe to incorrectly blur out small heads in the background, rather than prominent heads in the foreground.

Accuracy in different social scenarios: Figure 4.5 shows the recall & precision of the head detection and face recognition components, categorized by the social scenarios presented in Section 3.6.4. We find that in all scenarios, recall for the head detector is more than 90%, which is a significant increase compared to Figure 3.18 previously, where it ranged from 60 to 80%. Head detection precision also shows an improvement

of 5 to 10 percentage points in all the scenarios. Face recognition results are nearly similar to the previous results, which is expected.

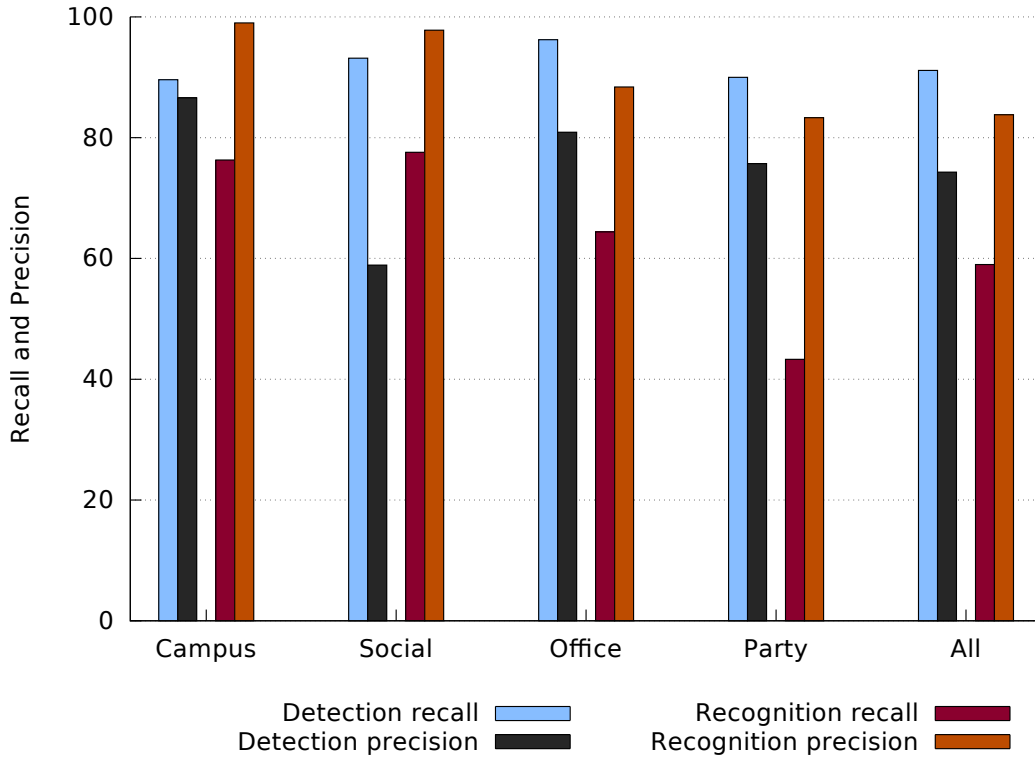


Figure 4.5. Recall and Precision for Head detection and Face recognition(matched faces)

Head recognition: In the discussion above we used face recognition to match privacy preference of captured bystanders to their visual signatures. Here, however, we show how the performance is impacted if use head recognition instead. To do that we followed steps similar to the ones described in Section 3.5.1. For each registered user, we trained a head recognition SVM classifier using the training images previously collected and 3000 heads images from PASCAL VOC 2010 training set [35] (as the generic negative set). Features vectors from head images were extracted using a deepnet similar to FNet, which was trained on head images instead of face images.

We evaluated the head classifiers both on *DetectedHeads* and *matched heads*. Overall the results were not very promising: the recall and precision for head recognition was less than 30% in most scenarios. It turns out that state-of-the-art systems for head recognition, such as [55] and [79], also have low accuracies, 46% and 60% respectively. One reason for I-Pic’s even lower performance (less than 46%) is that we use a generic negative set to train person-specific classifiers. This makes I-Pic’s head recognition use case even more challenging than the most difficult scenario (the *day-split* configuration) evaluated in state-of-the-art systems [55, 79]. Although not a viable option for I-

Pic, training one-vs-all classifiers similar to the ones described in [55] brought I-Pic’s accuracies closer to the accuracies reported in [55].

Overall we found that accurate head recognition with arbitrary poses that works well in a wide range of social situations, and also in an open world setting (as required by I-Pic) is still an open research challenge. Substantially improving I-Pic’s head recognition performance will require significant improvements in state-of-the-art technology for head recognition, which is currently beyond the scope of this thesis.

4.2 Exploring I-Pic’s runtime performance on a new Mobile SoC

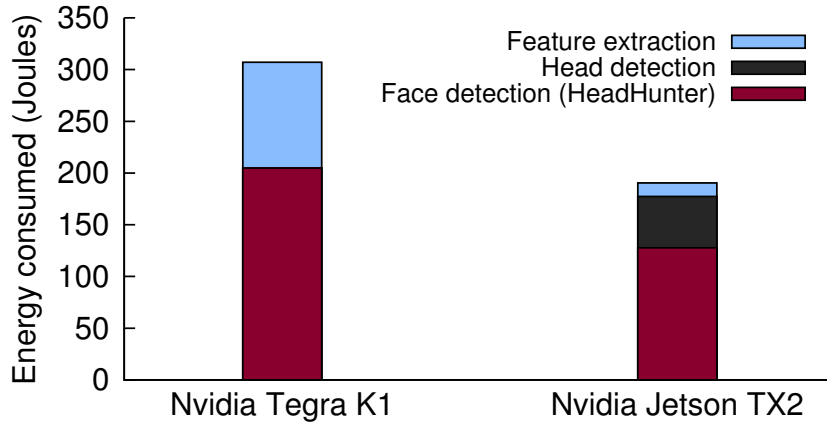
In this section we explore how I-Pic can benefit from powerful new hardware likely to be available in future mobile devices. Previously, in Section 3.6.6, we had described the energy consumption of I-Pic on the Nvidia Shield Tablet containing the Nvidia Tegra K1 Mobile SoC, released in 2014. Here we explore how much better I-Pic can perform using a powerful new Mobile SoC, the Nvidia Jetson TX2 [77], released in 2017.

To measure I-Pic’s runtime performance, we ported the three main components of I-Pic’s new processing pipeline, the head detector, face detector, and the feature extractor, on to the Nvidia Jetson TX2 Developer Kit [80]. The developer kit runs the ARM port of the standard Ubuntu 16.04 Linux distribution and provides display, Ethernet, & USB ports to conveniently access the Jetson TX2 module. We also ported the Matlab code accompanying the head detector (described in Section 2.2.2) to python, since Matlab is not supported on the ARM platform.

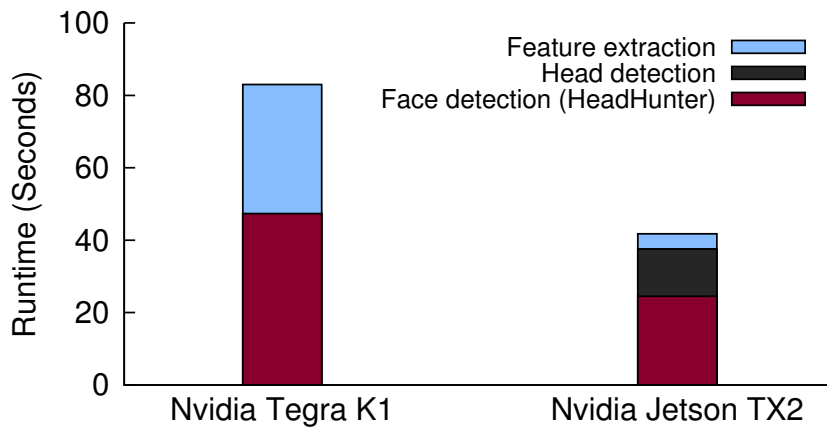
We used the Keysight B2961A Low Noise Power Source [81] to power up and measure the energy consumption of the developer kit. The Keysight B2961A was configured to provide a constant 6 volts with the maximum current capped at 3 amperes. During the actual energy measurements of the three components, all other cables, i.e., display, Ethernet, and USB were disconnected from the board.

Figures 4.6(a) and 4.6(b) show the energy consumption & the runtime of the three components. Similar to Section 3.6.6 we processed an image containing 30 faces for our measurements. Note the head detector was only implemented on the Jetson TX2.

Overall we find that the energy consumption and runtime on the new Jetson TX2 module is significantly better than Tegra K1. This is encouraging since head detection in I-Pic’s new processing pipeline comes with substantial additional computation overhead. In fact, on Jetson TX2, we are able to perform all the three tasks within the same energy (and runtime) budget as that of the face detector on Tegra K1. The head detector is at least twice as energy efficient (and faster) than the face detector on Jetson TX2. The face detector itself is 1.6 times more efficient on Jetson TX2 than on Tegra K1. Feature extraction for 30 faces (using the Caffe framework) is also 8 times more efficient and faster on Jetson TX2 than Tegra K1.



(a)



(b)

Figure 4.6. Energy consumption and running time of I-Pic's components on two different mobile SoCs.

The total energy consumption of all the I-Pic components on Jetson TX2, including face detection, head detection, feature extraction, secure matching and BLE scanning¹ is estimated to be 216.44 Joules. At this rate of consumption, the Jetson TX2 implementation will be able to process 322 images on a single charge of 5100 mAh battery (operating at 3.8 volts), as compared to 288 images measured previously on Tegra K1 in Section 3.6.6. For comparison, I-Pic's original prototype (used in Section 3.6.6), which only uses face detection instead of both face and head detection, running on Jetson TX2 is able to process 576 images on a single charge.

4.3 Conclusion

Overall we conclude that I-Pic can benefit significantly from recent advances both in software and hardware. We explored this possibility, in this chapter, by integrating

¹We have conservatively assumed the combined energy consumption for secure matching and BLE scanning to be the same across the two platforms, the same as that measured on Tegra K1 in Section 3.6.6 (which was 26 Joules)

a state-of-the-art head detector in I-Pic's processing pipeline and by using a powerful new Mobile SoC. We found that with these changes, I-Pic could provide better accuracy at faster runtimes, while operating within a lower energy budget.

Specifically, in Section 4.1, we showed that using a state-of-the-art head detector [17] improved I-Pic's ability to detect all the people in an image from 70% (previously) to 90%. This improvement also increased I-Pic's ability to protect users' privacy from 86% (previously) to 94%. We also modified the I-Pic's processing pipeline to use a combination of face detection and head detection, which also helped in automatically eliminating false positives produced by the face detector.

Using a powerful new Mobile SoC, the Nvidia Jetson TX2, we demonstrated, in Section 4.2, that I-Pic's three main computer vision components could operate significantly faster, while consuming less energy than previously measured on the Tegra K1. The head detector was twice as fast and 2x more energy efficient as compared to the face detector, while providing higher accuracies. The face detector itself performed 1.6 times better on Jetson TX2 than Tegra K1. Overall we found that, on Jetson TX2, all the three components combined could operate at a lower energy budget than that of the face detector on Tegra K1. We further estimated that a complete prototype implemented on the Jetson TX2 platform will be able to process more images on a single charge, increasing from 288 images (previously) to 322 images.

In conclusion, we find these results to be very encouraging as they suggest that there is still room for improving I-Pic's performance, without compromising on its energy efficiency.

Chapter 5

EnCore: Private, Context-based Communication for Mobile Social Apps

5.1 Introduction

Mobile social apps consider users' location, activity, and nearby devices to provide context-aware services (such as Highlight [82], Facebook Nearby[83], YikYak [84], Whisper [85]). Users of these increasingly popular apps are exposed to various privacy risks. Most currently deployed mobile social apps rely on a trusted cloud service [82, 83, 84] to match and relay information, requiring users to reveal their personal information, such as a trace of their location, social networking profile, etc., the perils of which have been extensively noted [5, 6, 7, 8, 9].

Some recent apps [86, 87] additionally use device-to-device (D2D) communication via short-range radio (e.g., Bluetooth, Wi-Fi Direct). D2D communication permits new capabilities: first, devices can precisely identify nearby devices, enabling powerful ad hoc communication and sharing. Second, D2D enables devices to create pairwise shared keys, which can be used to bootstrap secure and private communication without a trusted broker.

Recognizing this opportunity, new secure D2D handshake protocols, such as SMILE [88], SmokeScreen [89] and SDDR [10] have been developed. SDDR provides a *secure encounter* abstraction: pairs of co-located devices establish a unique encounter ID and associated shared key using D2D communication, which encounter peers can subsequently use for secure communication. While specific apps have been built using encounters [88, 89], no platform exists that relies on encounters to enable a wide range of privacy-preserving mobile social communication and sharing.

In this project, we leverage the notion of addressable secure encounters introduced in SDDR to build EnCore, a communication platform that provides powerful new capabilities to mobile social apps, with strong security and privacy guarantees, without requiring a trusted provider. Using EnCore, apps can:

- Rely on encounters to conveniently and securely bootstrap *events*, which represent socially meaningful groups of proximal users and are associated with inferred context and user annotations.
- Send, receive, share, organize, and search information and contacts by referring to events by their name, time or location, while maintaining confidentiality and full control over participants' anonymity and linkability.
- Use *conduits* to distribute and store information within events, before, during and after the actual social event. Current conduits rely on e-mail, Dropbox and Facebook.

To illustrate the space of apps supported by EnCore, consider a scenario of tourists visiting a site. While there, visitors wish to share live recommendations on nearby sights, shows to attend, and eateries to try, but do not wish to reveal any (long-term) linkable information about themselves. If, unbeknownst to them, a friend or person with a shared interest is in the area, they would like to be notified, yet they wish to remain anonymous to all others. At a later time, attendees may like to share content (e.g., photos) and commentary related to the visit, but only with those who were there. Lastly, some might wish to follow up with a special person they met but failed to exchange contact information with. EnCore supports all these capabilities and more.

The primary contributions of this project are as follows:

- We present the design of EnCore and its implementation on Android devices.
- We demonstrate EnCore's capabilities through *Context*, an Android application that provides communication, sharing, collaboration and organization based on events. The application was shaped by user feedback from a series of test bed deployments.
- We report on a series of live deployments of *Context* and EnCore, with 35 users at MPI-SWS.

The structure of this chapter is as follows: in the next section we will describe the related work (section 5.2), followed by a description of EnCore's requirements (section 5.3), and overall design (section 5.4), followed by a description of the *Context* app built on top of EnCore (section 5.5), results from a series of live

deployments of EnCore (section 5.6) and qualitative feedback we received from our users (section 5.7).

5.2 EnCore Related Work

Mobile social apps Most currently deployed mobile social apps like Highlight [82], Facebook nearby [83], YikYak [84] and Whisper [85] rely on a cloud service to match co-located devices and relay data among them. Users must trust the provider with their whereabouts, activities, and social encounters.

More recent systems like LoKast [90], AllJoyn [86], Hagggle [91, 92] and Musubi [93], as well as lost-and-found apps like Tile [94], use D2D radio communication, which enables infrastructure-independent and accurate detection of nearby devices (e.g., those within Bluetooth range). In principle, these systems could be designed so that users do not have to trust the cloud provider with their sensitive data. Unfortunately, once Bluetooth discoverability is enabled, devices can be tracked even when they are not actively communicating, introducing a new threat to privacy. Unlike the tracking of cellular phones by mobile operators, such “Bluetooth surveillance” by stores and businesses is not regulated [95].

EnCore relies on SDDR [10] for D2D radio communication. SDDR incorporates an efficient periodic MAC-address change protocol that ensures users cannot be tracked using their MAC address. The SDDR handshake protocol is provably secure and does not leak users’ identity or profile information except to selected users.

The AirDrop [96] service in Apple’s iOS 7 enables iPhone users to share content with nearby devices. AirDrop uses Bluetooth for device discovery and token setup, and an ad hoc Wi-Fi network to transfer data. AirDrop is designed for synchronous pairwise sharing among co-located users. Android Beam [97] is similar to AirDrop but relies on NFC [98] to initiate communication by physically placing devices back to back, and uses Bluetooth or Wi-Fi Direct [99] to transfer content. EnCore instead enables communication with all encountered EnCore devices, both during and after co-location. Moreover, EnCore prevents tracking, and supports anonymous and group communication.

Life-logging apps Friday [100] keeps an automated journal of user activities such as calls, SMSes, location history, photos taken and music history for browsing and sharing purposes. Memoto [101] is a life-logging camera that takes a picture every 30 seconds. The Funf framework used in the Social fMRI project [102] is a platform for social and behavioral sensing apps. Since all these services upload the collected data to the cloud, users have to trust the cloud provider with their private information.

Private mobile social communication systems SMILE [88] is a mobile “missed connections” application, which enables users to contact people they previously met,

but for whom they do not have contact information. SMILE creates an identifier and an associated shared key for any set of devices that are within Bluetooth range at a given time. Users can subsequently exchange messages (encrypted with the shared key) anonymously through a cloud-based, untrusted mailbox associated with the encounter ID.

In SmokeScreen [89], devices periodically broadcast two types of messages, *clique signals* (CSs) and *opaque identifiers* (OIDs). CSs enable private presence sharing, announcing the device’s presence to any nearby member of a mutually trusting clique of devices (e.g., friends). The sender’s identity can be determined from the signal only by clique members, who share a secret. OIDs enable communication with strangers. A trusted broker can resolve OIDs to the identity of their sender, assuming that two or more devices agree to mutually reveal their identities. In comparison, EnCore supports anonymous communication with strangers without requiring a trusted broker.

SPATE [103] uses physical encounters among mobile devices to allow users to explicitly establish private communication channels, so that they can communicate and share data securely in the future. SPATE does not address anonymity, does not support communication among strangers who did not explicitly introduce their devices, and does not provide a way to address devices by referring to a shared context.

PIKE [104] is a key exchange protocol designed for secure proximity-based communication among the participants of an event. Keys are exchanged using an existing service like Google Calendar or Facebook, which require knowledge of the contact details for each participant. EnCore, on the other hand, leverages encounters to exchange keys with previously known and unknown participants, and without explicit user action.

SDDR: Secure Device Discovery and Recognition SDDR [10] builds on the encounter-based communication style introduced by SMILE, adding selective and unilaterally revocable linkability. The SDDR handshake protocol is provably secure, non-interactive and energy-efficient. SDDR attempts to form a pairwise encounter with each nearby device, establishing a shared key in the process. SDDR can also recognize specific users or users with specific attributes if both peers in an encounter agree to be recognized by each other, while remaining unlinkable by other devices. This *selective linkability* can be revoked and reinstated efficiently and unilaterally by each peer.

To prevent devices from being linked across encounters by their link-layer addresses, SDDR changes the MAC address every “epoch” (roughly every 15 minutes). However, periodic address changes are not natively supported in Bluetooth 2.1 and cause established connections to reset. EnCore uses a Bluetooth 4.0 SDDR implementation that maintains all of the security properties of SDDR over Bluetooth 2.1, and preserves interaction with legacy accessories and devices.

Privacy-preserving MAC protocols SlyFi [105] is a link layer protocol for 802.11 networks that obfuscates packet bits, including MAC address identifiers and management information, in order to prevent adversaries from identifying or tracking users in an application-independent manner. EnCore addresses the complementary concern of enabling anonymous, context-based communication based on encounters. EnCore, however, additionally includes a Bluetooth MAC address change protocol to prevent cross-encounter linking. Bluetooth 4.0 [106] protocol incorporates low-power, low-latency discovery and security extensions relevant to EnCore.

Location privacy Several works investigate location privacy for mobile devices [107, 108, 109, 110, 111]. Roughly speaking, the following two classes of approaches have been proposed. The first class proposes to send fake or perturbed location data, or send location data at coarser granularity [108, 109, 110, 111, 112]. This class of approach essentially trades off utility with privacy. The second class of approaches does not require data obfuscation, but resorts to anonymity [107, 113, 111]. For example, Koi [107] sends unperturbed locations to a cloud server; however, the location is not linkable with a user’s identity (assuming two non-colluding servers). In comparison with these approaches, EnCore achieves location privacy without relying on trusted, centralized infrastructure.

Device discovery Energy-efficient device discovery in wireless networks has been studied extensively [114, 115, 116, 117, 118]. EnCore currently uses a simple, static device discovery scheme, but could easily incorporate the more sophisticated protocols in the literature. Other work aims to enable users to prove that they were in a particular location [119, 120]. EnCore addresses the orthogonal problem of allowing users to prove that they were in the vicinity of certain other devices. The Unmanaged Internet Architecture [121] (UIA) provides zero-configuration naming and routing for personal devices. While it shares with EnCore the goal of enabling seamless communication among personal devices, UIA is not concerned with the specific communication model and privacy needs of mobile social applications.

5.3 EnCore: Capabilities and Requirements

In this section, we describe EnCore’s capabilities in light of the communication requirements of mobile social apps and the privacy needs of users. EnCore provides its capabilities without relying on a third-party provider that is entrusted with users’ whereabouts, activities, and social encounters.

5.3.1 Detecting nearby users and resources

A basic requirement of mobile social apps is the detection of nearby resources and users. EnCore’s secure encounters enable this capability using D2D communication. Detecting nearby resources has several variants:

Discovering when a known friend is nearby Friends can be members of certain online social network circles (e.g., friends, family, colleagues), or specific users that have previously paired their devices. For privacy reasons, a user should be able to control discoverability by individuals or circles manually, and based on the present time, location, or activity. Moreover, a user’s device should be unlinkable by all other devices.

Discovering relevant resources and nearby strangers that match a profile The profile might include interests (e.g., “tango”) or relationship status (e.g., “single male age 27 seeking female”). For privacy reasons, a user should be able to control discoverability by individual profile attributes manually, and based on the present time, location, or activity. Moreover, an attribute should be visible only to devices that advertise a matching attribute.

Keeping a record of (strangers’) devices encountered This record is useful to communicate and share information related to a shared experience, taking place in the present (e.g., sharing recommendations for menu items while at a restaurant) or in the past (e.g., sharing selected photos from a joint tour bus ride). For privacy reasons, this record must not contain personally identifying or linkable information.

5.3.2 Event-based communication/sharing

Mobile social apps enable communication among members of a social event like a meeting or gathering. A key abstraction in EnCore is an *event*, a set of encounters relevant to a social event along with inferred context and user annotations. Typically, an event includes a subset of a device’s ongoing encounters at a given time, and a device may be part of multiple concurrent events. For instance, while at a restaurant, Alice’s device may participate in a dinner event comprising encounters with each of the devices present at her dinner party. Concurrently, her device may be part of a restaurant event comprising encounters with other guests at the restaurant. Both events are socially meaningful, and may be used to share photos and notes about the dinner with her party, and menu suggestions with the other guests, respectively. Note that Alice’s device may also encounter devices of people who pass by outside the restaurant, which are not part of any event.

EnCore is able to infer certain types of events automatically, and users can create named events manually by annotating specific encounters. Events occur naturally as users are presented with relevant encounter and context information. For instance,

moviegoers at a theater might wish to share movie recommendations on the spot, while participants of a sightseeing tour may wish to share selected photos and videos days later. Attendees at a conference might wish to virtually carry on a conversation started in the hallway, texting and sharing links long after the conference is over. Supporting events has the following requirements:

Ad hoc event creation The ability to set up an event without the inconvenience of having to pair mobile devices with every attendee or enumerating every attendee by their contact details. This capability lowers the bar for setting up communication and sharing related to an ad hoc event or meeting.

Event-based communication The ability to send, receive, share, organize and search information and contacts by referring to the time/location or name of the appropriate event. This capability makes it easy to communicate with people one has met on a particular occasion, without needing to remember everyone's name or contact details.

Furthermore, the platform must protect user's privacy and data confidentiality, leading to two additional requirements:

Privacy control To protect privacy, users must retain the option to participate in an event with full contact details, a permanent nickname ("Alice"), or under an unlinkable, one-time pseudonym. The former may be appropriate for a meeting with business partners at a conference, while the latter are appropriate for sharing content related to a shared activity with strangers.

Access control The ability to control access to the event is critical for private event-based communication. An event may be restricted to any subset of those physically present during a specific event, and may optionally include additional users who are invited by a member.

5.4 EnCore Design

In this section, we describe the services supported by the EnCore platform. Figure 5.1 depicts the various components of the EnCore architecture. EnCore uses the SDDR protocol to form D2D encounters, and store these in the EnCore database. The Event Generator component groups, under user direction, related encounters into socially relevant named events, and stores these in the database. Users use applications to communicate with event peers. Depending on the event specification and the type of content shared, the Routing module decides how to forward event invitations, content and notifications to the members of an event. The information is sent using Conduits, which rely on an existing communication, storage or OSN service to effect communication. Applications usually default to specific conduits for particular event

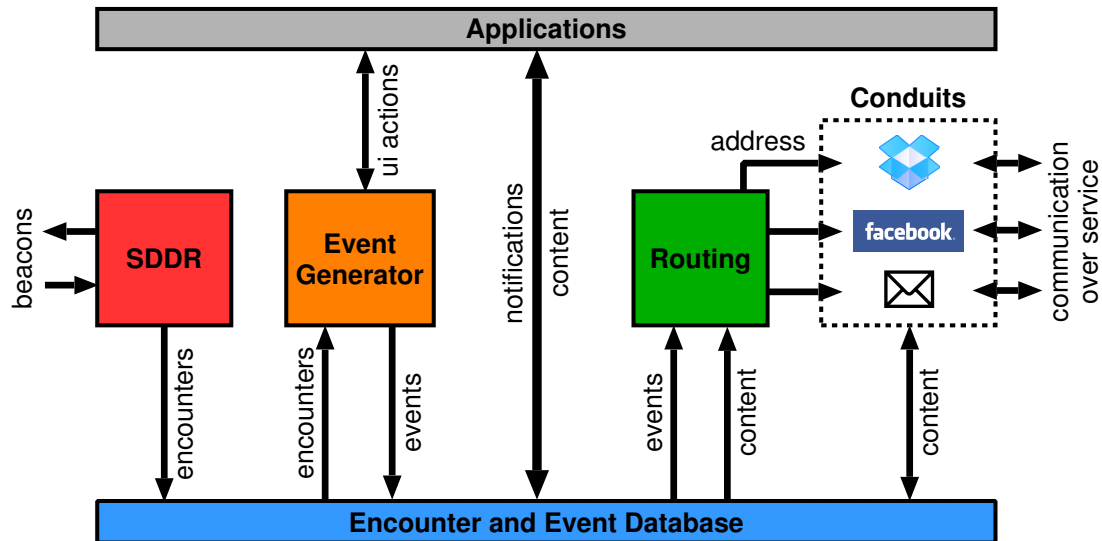


Figure 5.1. EnCore Architecture

and data types, e.g., Dropbox for video sharing. Before describing each of these components, we discuss EnCore’s security properties and threat model.

5.4.1 EnCore security properties

Threat model

We assume that a subset of devices is controlled by attackers who participate in the EnCore protocol as peers and may also collude. Also, attackers can observe network communication and data stored in the Cloud. However, attackers cannot decrypt content without knowledge of the encryption key or invert cryptographic hashes. Furthermore, we assume that user devices are not compromised, i.e., attackers cannot learn honest users’ private keys or the shared keys associated with encounters or events in which no attacker was a participant. Finally, honest users do not share event keys with non-members (users not part of an event).

We assume that user devices, including the operating system and any applications the user chooses to run, do not divulge information identifying or linking the device or user through EnCore conduits or other communication channels (the EnCore protocols themselves do not leak identifiable information, down to the MAC layer.). Finally, we rule out radio fingerprinting attacks, which can identify a device by its unique RF signature [122]. Such attacks require sophisticated, non-standard signal capture hardware, and are outside our threat model.

Security properties

Under the assumptions stated above, EnCore provides the following security properties:

Bluetooth device unlinkability Attackers cannot track a legitimate user’s device across Bluetooth radio contacts, unless the user’s device remains in Bluetooth contact with some attacker’s device for a continuous period that is never interrupted by more than two SDDR epoch changes.¹

Encounter unlinkability/selective linkability Attackers cannot link different encounters with a user’s device, unless the user has explicitly linked their device with an attacker’s device and has not revoked the link.

Communication unlinkability Attackers cannot link communication or posts by a legitimate user in different events, unless the user has explicitly included identifying information, such as a nickname, in the posts.

Anonymity Attackers cannot learn the identity of a legitimate user or user’s device with whom they share an event, unless the user explicitly reveals this identity.

Confidentiality Attackers cannot learn the communication content of events in which no attacker participates.

Authenticity Users can verify that the communication or content received in an event originates from a member.

We highlight that EnCore’s threat model and security properties are mostly inherited from SDDR [10]. In principle, one could use a different platform, such as SMILE [88] or SmokeScreen [89] to support EnCore’s functionality. Minimally, EnCore expects the underlying platform to discover nearby devices, form encounters and provide pairwise shared keys with them. SDDR additionally provides selective linkability and revocation, as well as Bluetooth unlinkability.

5.4.2 Encounters

EnCore uses a modified version of the SDDR [10] protocol for device discovery and for forming D2D encounters. Below we give a brief overview of how SDDR forms encounters and provides selective linkability. Further details, including SDDR’s security guarantees and scalability, are available in [10].

Each device periodically performs a discovery (also known as an inquiry) to identify all nearby devices, collecting their MAC addresses and beacon messages in the process. Every device is also always discoverable, responding to incoming inquiry messages with information on how the discoverer can establish a connection (e.g., MAC address) and an additional payload, referred to as the beacon. This response is sent by the Bluetooth controller autonomously without requiring the attention of the main processor. Therefore, devices must only wake up to perform an inquiry. Otherwise, while simply discoverable, only the Bluetooth controller must be active, allowing the rest of the system to remain in a suspended state (consuming almost no energy).

¹Device unlinkability for Wi-Fi can be achieved using existing work like SlyFi [105]

Once SDDR receives the beacon(s), it forms an encounter with the remote device and computes a shared key. The beacon contains a Diffie-Hellman (DH) [123] public key which is used to compute this shared key.

While processing the beacon, SDDR additionally checks if the device belongs to a known, selectively linkable user. To support this, the beacon also includes a Bloom filter, which represents a set of salted hashes of secrets shared with the devices of linkable users. Two linkable devices advertise the same set member, which guarantees a match in the Bloom filters. The Bloom filters are padded to achieve a uniform load. Moreover, the salt is changed in every successive inquiry, which makes the probability of false positives quickly approach zero with each additional round in which the Bloom filters match. To a third (unlinked) device, on the other hand, the Bloom filters look like randomly changing sets of bits.

SDDR divides time into epochs (typically fifteen minutes long), during which the MAC address and DH public/private key pair remain constant. This allows the devices to track each other during an epoch, but remain unlinkable *across* epochs.

The original SDDR implementation used Bluetooth 2.1 to provide an efficient discovery and encounter formation implementation. Specifically, it encoded the beacons in the additional 240 bytes that the Bluetooth 2.1 Extended Inquiry Response (EIR) feature allows a device to include as part of the inquiry response. However, Bluetooth 2.1 does not support changing MAC addresses, which is *required* by SDDR; otherwise, users could simply be tracked by their MAC addresses regardless of the privacy provided by SDDR. As a result, the original SDDR implementation reset the Bluetooth controller with a new MAC address every epoch, every fifteen minutes or so. While this provided the necessary security guarantees, it also rendered the device unable to maintain long-term connections with other paired accessories such as headsets.

For use in EnCore, we used a Bluetooth 4.0 implementation of SDDR, which provides native support for randomized, ephemeral MAC addresses. This feature enables EnCore to maintain compatibility with legacy accessories. Furthermore, the communication model supported by Bluetooth 4.0 is different from Bluetooth 2.1, so the Bluetooth 4.0 implementation uses a different wire protocol and a FEC-based message encoding scheme. Design and implementation of SDDR over Bluetooth 4.0, though not a contribution of this thesis, is described in [124].

5.4.3 Events

Events are socially meaningful sets of encounters. The *Event Generator* creates events by selecting encounters that are taking (or took) place concurrently and form a social event meaningful to users. There are two methods for generating events: relying on explicit user input from the *Context* application, or using existing user annotations (e.g., the user's calendar entries). Once an event is created, the generator, using

suitable conduits (Section 5.4.4), sends an invitation to all participating encounter peers, containing an event ID and a shared event key, which can be used for communication among event members. For privacy reasons, users are required to explicitly invite others for events inferred from their private calendar entries.

EnCore provides several methods by which users can create events: For small meetings, all participants tend to interact in close proximity with one another, and thus all devices form encounters with each other. Users can manually select appropriate encounters, using cues such as whether an encounter corresponds to a known user, or a received signal strength indicator (RSSI), which helps to distinguish between nearby and distant users.

For larger events and future events, it is inconvenient or impossible for one user to select all participants from the set of encounters they observe at the time of event creation. If the event is managed, and has a list of attendees, it is possible to bootstrap the EnCore event similar to PIKE [104], using Facebook or another existing registration system. However, unlike PIKE, EnCore can also handle ad hoc events. For these events, the event creator can specify a time period and geographic area, such that any devices within the specified space-time region is automatically invited to the event. This can be implemented by having each event member forward invitations over their encounters that meet the spatial and temporal constraints. Evaluating such policies in a large scale deployment is part of ongoing work.

In designing EnCore, we chose not to use protocol means to disambiguate multiple EnCore events that correspond to the same social event. Thus, users are free to create multiple EnCore events for the same social event, or (somewhat more commonly for large events), a few users may end up creating their own EnCore event corresponding to the same social event. Our experience is that event peers themselves resolve this ambiguity by gravitating to one event, abandoning the others without requiring an arbitration protocol (this was observed in our deployment as well, see section 5.6). Of course, applications atop EnCore may choose to provide their own arbitration protocol.

5.4.4 Communication

All application-level communication in EnCore occurs among the members of an event. Two types of components within EnCore are responsible for communication, *Conduits* and the *Router*.

Conduits are adapters to existing communication, storage and online social network services, and are used to convey information between event participants. Conduits accept messages or content and, depending on the type of conduit, either a list of encounter IDs (the communication end-points for pairwise message transport) or an event ID (the rendezvous point for group communication and sharing). They convert the encounter IDs or event ID into addresses or names used by the underlying

communication, storage, or OSN service that the conduit relies on. To provide secure communication among the members of an event, conduits normally encrypt the communication using either the shared encounter key(s) established during the handshake protocols, or the shared event key distributed during the event creation.

The *Router* component decides, based on the event specification and the type of information shared, how information is forwarded among event members. Three types of information are routed: event invitations, content, and notifications. If the conduit used for the event and information type supports multicast or shared storage, then the router delivers the information in one step, using the event ID as an address and the event key to encrypt. If the conduit supports pairwise communication, then the router sends the information to each member with which the local device shares an encounter, using the encounter IDs as addresses and the associated shared keys to encrypt. If not all pairs of event members share an encounter, then the routers on each member device forward the information to all of their local encounter peers that meet the event membership specification and have not already received it.

5.4.5 Security guarantees

Building EnCore on SDDR guarantees the security properties related to unlinkability. Since SDDR requires periodic MAC address changes, devices are not linkable at the Bluetooth layer unless the tracking device is present whenever the SDDR device changes addresses. Similarly, since SDDR ensures that the advertisements do not carry identifying information (except to linked users), devices remain unlinkable. The confidentiality and authenticity guarantees are provided by ensuring that all communication is encrypted and protected by a message authentication code, using either an event key or an encounter key. Since only encounter or event peers possess the same shared keys, this ensures both confidentiality (event peers can decrypt) and authenticity (only encounter or event peers can post).

5.5 Using Events with Context

We have developed an Android application called *Context* over EnCore². *Context* maintains a private record of the user's activities and social encounters, and allows users to communicate, share, collaborate, organize and search information and contacts using events. The design of *Context* was shaped significantly by user feedback during a series of testbed deployments within our institute community between September, 2012 and September, 2013.

Even though *Context* still lacks the feature wealth, sleekness and visual polish of a commercial product, users in our institute-internal deployment have generally found

²The code for the *Context* app is available at <https://people.mpi-sws.org/~paditya/code/encore.html>

Context useful, and have come up with creative uses for its capabilities. We provide quantitative details of *Context*'s usage in our deployment in Section 5.6, as well as qualitative user feedback in Section 5.7.1. In the rest of this section, we briefly describe the main functions provided by *Context* and their value to the user: browsing the user's timeline and identifying socially relevant encounters, managing events, posting and receiving information, and managing user linkability.

5.5.1 Browsing the timeline

Figure 5.2 shows a screenshot of *Context*'s timeline view for a hypothetical user Bob. Bob can navigate this view by scrolling, zooming or searching by keyword or date.

The timeline view shows encounters, events and calendar entries as horizontal bars spanning time intervals. For linked encounters, the peer's name is displayed. Anonymous encounters show "Unknown" as the peer name. The height and color of the encounter bar indicates signal strength and is a rough proxy for proximity. This view scrapes the user's calendar and displays previously scheduled entries. Any EnCore events are displayed in the events area. Events are marked with pending invitations or notifications (if any).

Selecting any UI element reveals more information about, and shows a menu of possible actions on the element. For example, selecting an event highlights the participating encounters, allows the user to inspect or edit the event's metadata, invite more participants, or launch an application to browse the event's content. Selecting a location switches to a map-based view.

This simple linear view provides remarkable functionality: For instance, as shown by (1) on the Figure, by navigating back to this view, Bob can remind himself that he was with Alice at the table tennis championship before lunch on September 10, and eventually they walked to lunch together. The events pane shows that Bob was invited to the associated event and has a pending notification.

The rectangles (2) and (3) show how social events, both scheduled and impromptu, naturally line up vertically along the timeline. (2) shows that Dave joined Bob and Alice for lunch, and that there was someone else (unknown with high signal strength) nearby. This may be a sufficient hint for Bob to recall that Dave was with his guest at lunch. There were other lower signal strength encounters with unlinked users at lunch. Similarly, (3) shows a scheduled event, the Reading Group, that has a calendar entry and an associated EnCore event. Once again the vertical alignment of the high signal strength encounters with Kelly and Jack serve as reminder that they were at the reading group meeting. The encounter with Amy has low signal strength and likely is an artifact of her being in a nearby room but not at the reading group meeting.

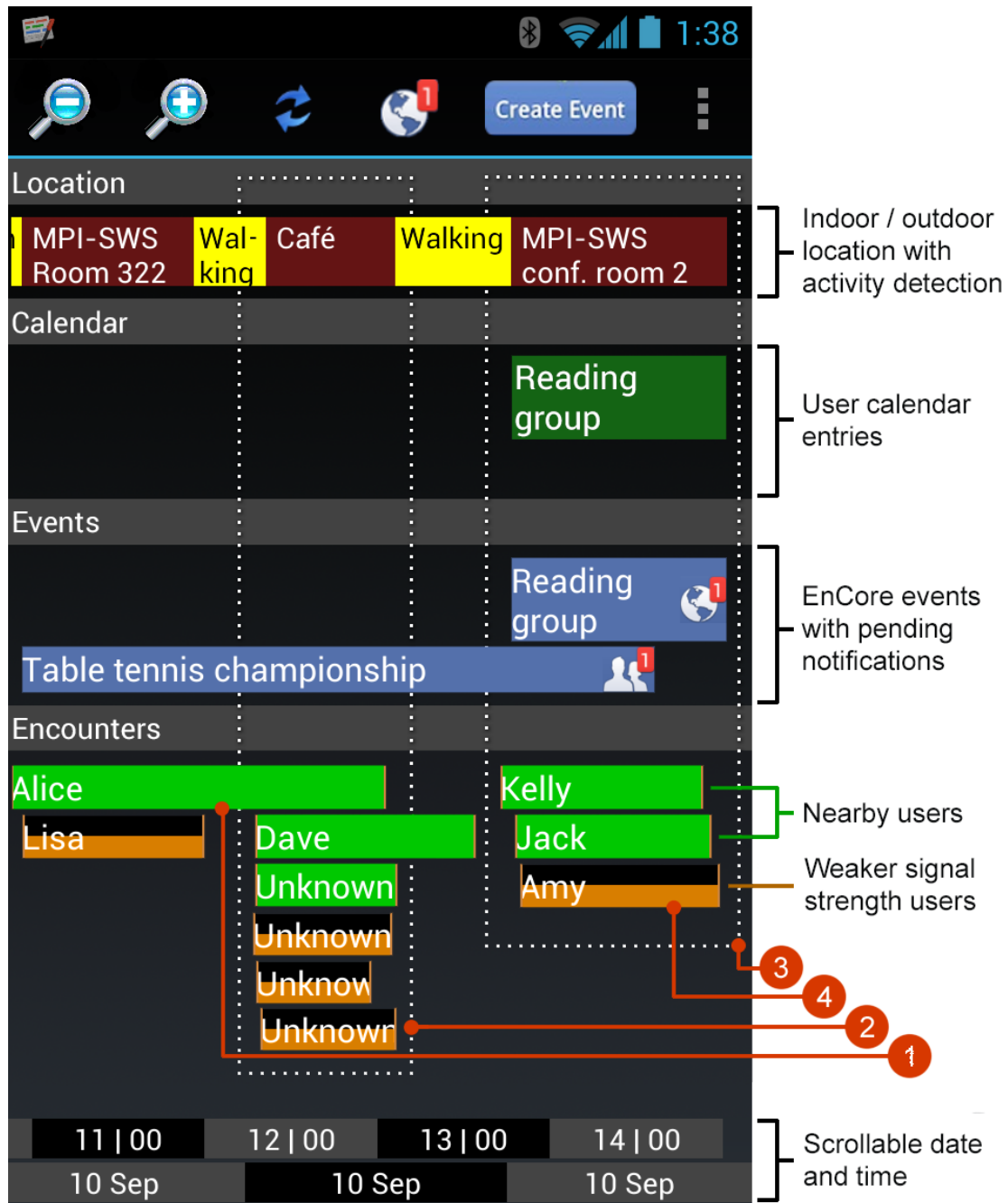


Figure 5.2. *Context* Timeline view (User names were changed for privacy)

5.5.2 Creating events

Users create events by touching the “Create Event” button and selecting a set of encounters to be included. If the event was previously scheduled in the calendar, its metadata (name, duration) is automatically imported; otherwise, the user can adjust the default duration inferred from the selected encounters, enter a name for the event, and optionally add the event to the calendar. Once the user confirms, the event is created and invitations are sent to the selected encounters. To support events where some of the users are not physically present (e.g. users attending an event virtually), event members can additionally invite any of their past encounters or known contacts to the event.

For more complex events or future events, the user can specify temporal and spatial (e.g., within the current building) constraints for included encounters. Future and transitive encounters that meet these constraints are invited automatically.

By default, the appropriate conduit to implement an event is chosen automatically. When all participants are linked encounters who provide Facebook account details (as is the case in our deployment at users’ request), then a conduit is chosen that maps the event to a private Facebook event. Otherwise, a conduit is chosen that maps the event to a folder in Dropbox. The Dropbox conduit supports anonymous participants and provides the same basic sharing functionality, albeit without the integration and the sophisticated event presentation of Facebook.

These facilities make it easy to set up communication and sharing among a socially meaningful group of users in an ad hoc fashion. The event creator does not require contact details of the participants, and can include anonymous users via unlinked encounters.

5.5.3 Posting information

Context appears as a choice in Android’s Share menu.³ Therefore, any type of content can be selected (e.g., pictures and videos from the Android gallery, audio from a recording app, pin drops from a map app, text from a notes app) and shared via *Context*. Within *Context*, the user simply touches an event in which the content is to be posted.

As a convenient shortcut, users can post information directly from within *Context* and select encounters with whom the information should be shared, without creating an event. Internally, *Context* creates an event with default metadata to handle the posted information.

These facilities make it very convenient to send messages and share content with nearby or previously encountered users.

³Most Android content apps have a “Share” button, which opens a menu of applications through which the selected content can be shared.

5.5.4 Receiving information

Notifications about incoming messages, posts or pending event invitations are shown as icons with red flags on the corresponding event, or on an encounter in the case of a message sent directly to an encounter. For instance, in Figure 5.2, there is a new post in the “Reading group” event, and a pending invitation to the “Table tennis championship” event. Notifications are also summarized in the notification center shown as an earth icon at the top of the screen. Touching it will scroll to the nearest event or encounter with a pending notification. Preference settings allow users to suppress notifications by type or source.

To respond to an event invitation, the user touches the event, optionally views the event’s metadata, and then accepts, declines or defers the invitation. To read incoming messages or posts, the user selects the relevant event or encounter. Touching an event launches an external application (e.g., Facebook) to show the latest post in the event.

These facilities enable users to prioritize, filter, browse and navigate incoming information according to its context: event, encounter, time and location.

5.5.5 Controlling linkability

Context allows users to control the information revealed in an encounter in a variety of ways. The user can choose to reveal a linkable nickname or their real identity to selected peer devices. The linkable peers can be selected based on existing relationships in an online social network (e.g., Facebook) or a contact list, or by pairing devices individually. Moreover, linkability can be controlled based on the user’s present location or time. For instance, users can choose to be linkable to colleagues only when in the office and not be linkable by anyone at certain times.

Recall that a Facebook private event is used by default for events among linkable encounters who provide Facebook details. User posts are linkable across such events. Users can use a separate Facebook account under a pseudonym for this purpose; in fact, all participants in our deployment use test accounts separate from their main Facebook account. To avoid linkability across events, the creator of an event can choose to use a Dropbox conduit instead, and users can decline invitations to Facebook-backed events if they so choose. These facilities enable users to effectively control their privacy.

5.5.6 Implementation of conduits and router

EnCore and *Context* currently support the following conduits:

SMTP conduit The SMTP conduit allows users to securely exchange e-mails with the participants of an event. The SMTP conduit allows any email client on the device to send a message to the email addresses associated with one or more encounter peers. If an encounter is linkable and has an associated email address, the message

is simply sent to that address. If the encounter is unlinkable, then a one-time email address is derived from the encounter ID, and the message is sent to that address. The current implementation uses `mailinator.com` [125] for this purpose, which does not require user registrations and creates mailboxes on-the-fly as mail arrives for an address. The mail is public for all who can guess an email address, but this does not affect confidentiality⁴ since all EnCore mail is encrypted using an encounter or event key. The Mailinator conduit is limited by the fact that Mailinator caches messages only for a few hours, and applications need to periodically resend messages to ensure persistence. Alternatively, one could easily setup a similar one time email system that does not delete messages⁵.

Dropbox conduit The Dropbox conduit converts an event ID into a folder name on Dropbox, and stores all content posted to the event into that folder, encrypted with the shared event key.

Facebook conduit This conduit associates an EnCore event with a private Facebook event. It requires that all participants in the event are linkable and provide details for a Facebook account. The event's participants appear in the Facebook event with the identity of the account they provided. Textual posts, comments, likes and photos are posted in the Facebook event in cleartext, to maintain the flexibility and convenience of the Facebook interface⁶. However, video and audio recording posted to the event are uploaded using the Dropbox conduit (encrypted with the shared event key), and a URL to that content is posted in the Facebook event.

Using Facebook allowed us to leverage the familiarity of users with its app. The Facebook conduit cannot support unlinkable users, which was irrelevant in our deployment. As part of ongoing work, we plan to recreate similar functionality within *Context* with the ability to create events amongst unlinkable users.

Router The current router implementation is limited to forwarding information within events in which all members share pairwise encounters. We are in the process of adding transitive forwarding. There has not been much demand for this feature so far, due to the relatively small size of our deployment and the types of events users have requested.

5.6 Evaluation

In this section we report on the field deployment of EnCore. Our latest EnCore field deployment began on September 9, 2013, with 35 volunteers in the Saarbrücken office of MPI-SWS. The participants were staff members and researchers, and were informed

⁴It does reveal that a message was sent to particular encounter or event, but the actual content is not revealed.

⁵We implemented such a system and used it during one of our initial deployments.

⁶Note that Facebook has access to cleartext posts; this can be avoided by using the Dropbox conduit or a private OSN platform like Persona [126] to share all information.

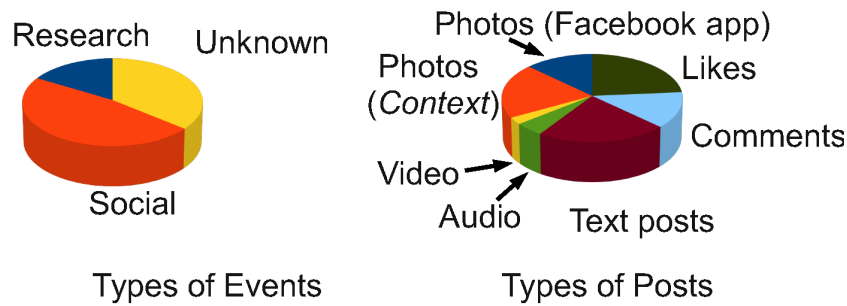


Figure 5.3. Types of events and posts

about the purpose of the experiment and what data would be collected. Most of the participants (32 out of 35) were not directly involved in the project.

Deployment setup We provided each participant with a Galaxy Nexus phone running EnCore with *Context* and the Facebook app. All phones were configured with the account details of a different Facebook test account with a pseudonym. At users' request all phones were selectively linked with each other by default. Users were able to change the linkability settings, configure their personal calendars for display in *Context*, and change the pseudonym to their real name if they wished to do so (30 of the 35 users did). None of the users modified the default linkability setting (i.e., link with all other users). This is not surprising since the deployment was carried out among mutually trusting users and linkability was limited to experimental devices only.

We requested users to carry the device, and encouraged them to use *Context* to create events, communicate and share content as they saw fit. On September 16, we installed an audio recording and a note taking application on each of the devices because several users requested it, in addition to the default camera, gallery, calendar and map applications already available on each device.

The phones ran EnCore using the Bluetooth 4.0 implementation of the SDDR protocol. The phones executed discoveries every 13.5 seconds and changed MAC addresses every 15 minutes.

Statistics from the deployment After the the first two weeks of the deployment, we collected statistics for all the events created, which we present below. The users' activity level was roughly bimodal. 17 users created fewer than three events and made fewer than five posts, while 18 users exceeded these numbers. Among the users not related to the project, the maximum number of events created by a single user was 12 (median 3) and the maximum number of posts created by a single user were 30 (median 4).

Event usage After removing from consideration a number of events that had been created by project members for demonstration purposes, a total of 128 events remain. We have divided these events into three categories based on their names: research

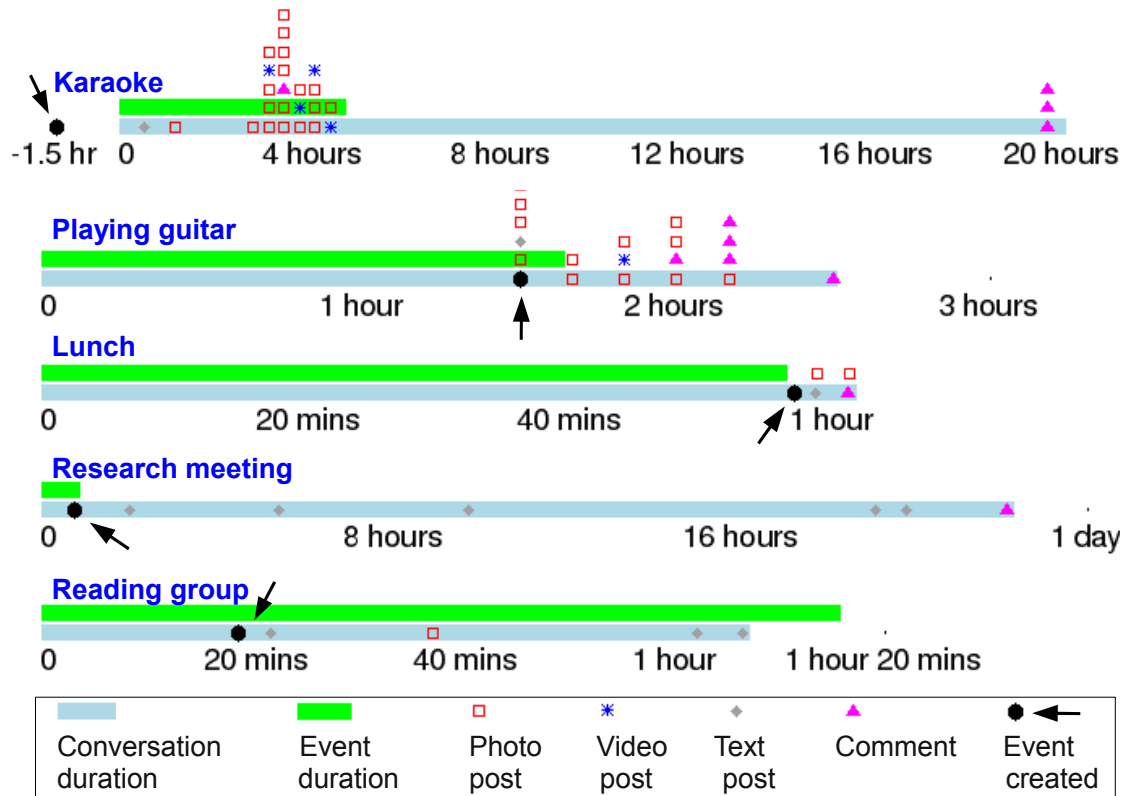


Figure 5.4. Timelines of five actual events. Each point shows the type of activity performed along with the time elapsed since the specified start time of the event

meetings (16%), social events (48%), and unknown (36%). We classified an event as unknown if its purpose was not obvious from its name. Figure 5.3 shows the distribution of event types and Figure 5.4 presents the timelines of a selection of five actual events created during the deployment.

Based on informal feedback from users and our own observations, there was an interesting mix of expected and creative uses of events. Events were created for research gatherings, such as meetings, reading groups, etc., and used to exchange meeting notes, audio recordings, followup comments and links to related papers. The ‘research meeting’ and the ‘reading group’ events in Figure 5.4 show the activity timelines of two events in this category. Also, as seen in Figure 5.5, these events tend to contain a moderate number of participants, which is what we commonly observe for project/group meetings and reading groups in our institute.

Almost half of the events were created for social or informal gatherings, such as lunches, coffee breaks, sports activities, bus rides, karaoke events, etc., and used to share photos, videos and comments during and after the event. The ‘karaoke’, ‘playing guitar’ and ‘lunch’ events in Figure 5.4 are typical examples. With the ‘karaoke’ event

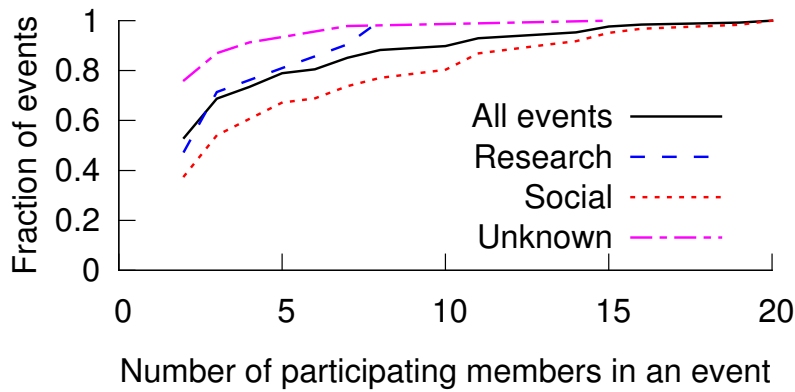


Figure 5.5. Distribution of event size

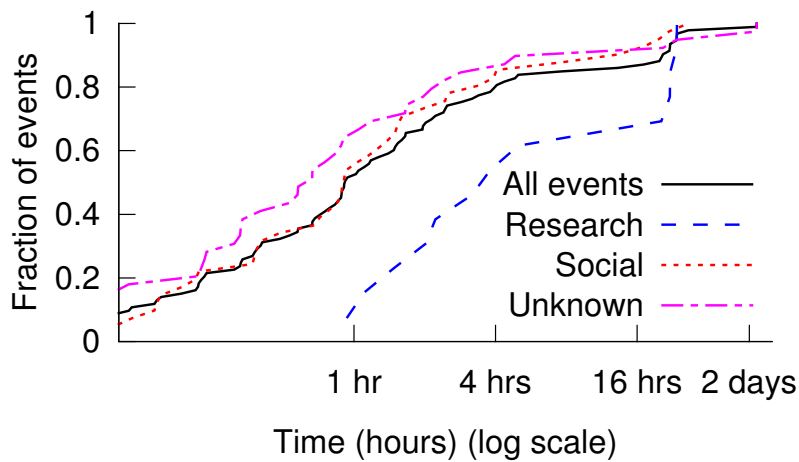


Figure 5.6. Distribution of conversation durations

we also observed an instance of users resolving multiple EnCore events created for the same social event by gravitating to one event (see Section 5.4.3). We observed that users stopped posting to one of the two events created for karaoke and continued their interactions on the other. Some creative uses include creating events to invite nearby people for coffee breaks, or to inform nearby people about leftover party food using a picture of the food. The number of members in social events ranges from 2 to 20 users (median 3), which is expected given the size of our deployment (Figure 5.5).

Figure 5.4 also highlights that events can be created after the associated event has ended, and that conversations tend to extend beyond the event duration. The former can be observed for the ‘Lunch’ event by comparing the time when the EnCore event was created (circular dot with an arrow) and the timespan of the actual social event (the first horizontal bar from the top for each event).

Figure 5.6 shows the distribution of conversation durations for different types of events. Even though most of the events did not have conversations longer than five

hours, there were cases where users referred back to events they had attended in the past and posted content to them long after the actual event had finished.

Figure 5.3 shows the distribution of the types of posts within events. Note that the audio recording application was installed one week after the deployment had started; the proportion of audio posts might have been higher had it been installed from the beginning. A large portion of photos were uploaded directly from *Context*, rather than via the Facebook application, which suggests that users found it convenient to refer to an event directly from *Context*'s timeline.

Summary EnCore and *Context* provide a basis for exploring secure ad hoc interactions. Both the analysis of data from the deployment and personal user feedback show that real users find the paradigm useful and found new ways to collaborate and share with colleagues using *Context*.

5.7 Discussion

In this section, we describe the qualitative feedback we received from our users, and discuss the remaining risks and challenges.

5.7.1 Qualitative user feedback

Quantitative performance evaluations are often inadequate in capturing the utility of new functionality. User engagement can be an important metric, and here we describe the qualitative feedback we received from our users, both during and after the test deployments. At the end of our latest deployment, many of our users expressed an interest in using the system on a permanent basis once we have a version they can install on their primary devices. We believe this is encouraging since it shows that users (albeit highly technically proficient ones) find the system useful. In the rest of this section, we discuss features that users have requested. We believe feature requests are illuminating. While they obviously point out shortcomings in the existing system, they also point to innovation enabled by, and creative use of, EnCore's capabilities, some of which were not anticipated by the design team.

Support for sharing audio recordings and import/export of events to/from calendar was requested by users and rolled out during the last deployment. Various requests were related to encounter export, to make information about nearby users available to other applications. For instance, exporting a "Nearby" group into the Android address book, which includes the contact details of currently nearby users, if known.

Another feature requested was the ability to create ephemeral pseudonyms as a way to control linkability (in addition to the pairwise linkability offered by EnCore). These pseudonyms allow other users to link their encounters with a device under a pseudonym for a limited period and location (e.g., while attending a conference). Users can change

(or remove) their pseudonym and can also choose to stop receiving messages addressed to a pseudonym anytime.

A frequently requested feature is the ability to browse the EnCore timeline, post content and manage events from a desktop computer. Users have also requested the option to thread recurrent or related events, and view posts and comments within a thread in a linearized manner. Finally, users asked that *Context* suggest events and content to post, based on the users' history, preference, and current context. For instance, if Alice, Bob and Charlie have had frequent meetings recently, then *Context* could automatically suggest another instance of the event when it notices similar circumstances (encounters, location, time). These feature requests suggest that users find it useful to be able to explore an encounter timeline, coupled with the ability to create links, and to relate and recount events.

5.7.2 Risks and challenges

User privacy and security has informed every step of EnCore's design. Unlike virtually all existing mobile social apps, EnCore does not require the user to reveal their sensitive context data, which often combines location, social contact and communication trace, to a provider. Moreover, EnCore prevents Bluetooth device tracking and provides strong security and privacy guarantees within its threat model. However, privacy risks and usability challenges remain.

EnCore database confidentiality The data logged by EnCore resides on the mobile device, and is susceptible to loss, theft, or subpoena. It is not clear what legal rights regarding privacy and self-incrimination, if any, users can assert with respect to data stored on their personal devices. Encrypting the EnCore database protects the data in the case of loss or theft, though it will not stop a court from compelling the user to provide the decryption keys. The risk can be somewhat reduced by configuring the database to store a limited history. Since the usefulness of encounter information likely diminishes over time, the resultant loss of functionality may be acceptable.

Private profile matching Linking encounters based on shared attributes is supported by EnCore, but currently not fully exported by *Context*. A challenge in this regard is how to prevent attacks where a malicious device advertises attributes in order to learn as many attributes of nearby users as possible. The problem can be partly mitigated by ignoring devices that advertise too many attributes or change their attributes too frequently, but a more general defense is hard, unless attributes can be certified by an external authority.

Reliably identifying socially relevant encounters Identifying relevant encounters (e.g., the participants of a shared event) was not a problem in our deployment. The fact

that all participants revealed their name or a pseudonym while in the office, combined with the signal strength indication, proved sufficient.

However, identifying socially relevant encounters in a larger and denser environment with many unlinkable devices is an open challenge. For instance, it is important to reliably identify the attendees of a private, closed-door meeting that takes place in an office building with EnCore devices in adjacent rooms. We are currently experimenting with an audio-based confirmation protocol, where devices have to answer a (fast attenuating) challenge transmitted as an audio chirp, in order to identify devices in the same room. Another option would be to follow a non-interactive approach similar to that of Sound of Silence [127], where each device uploads a signature of their acoustic environment to a cloud service that, by comparing signatures, identifies nearby devices.

In other situations, like a crowded party, distinguishing individual attendees is usually not necessary, because the most likely types of interaction (e.g., sharing photos) are directed to the group as a whole. In situations where users wish to identify individuals within a crowded space (e.g., a dinner party at a busy restaurant), people tend to know each other and have their devices linked already. If not, they can resort to bumping devices via NFC or shake-to-connect [128, 129]. In this case, no contact details would be exchanged (unless desired), but the encounter would be marked as “confirmed” on both devices.

Communication with strangers The limited deployment within our institute has not yet allowed us to experiment with communication among strangers, as it would occur, for instance, in the sightseeing scenario described in the introduction. This case, as well as other challenges described above will require experience with deployments at larger scale. Toward this end, we are developing a version of EnCore that does not require rooting the phone, which is currently a major hurdle for a larger deployment. Nevertheless, we believe we have shown that EnCore provides a robust foundation for building secure, privacy-preserving mobile social applications that exploit the opportunities afforded by D2D communication and secure encounters.

5.8 EnCore Summary

We have described the design, implementation, and evaluation of EnCore, a mobile platform for social applications based on secure encounters. EnCore can support a wide range of event-based communication primitives for mobile social apps, with strong security and privacy guarantees, without requiring a trusted provider, and while integrating with existing communication, storage and OSN services. As part of our evaluation, we have conducted small-scale deployments of *Context*, an app for event based communication, sharing and collaboration. User experience was favorable:

users were engaged, requested new features, and used the app in interesting ways not envisioned by us.

While our small-scale deployments have been invaluable in developing the system, secure encounter-based communication promises more than what we have been able to evaluate among a small set of mutually trusting, technically savvy users. Evaluating EnCore's primitives in dense environments and among strangers requires larger scale deployment onto a more heterogeneous population. Our experience catalogued in this document gives us confidence that EnCore and the secure encounter primitive will continue to prove useful, and a larger userbase will yield compelling new ways to communicate using EnCore.

Chapter 6

Discussion and Future work

I-Pic has opened up interesting new opportunities for protecting bystander privacy in image capture. Using I-Pic, bystanders can securely communicate their privacy preferences to nearby photographers, and have these preferences automatically applied to photographs they appear in, without having to personally approach the photographer or vice versa. In Chapter 3 we presented one end-to-end design for I-Pic that was deployable and consistent, one that individual users could start running.

In this chapter we explore alternative designs for I-Pic. We give suggestions on how one can combine the technical blocks used within I-Pic (and EnCore) in different ways to extend I-Pic beyond its current capabilities and create systems with different properties.

This chapter is structured as follows. In Section 6.1, we envision a system that extends I-Pic with EnCore’s encounter-based communication platform, which enables anonymous communication between groups of nearby users. We explore how this combination would enable completely new ways of specifying & communicating privacy preferences. In Section 6.2 we explore how specifying privacy preferences can be made completely dynamic by performing enforcement of privacy preferences on the software used for viewing an image on an end user device. In Section 6.3 we give a brief description of how I-Pic can be extended beyond still images to protect privacy of bystanders in video and audio recordings. In Section 6.4 we describe how I-Pic can be extended to offload computer vision tasks to a trusted agent, such as photographer’s own desktop/server machine. In Section 6.5 we explore how trusted computing technology, such as ARM TrustZone [130], can be used to extend the threat model of I-Pic to ensure that privacy preferences are applied even if the photographer’s device is rooted. In Section 6.6, we discuss other future work directions for I-Pic, such as how to best remove someone from an image, and, if needed, how to accommodate requests to override users preferences from authorities, journalists, and professional photographers.

6.1 Leveraging EnCore for I-Pic

Imagine a privacy preference that states: “I am OK appearing in photographs, but please send me a courtesy copy”, or “I will decide my privacy preference after seeing the photograph”. These are examples of alternative ways of specifying privacy preferences that a bystander might find useful while using I-Pic. While it is possible to broadcast these privacy preferences in I-Pic, to enable any subsequent interaction between users, they would also need to broadcast their contact details to nearby strangers, which might not be desirable. In principle, EnCore’s anonymous encounters and event-based communication could bridge this gap.

EnCore provides a set of convenient features to enable secure communication between co-located users. First, EnCore automatically discovers other nearby users using Bluetooth and forms pairwise encounters. Subsequently, a pair of users who have shared an encounter can communicate with each other using EnCore’s cloud-based *conduits*, even when they are no longer within Bluetooth range. Furthermore, communicating via encounters does not reveal any linkable information about the users, unless they choose to do so themselves. Finally, the *Context* app, built on top of EnCore, provides additional contextual information to identify socially relevant encounters, enables grouping encounters into *events*, and provides a platform for sharing digital media by addressing it to encounters (or *events*).

I-Pic can leverage these features to enable photographers to anonymously communicate with bystanders who were present at the time a photograph was captured, regardless of whether they were captured in the photograph. Furthermore, using visual signatures broadcast by bystanders, a photographer’s device could also automatically associate a bystander’s face, captured in a photograph, to an encounter with that bystander’s device. In principle, the photographer could then initiate a conversation by simply selecting a face from the photograph¹. On the bystander’s device, this conversation will show up associated with a specific encounter, inside the *Context* app. This encounter would then serve as the communication channel between the bystander and the photographer.

Overall, integrating EnCore with I-Pic would enable the photographer to interact with bystanders who were captured in a photograph, or to interact with those bystanders who were present at the time a photograph was taken but were not captured in it. Having these abilities in I-Pic would be powerful new additions that would change the user experience in a profound manner, both for the photographer and the bystander. A user would no longer be limited to specifying upfront a privacy preference based on time, location, and who the photographer is. Instead, bystanders could choose to be notified

¹The Photographer’s device could also initiate a conversation automatically in response to requests from bystanders. In Section 6.1.1 we present examples of such interactions.

when they are captured in a photograph, and interactively respond with their privacy preference, based on the content of the photograph. A photographer could also get in touch with a bystander, who had originally chosen a restrictive preference, and ask them to change their preference, in return for a copy of the photograph. These are just two examples of the many new workflows/interactions that could be enabled by integrating EnCore with I-Pic.

6.1.1 New workflows enabled by integrating EnCore with I-Pic

Below we present a comprehensive list of new workflows that could be enabled by combining I-Pic with EnCore's encounter-based communication platform.

- **Capture-notifications:** A photographer may be interested in notifying all bystanders who were captured in a photograph, possibly to allow them to conveniently contact the photographer in future. To do this, first, the photographer's device would match encounters to bystanders captured in a photograph using visual signatures received from bystanders. These matched encounters would then be used to notify captured bystanders. This workflow could be automated on the photographer's device, and I-Pic could be configured to send these notifications by default. An extension to this workflow could involve broadcasting notifications to unmatched encounters as well, informing those bystanders that they may have been captured in a photograph.
- **I am OK, but send me a copy:** Bystanders could specify a privacy preference stating that they are OK appearing in a photograph, as long as they receive a copy of the photograph. This workflow could be executed automatically by the photographer's device, without requiring any input from the photographer.
- **I will decide after seeing the photograph:** Bystanders need not specify a privacy preference a priori. Instead, bystanders could broadcast their visual signatures, along with a preference stating that they would prefer to decide their privacy preference after seeing the photograph. At a later time, the bystander would receive a notification from the photographer's device, along with the photograph. The bystander could then swipe right (or left) to indicate that they are OK (or not OK) about appearing in the photograph. Until the decision is made, a bystander's face will be conservatively blurred on the photographer's device. This workflow could also be processed automatically by the photographer's device, without requiring any input from the photographer.
- **I would like to change my decision:** Bystanders could retroactively change their privacy preference for photographs they were earlier captured in. With

capture-notifications enabled on photographers' device (described in the first workflow), bystanders who retroactively wish to change their privacy preference could send an update over the encounters from which they had received a capture-notification. On the photographer's device, these updates could be automatically applied to the necessary photographs. Note, that this workflow would require storing an unedited (possibly encrypted) version of the photograph on the photographer's device. In Section 6.2 we explore the possibility of enforcing the privacy preference on the viewing software of an end user device. This would allow bystanders to completely dynamically specify and modify privacy preferences.

- **Do I appear in any photographs you captured?** If bystanders are interested in obtaining photographs they might have been captured in, they could communicate their visual signatures to past encounters at a certain time and place. A photographer's device, receiving such a request over an encounter, would automatically compare accompanying visual signatures to photographs that were captured during the time period of that encounter. The photographer could then respond back to bystanders whose visual signatures produced a positive match.
- **Please reconsider your preference for this photograph:** A photographer could reach out to a bystander captured in a photograph who had broadcast a restrictive preference, to request him to change his preference. The bystander could request a copy of the photograph before making his decision. We assume that the bystander had also broadcast his visual signature at the time the photograph was captured. If the bystander was blurred out by default, which happens when a captured subject cannot be matched with any of the received broadcasts, then this workflow would not be possible.
- **I am a pro-photographer and would like to seek your permission:** A professional photographer might be legally required [11, 12] to seek explicit permission from all people appearing a photograph before publishing it further. To do so, the photographer could send messages over all (or specific) encounters that were ongoing (or created) at the time a photograph was captured. This initial interaction would be used by the photographer to reveal their identity, which could then be followed up by a formal request, if the bystander chooses to respond. These interactions could also be used by the photographer to offer to pay bystanders for their permission to appear in the photograph.

6.1.2 Privacy concerns

The workflows described in Section 6.1.1 could greatly improve I-Pic's user experience, but they could also raise privacy concerns that were not present in I-Pic's original prototype. Below we describe these concerns and possible solutions for them.

Most of the workflows presented in Section 6.1.1 involve sending photographs to bystanders. Therefore, it is important to consider which faces are clearly visible in photographs that are sent. Sending a photograph with all faces clearly visible would violate the privacy preferences of all the bystanders who had chosen a restrictive privacy preference upfront. Furthermore, the photographer might not even be aware of privacy preference of all the bystanders, as some bystanders might have chosen to decide their preferences after seeing the photograph. So a conservative choice will be to blur out all faces except the one for the bystander to whom the photograph is being sent.

The accuracy of I-Pic's computer vision pipeline will affect how well the approach described above works in practice. A combined false positive and false negative in face recognition could lead to a situation where a bystander receives a photograph with his own face blurred out, and a different face clearly visible. To conservatively accommodate for such possibilities, the photographer could start by sending out a photograph with all faces blurred. If a bystander is then interested in obtaining a photograph with their face clearly visible, the photographer could follow up with additional steps of verification. The aim of these steps will be to verify that the bystander who is requesting a photograph actually appears in that photograph. These verification steps could include: requesting the bystander to locate their face in the blurred photograph, and/or challenging the bystander to provide a selfie with them holding a card that displays a random number sent by the photographer's device. These steps would have to be manually verified by the photographer. To reduce the cognitive burden of verifying these steps, they could be integrated within existing workflows of a device's photo gallery application. For example, Google Photos, the default photo gallery application of many Android devices, provides a feature for automatically sharing photographs with captured bystanders [131]. Before actually sending the photos, confirmation for sharing these is obtained from the users while they are browsing photos within Google Photos.

Another privacy concern could be that even if some bystanders decide to appear in a photograph, their permissions might be limited to the photographer only. The permission, by default, might not extend to other bystanders being able to see them as well. In such cases, the bystanders could additionally specify, along with their original consent sent to the photographer, whether they also consent to appearing in photographs sent to others.

Overall, we remain confident that the combination of EnCore with I-Pic is a powerful one, and can enable workflows that would significantly improve user experience, both, for the bystander and the photographer. Additional privacy challenges that may arise in these new workflows can be mitigated by additional rounds of communication between bystanders and photographers. These additional communication steps can easily be automated and can be also integrated in existing workflows on a user’s device, to keep the cognitive overhead of these steps low.

6.2 Policy enforcement on the viewer

In I-Pic bystanders must either specify their privacy policies a priori before a photograph is captured, or communicate them afterwards using the combination of EnCore and I-Pic (as described in Section 6.1). In either case, privacy policies are communicated to the photographer who (or their capture device) then applies them to the captured photograph.

A possible extension to this framework could be to push the enforcement of privacy policies to the software used for viewing a photograph (e.g. a web browser), which runs on the devices of end users viewing the photograph. In this design, for each successfully matched face in the captured photograph a web url will be added as metadata to the photograph. Viewing software will contact this url to obtain the latest privacy policy of a captured bystander and apply those before displaying the photograph. Bystanders can dynamically update their privacy policies by simply updating the information available at these urls.

A straight forward implementation of this design would be to have the viewing software carry out the secure matching step everytime it wants to display the picture. Doing so will require the photographer’s device to attach all the information received over bluetooth (at the time a photograph was captured) as metadata to the photograph.

Another approach could be to have the photographer’s device perform the secure matching step. Upon a successful match, secure matching will return a url instead of the actual privacy preference of that bystander. This url is then attached as metadata to the captured photograph. The viewing software will contact this url to obtain the most recent privacy policy. The url returned at the end of the secure matching step is generated by the bystander at the time of registering with the a bystander agent. To avoid being used as an identifier for the bystander this url should be changed over time.

6.3 Extending I-Pic to Video and Audio

The I-Pic architecture provides a pluggable framework for extending I-Pic to beyond still images. Below we give a brief description of how I-Pic can be extended to protect privacy of bystanders in video and audio recordings.

Video: A straight-forward approach to extend I-Pic to process video will be to run I-Pic’s face/head detection and recognition algorithms for each frame of a video clip. The visual signatures broadcast by bystanders in I-Pic’s current architecture would be used without any modifications to recognize detected faces in each frame. Although this approach is easy to integrate in I-Pic, in general executing state-of-the-art object detection networks on individual video frames would incur very high computational cost for most applications, particularly for mobile devices.

A better approach for detecting objects in videos would be to exploit data redundancy & continuity in adjacent video frames to reduce per-frame computation cost. Detecting such continuities in adjacent frames, referred to as estimating “Optical Flow”, is a fundamental task in video analysis. It has been studied for decades [132] and has been used extensively in video compression algorithms.

The system proposed in [133] detects objects in videos using a hybrid approach that combines a state-of-the-art still image object detector with optical flow information. It applies an image recognition network on sparse key frames and propagates the deep feature maps from key frames to other frames via an optical flow field. This flow field is estimated using algorithms such as [134, 135]. Using this hybrid architecture, [133] can achieve upto a 10x speed up over approaches that do not use optical flow fields.

Audio: Integrating an audio processing pipeline in I-Pic would require the following two computation steps to be carried out on the photographer’s device: 1) Speaker Diarization: given an audio recording, identifying audio segments where only one speaker is talking, and then clustering together segments coming from the same speaker, and 2) Speaker recognition: matching the speaker for each cluster to the audio signatures received from bystanders. These two steps are akin to the face detection and face recognition steps of I-Pic’s image processing pipeline.

Speaker Diarization has been extensively studied [136]. During this step, the audio recording is first segmented into small portions where only one speaker is talking. Feature vectors are extracted for each segment which are used to cluster together segments coming from the same speaker. For the second step, i.e., speaker recognition, feature vectors extracted during the diarization step can be used to identify the speaker by comparing these feature vectors with pre-trained speaker models. In case of I-Pic these pre-trained models will correspond to audio signatures received over bluetooth.

An example of such an architecture is the system presented in [137]. The system segments the audio recording in 1 second chunks, computes the frequency spectrum for each segment as an image, and then extracts feature vectors from this image using a convolutional neural network. These features vectors are then used for both clustering audio segments, and speaker identification.

Concurrently talking speakers create an additional challenge for the speaker diarization step. Handling these cases will additionally require, for each audio segment, computing the number of concurrently talking speakers and, if needed, isolating audio for individual speakers using “Audio Source Separation” techniques [138].

6.4 Using a trusted photographer’s agent

In I-Pic the cloud agents are assumed to be semi-honest. Hence the computer vision tasks that are performed on the photographer’s device are not offloaded to the photographer’s agent. This is because offloading these tasks will require sharing sensitive information, i.e. the captured photograph, with the photographer’s agent.

If required I-Pic can easily be extended to offload computer vision tasks to a trusted agent, such as photographer’s own desktop/server machine. In this design the captured photograph would be uploaded to the trusted photographer’s agent, which would perform face/head detection, recognition, and initiate secure matching. Offloading these steps will significantly decrease energy consumption on the photographer’s device.

One downside to this design is that the photographer will have to manage this trusted agent on their own. This will require keeping this agent accessible from the Internet and also keeping it updated with latest security patches. Furthermore, the bystander’s agent may require remote attestation of the software running on photographer’s agent before serving a request. This additional step will further increase the logistical overhead of managing a trusted photographer’s agent.

6.5 Using ARM TrustZone to extend I-Pic’s threat model

In I-Pic the operating system on photographer’s device is trusted to not release the captured photograph until it is fully processed, i.e., until secure matching has finished and privacy preferences have been applied to the photograph.

This requirement for trusting the operating system could be relaxed by using trusted hardware, such as ARM TrustZone [130]. TrustZone would allow enforcing privacy preferences on the captured photograph even if the operating system on the photographer’s device is compromised or if the device is rooted. This design assumes that the photographer uses a device that runs I-Pic and is ARM TrustZone compliant.

ARM TrustZone is a set of hardware security extensions that supports isolation of two “worlds” of execution: non-secure and secure, and allows for dynamic partitioning of the hardware into secure and non-secure components. Each processor core executes in the context of a single world at any time; a core can “switch” worlds using a privileged instruction (and, if configured, upon exceptions or interrupts). All accesses to memory and I/O devices are tagged with an additional bit, the ‘NS’ bit, which specifies whether the access was issued while the core was in non-secure mode. Components in the

system (e.g., bus and memory controllers) can be configured, in hardware, to only allow secure accesses.

Essentially, using ARM TrustZone, an isolated, trusted OS kernel could be run in secure mode, and control all memory/peripheral accesses and interrupts received by the non-secure kernel. In case of I-Pic, code responsible for enforcing privacy preferences on the captured photograph could be executed in secure mode as of part of the trusted kernel, which will be invoked as soon as a photograph is captured.

ARM TrustZone could also be used to reliably disable the camera of a device while it is operating within a restricted space. For e.g., disabling the smartphone camera of outside guests visiting the premises of a private organization [139, 140]. These approaches could be used to prevent the capture of openly visible sensitive information (e.g. information appearing on whiteboards or computer screens) within private premises without requiring guests to completely surrender their smartphones.

6.6 Future work

In this section, we discuss two additional future work directions for I-Pic: 1) how to best obscure someone's identity in a photograph, and 2) how to accommodate requests to override users' privacy preferences from authorities, journalists, and professional photographers.

6.6.1 Obscuring mechanisms

As a straw-man design for I-Pic's initial prototype, we chose to blur out faces of bystanders in order to hide their identity in a photograph. Although blurring faces is a commonly used technique, it suffers from two major drawbacks. First, blurring offers limited privacy, since it might still be possible for humans or computers to accurately identify people [141] based on their clothing, hairstyle, background, and their body structure. Second, blurring a small portion of a photograph, such as a face, creates a visually identifiable patch that significantly degrades the aesthetic quality of a photograph. In this section, we discuss alternative mechanisms for obscuring captured bystanders, and also describe the properties an ideal obscuring mechanism should have.

The ideal obscuring mechanism needs to strike the right balance between two often conflicting requirements. First, the ideal mechanism should offer strong privacy. That is, it should ensure that the probability a human or computer can accurately identify an obscured person is either zero or a very low value. At the same time, using such a mechanism should not introduce any visually identifiable artifacts that degrade the aesthetic quality of a photograph, or make it appear structurally implausible.

For example, completely removing a bystander by replacing them with the background offers strong privacy. It is virtually impossible to directly² identify that bystander as their visual appearance has been completely removed from a photograph. This approach should work well for removing isolated bystanders who do not feature prominently in a photograph. The same mechanism, on the other hand, will not be suitable for removing one of two bystanders who are hugging each other in the captured photograph. Doing so would leave the remaining bystander in an odd pose, and might make the photograph appear structurally implausible. In this particular scenario, in-place replacing the face of the obscured bystander by another face might be a better option.

This example highlights the difficulty of constructing an ideal obscuring mechanism that provides strong privacy without degrading the aesthetic quality of a photograph. Furthermore, it may be that no single mechanism is ideal for all photographs, and even for a single photograph, depending on its structure, a combination of techniques might be required. In the following paragraphs we describe some related work for in-place replacement of faces, and for completely removing bystanders from photographs.

In-place face replacement: The Controllable Face Privacy [142] system enables users to alter images of faces by selectively changing semantic attributes, such as age, gender, or, ethnicity of a face. One or more attributes can be changed at a time, while keeping other attributes unchanged. For example, the system can change the gender of a face while keeping the ethnicity the same as before. I-Pic could potentially use this system as an aesthetically pleasing alternative to blurring faces. As of now, this system is limited to altering frontal faces only, and it tends to synthesize distorted faces if more than two attributes are changed at a time. More recently, Karras et al. demonstrated a system [143] that could synthesize photo-realistic fake facial images that are free from any distortions. Using a dataset of 250 thousand frontal face images of celebrities, the authors trained a generative deep neural network capable of producing fake face images that were indistinguishable from actual photographs. While this system produces surprisingly realistic images, it is still limited to synthesizing frontal faces only. Both the systems described above will have to be extended to synthesize faces in arbitrary poses before they can be used as obscuring mechanisms in I-Pic. Furthermore, these systems could also be extended to replace the entire body of a bystander rather than just faces, to enhance the privacy offered by these systems.

Completely removing a bystander: Instead of altering the face (and the body) of a bystander, in many scenarios a viable option might be to simply remove a bystander entirely. The *Content-Aware Fill tool* [144], offered by Adobe Photoshop, allows users

²It might still be possible to indirectly infer a bystander's presence and their identity, given additional contextual information.

to remove distracting objects from a photograph by automatically extrapolating the background to replace the object. The tool can remove objects without introducing any identifiable image processing artifacts, particularly if the object is not overlapping with other objects, and is in front of a generally flat background. When the background contains many details, the tool often produces identifiable distortions in the photograph. More recently, Yang et al. [145] demonstrated a deep learning based approach for high-resolution image inpainting that produces aesthetically better results, even in difficult cases where the *Content-Aware Fill* tool fails. Image inpainting refers to the process of filling holes in images with semantically plausible and context-aware details. Preliminary results presented by the authors are encouraging, and this system could be investigated further to study its performance on images from our dataset, and to measure the energy overhead it would impose on the I-Pic prototype.

Both approaches presented above — in-place replacement of faces and completely removing bystanders — could be used in I-Pic as obscuring mechanisms. It is unlikely that either one of these approaches will work well for all photographs. The ideal obscuring mechanism might require a combination of both these techniques, and will also require learning to decide which technique to use for removing a particular bystander.

6.6.2 Requests to override user preferences

There could be scenarios where authorities might wish to obtain un-edited versions of photographs, overriding privacy preferences of captured bystanders. For example, law enforcement agencies might request photographs from users who were present near a crime scene. An employer could ask their employees for photographs to investigate cases of misconduct within company premises. Journalists and professional photographers might also need access to unedited versions of photographs they have captured. There can be many more scenarios where one might be interested in obtaining unedited versions of photographs. Local laws would ultimately govern who has the authority to make these requests.

Nevertheless, if legislation requires it, future versions of I-Pic could be extended to include a mechanism for obtaining unedited versions of photographs. One possible mechanism might be to additionally store, on a user's device, an unedited version of a photograph that is encrypted with a user-specific key provided by a key-escrow [146]. The encryption key could be released by the escrow to authorities after they have obtained proper authorization.

Deploying trusted key-escrows remains an open research problem [147, 148] that is beyond the scope of this thesis. These challenges are not specific to I-Pic. Any viable solution for deploying trusted key-escrows will benefit all applications that might be legislatively required to cooperate with authorities.

Chapter 7

Conclusion

Mobile devices are capable of collecting a detailed record of users' personal information such as a trace of their location, online and offline activities, and social encounters, including an audiovisual record. This information, though extremely useful for enabling new apps, is also highly sensitive and private. Such a record is subject to numerous privacy risks as described by Aditya et al. [1]. In this thesis, we have investigated and built systems to mitigate two such privacy risks.

In the first project, I-Pic, we investigated risks to users' privacy which arise due to a user's audiovisual appearance being inadvertently captured and shared (as photographs and videos) by other nearby users, without the captured user being aware of it. To mitigate this risk, we built and deployed I-Pic, a trusted software platform that integrates digital capture with user-defined privacy. I-Pic allows users to respect each others' individual and situational privacy preferences, without giving up the spontaneity, ubiquity, and flexibility of digital capture. The I-Pic design and prototype demonstrates that the technical impediments for privacy-compliant imaging can be reasonably overcome using current hardware platforms. I-Pic leverages cutting-edge face/head detection and recognition technology, which is often perceived as a threat to privacy, to instead increase a user's privacy regarding digital capture. Furthermore, our evaluation also shows that future advances in mobile platform hardware and computer vision techniques will directly benefit I-Pic further improving the accuracy of its privacy enforcement, without compromising its energy efficiency.

In the second project, EnCore, we presented a platform for building mobile social applications based on secure encounters. EnCore can support a wide range of event-based communication primitives for mobile social apps, with strong security and privacy guarantees, without requiring a trusted provider, and while integrating with existing communication, storage, and OSN services. As part of our evaluation, we conducted multiple user deployments of *Context*, an app based on EnCore for event based communication, sharing and collaboration. We received favorable user feedback

during these deployments, which gives us confidence that EnCore can serve as a platform for developing social apps in a privacy-compliant manner.

Finally, we also explored the possibility of integrating the two projects presented in this thesis. Specifically, we discussed how extending I-Pic with EnCore's encounter-based communication would create a powerful new combination, one that would enable us to go beyond the current capabilities of I-Pic. We show that such a combination would enable completely new ways of specifying & communicating privacy preferences that do not exist currently, and which would further improve user experience, both for the bystander and the photographer. These novel workflows could also be useful for compliance with governmental privacy regulations, such as, GDPR [11] and AB375 [12].

Both I-Pic and EnCore provide platforms for building mobile apps in a privacy-compliant manner that puts users in control of what personal information is collected, and how it is shared. The primary contribution of this thesis is to demonstrate, through actual deployments of applications built using these platforms, that one can preserve most of the functionality of spontaneous image capture and build mobile social applications without giving up privacy.

Even though a single platform alone may not be able to provide an ideal end-to-end privacy-preserving infrastructure, we remain confident that technical innovations that mitigate specific risk vectors will not only provide a strong basis for a broader societal conversation about the value of user privacy, but will also be needed for future mobile and wearable technology to be broadly accepted by users.

Appendices

Appendix A

I-Pic User Survey

In this appendix we reproduce the user survey conducted during the I-Pic project, described in Section 3.3. The survey is also available online at <http://goo.gl/forms/6tGG0YmFFG>.

Being photographed as a bystander

This survey is about your preferences when you are photographed as a bystander. The questions will present you with common situations and ask you about your photo preferences when you happen to be photographed by a nearby user.

A further description of our research project is provided towards the end of the survey before the survey form is finally submitted.

How much time will it take to complete this survey

5 - 7 minutes.

Privacy of the data collected in this survey

The survey will be conducted anonymously and the only private information we will inquire will be demographic information such as, your age-group, sex, nationality, etc. All individual responses from the survey will be kept confidential and only aggregate statistical information will be derived and published. The participation in the survey, and providing the private information, are both voluntary. Your responses will only be made available to the researchers when the form is finally submitted.

Contact Information

This survey is being conducted by Paarijaat Aditya, Rijurekha Sen and Peter Druschel at Max Planck Institute for Software Systems, Germany, Bobby Bhattacharjee at University of Maryland and Tong Tong Wu at University of Rochester. For further information, please contact the authors at ipic@mpi-sws.org or +49-681-9303-9122.

This survey has been reviewed according to the University of Maryland, College Park IRB procedures for research involving human subjects.

If you have questions about your rights as a research participant or wish to report a research related injury, please contact: Institutional Review Board, University of Maryland, irb@umd.edu, 301-405-4212.

By clicking "Continue", you agree to our survey data privacy statement.

Continue »

 33% completed

Being photographed as a bystander

* Required

An example scenario

Smartphones and wearable devices have enabled spontaneous photography at all places and at all times. Here is an example photograph, whose primary subject is the car. But at least two female passengers are clearly recognizable in the image. This survey wishes to capture your sentiments regarding privacy in situations like this.



Situations and desired privacy actions

This section tries to capture your desired privacy actions in different scenarios. Please imagine yourself in these different situations, and choose the privacy action you would be most comfortable with.

The possible privacy actions are:

- (A) I agree to be captured in any photograph.
- (B) I agree to be captured, but please send me a copy of any photograph that includes me.
- (C) Please obscure my appearance in any photograph that includes me.
- (D) I can decide my preference only after I see the photograph.
- (E) I do not wish to be captured in any photograph.

When I am *

	A	B	C	D	E
engaging in a daily outdoor activity (e.g. walking, cycling, going to market places, etc.)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
in a restaurant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
at a private gathering with friends or family (e.g. birthdays, weddings, etc.)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
using public transport	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
at the beach	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
in a bar or a nightclub	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
at a hospital	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
at a public gathering (e.g. exhibitions, concerts, movies, etc.)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
at a gym	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
at my workplace	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
at a place of worship	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



Other factors affecting your privacy choices

Regardless of the specific situation, how would the following factors affect your comfort at being photographed?

In each case, the choices are:

- (A) I will feel much more comfortable
- (B) I will feel a bit more comfortable
- (C) I will feel the same
- (D) I will feel a little less comfortable
- (E) I will feel much less comfortable



When *

	A	B	C	D	E
the photograph may be posted in a forum with restricted membership (e.g. company/university mailing list)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I am photographed while I am with strangers	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The photographer is an acquaintance	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
the photograph may be published online without my knowledge (e.g. social networks)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The photographer is a stranger	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
the photograph may be published online and I am notified afterwards (e.g. social networks)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I am photographed while I am with acquaintances	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The photographer is a professional photographer (e.g. wedding photographer, journalist, artist)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
the photograph will be limited to personal use by the photographer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
There are minor children in your vicinity who might also be photographed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Other contexts where your privacy matters


Are there any other venues or activities, where you have particular privacy desires regarding image capture? Please add any such additional concerns below.

Your answer



As a photographer, would you like to respect the privacy preferences of people around you?

- Yes, I care about others' privacy choices.
- Yes, provided the aesthetics of the photograph is good.
- Yes, provided the overhead of privacy aware photography is low.
- No, I do not care about privacy preferences of others.

 Page 2 of 3

BACK

NEXT

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#)

Google Forms



Being photographed as a bystander

Demographic information

Please fill in the following optional demographic information. All individual responses from the survey will be kept confidential and only aggregate statistical information will be published.

Age group

- less than 20 years
- 20 - 30 years
- 30 - 40 years
- 40 - 50 years
- more than 50 years

Gender

Your answer

Nationality

Your answer

Education

- High school graduate, diploma or the equivalent (for example: GED)
- Some college credit, no degree
- Trade/technical/vocational training
- Associate degree
- Bachelor's degree
- Master's degree
- Professional degree
- Doctorate degree
- Other: _____

What our project is about

The project explores privacy concerns of subjects who happen to be photographed by third parties using image capture devices like smart phones, smart glasses, and other wearable devices with integrated cameras. The data collected from the survey will be used to inform the design of a mobile platform that seeks to automatically respects the privacy preferences of subjects captured in images. The platform will allow users to specify their privacy choices in different situations, and have the choices securely communicated to any nearby image capture devices, without revealing any personally identifiable information about the user. The photographer's device will honor the received policies by editing the captured image accordingly (e.g., obscure the faces of captured subjects according to their wishes).

Thanks a lot for your valuable time!

 Page 3 of 3

BACK

SUBMIT

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service

Google Forms



Bibliography

- [1] Paarijaat Aditya, Bobby Bhattacharjee, Peter Druschel, Viktor Erdélyi, and Matthew Lentz. Brave new world: Privacy risks for mobile users. In *Proceedings of the Workshop on Security and Privacy Aspects of Mobile Environments (SPME 2014)*, 2014.
- [2] Natasha Lomas. WhatsApp's privacy U-turn on sharing data with Facebook draws more heat in Europe. <https://techcrunch.com/2016/09/30/whatsapps-privacy-u-turn-on-sharing-data-with-facebook-draws-more-heat-in-europe/>.
- [3] Google Glass. https://en.wikipedia.org/wiki/Google_Glass.
- [4] Google will stop selling Glass next week. <http://time.com/3669927/google-glass-explorer-program-ends/>.
- [5] Google fires engineer for violating privacy policies. <http://www.physorg.com/news203744839.html>. Last accessed: September 2012.
- [6] Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. "you might also like: " privacy risks of collaborative filtering. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, 2011.
- [7] Keir Thomas. Microsoft cloud data breach heralds things to come. http://www.pcworld.com/article/214775/microsoft_cloud_data_breach_sign_of_future.html, 2010.
- [8] Liana B. Baker and Jim Finkle. Sony PlayStation suffers massive data breach. <http://www.reuters.com/article/2011/04/26/us-sony-stoldendata-idUSTRE73P6WB20110426>, 2011.
- [9] Fahmida Y. Rashid. Epsilon data breach highlights cloud-computing security concerns. <http://www.eweek.com/c/a/Security/Epsilon-Data-Breach-Highlights-Cloud-Computing-Security-Concerns-637161/>, 2011.

- [10] Matthew Lentz, Viktor Erdelyi, Paarijaat Aditya, Elaine Shi, Peter Druschel, and Bobby Bhattacharjee. SDDR: Light-Weight Cryptographic Discovery for Mobile Encounters. <http://www.cs.umd.edu/projects/encore>.
- [11] EU. Eu general data protection regulation (gdpr). <https://www.eugdpr.org>.
- [12] Ab-375 privacy: personal information: businesses. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375.
- [13] T. Chou and C. Orlandi. The simplest protocol for oblivious transfer. In *LATINCRYPT*, 2015.
- [14] Peter Snyder. Yao’s garbled circuits: Recent directions and implementations. https://www.cs.uic.edu/pub/Bits/PeterSnyder/Peter_Snyder_-_Garbled_Circuits_WCP_2_column.pdf.
- [15] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In *Advances in Cryptology (CRYPTO)*, 2003.
- [16] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. Fairplay - a secure two-party computation system. In *13th USENIX Security Symposium*, 2004.
- [17] A. Rohrbach, M. Rohrbach, S. Tang, S. Joon Oh, and B. Schiele. Generating descriptions with grounded and co-referenced people. In *CVPR*, 2017.
- [18] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *ECCV*, 2014.
- [19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.
- [20] D. Lowe. Distinctive image features from scale-invariant keypoints. In *IJCV*, 2004.
- [21] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
- [22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, , and A. Zisserman. The pascal visual object classes (voc) challenge. In *IJCV*, 2010.
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012.
- [24] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. Imagenet large scale visual recognition competition 2012 (ilsvrc2012).

- [25] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [26] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.
- [27] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. In *IJCV*, 2013.
- [28] R. Girshick. Fast r-cnn. In *ICCV*, 2015.
- [29] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *arXiv preprint arXiv:1605.06409v2*, 2016.
- [30] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and Kevin Murphy. Speed-accuracy trade-offs for modern convolutional object detectors. In *CVPR*, 2017.
- [31] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.
- [32] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.
- [33] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *arXiv preprint arXiv:1703.06870*, 2017.
- [34] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In *NIPS*, 2015.
- [35] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results.
- [36] <http://caffe.berkeleyvision.org/>.
- [37] http://www.huffingtonpost.com/2013/11/27/lost-lake-cafe-google-glass_n_4350039.html.
- [38] Franziska Roesner, David Molnar, Alexander Moshchuk, Tadayoshi Kohno, and Helen J. Wang. World-driven access control for continuous sensing. In *ACM Conference on Computer and Communications Security (CCS)*, 2014.
- [39] Nisarg Raval, Animesh Srivastava, Ali Razeen, Kiron Lebeck, Ashwin Machanavajjhala, and Landon P. Cox. What you mark is what apps see. In

- ACM International Conference on Mobile Systems, Applications, and Services (Mobisys)*, 2016.
- [40] Cheng Bo, Guobin Shen, Jie Liu, Xiang-Yang Li, Yongguang Zhang, and Feng Zhao. Privacy.tag: Privacy concern expressed and respected. In *ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2014.
- [41] Roberto Hoyle, Robert Templeman, Steven Armes, Denise Anthony, David Crandall, and Apu Kapadia. Privacy behaviors of lifeloggers using wearable cameras. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp)*, 2014.
- [42] Tamara Denning, Zakariya Dehlawi, and Tadayoshi Kohno. In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies. In *ACM Conference on Human Factors in Computing Systems (CHI)*, 2014.
- [43] Jaeyeon Jung and Matthai Philipose. Courteous glass. In *UPSIDE, Workshop at ACM International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp)*, 2014.
- [44] Loris D'Antoni, Alan Dunn, Suman Jana, Tadayoshi Kohno, Benjamin Livshits, David Molnar, Alexander Moshchuk, Eyal Ofek, Franziska Roesner, Scott Saponas, Margus Veanes, and Helen J. Wang. Operating system support for augmented reality applications. In *Workshop on Hot Topics in Operating Systems (HotOS)*, 2013.
- [45] Suman Jana, Arvind Narayanan, and Vitaly Shmatikov. A scanner darkly: Protecting user privacy from perceptual applications. In *IEEE Symposium on Security and Privacy*, 2013.
- [46] Suman Jana, David Molnar, Alexander Moshchuk, Alan Dunn, Benjamin Livshits, Helen J. Wang, and Eyal Ofek. Enabling fine-grained permissions for augmented reality applications with recognizers. In *Usenix Security Symposium*, 2013.
- [47] Christopher Snowton, Jacob R. Lorch, David Molnar, Stefan Saroiu, and Alec Wolman. Zero-effort payments: Design, deployment, and lessons. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2014.

- [48] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [49] Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Describing people: Poselet-based attribute classification. In *International Conference on Computer Vision (ICCV)*, 2011.
- [50] Ning Zhang, Manohar Paluri, Marc'Aurelio Ranzato, Trevor Darrell, and Lubomir D. Bourdev. PANDA: pose aligned networks for deep attribute modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [51] Zhou Lingli and Lai Jianghuang. Security algorithm of face recognition based on local binary pattern and random projection. In *International Conference on Computational Intelligence (ICCI)*, 2010.
- [52] Yongjin Wang and Konstantinos N. Plataniotis. An analysis of random projection for changeable and privacy-preserving biometric verification. *IEEE Transactions on Systems, Man, and, Cybernetics: part B: CYBERNETICS*, Vol. 40, No. 5, 2010.
- [53] Per Hallgren, Martin Ochoa, and Andrei Sabelfeld. Innercircle: A parallelizable decentralized privacy-preserving location proximity protocol. In *Proceedings of the 13th Annual Conference on Privacy, Security and Trust (PST)*, 2015.
- [54] Nisarg Raval, Animesh Srivastava, Kiron Lebeck, Landon P. Cox, and Ashwin Machanavajjhala. Markit: Privacy markers for protecting visual secrets. In *UPSIDE, Workshop at ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2014.
- [55] S. Joon Oh, R. Benenson, M. Fritz, and B. Schiele. Person recognition in personal photo collections. In *ICCV*, 2015.
- [56] Bart Goethals, Sven Laur, Helger Lipmaa, and Taneli Mielikainen. On private scalar product computation for privacy-preserving data mining. In *7th Annual International Conference in Information Security and Cryptology (ICISC)*, 2004.
- [57] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (FOCS)*, 1986.

- [58] Terence Sim and Li Zhang. Controllable face privacy. In *The 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2015.
- [59] Antonio Criminisi, Patrick Perez, , and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. In *IEEE Transactions on image processing, vol. 13, no. 9, September, 2004*.
- [60] X. Zhu and D. Ramanan. Face detection, pose estimation and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [61] Ning Zhang, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir Bourdev. Beyond frontal faces: Improving person recognition using multiple cues. In *CVPR*, 2015.
- [62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. *CVPR*, 2009.
- [63] Gary B. Huang Erik Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. Technical Report UM-CS-2014-003, University of Massachusetts, Amherst, May 2014.
- [64] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 2008.
- [65] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology (EUROCRYPT)*, 1999.
- [66] Yan Huang, Lior Malka, David Evans, and Jonathan Katz. Efficient privacy-preserving biometric identification. In *18th Network and Distributed System Security Conference (NDSS)*, 2011.
- [67] Moni Naor and Benny Pinkas. Computationally secure oblivious transfer. In *Journal of Cryptology*, 2005.
- [68] Yehuda Lindell. Fast cut-and-choose based protocols for malicious and covert adversaries. In *Advances in Cryptology (CRYPTO)*, 2013.
- [69] Google. Project Tango Tablet Development Kit. https://store.google.com/product/project_tango_tablet_development_kit.
- [70] NVIDIA. CUDA. http://www.nvidia.com/object/cuda_home_new.html.

- [71] Jose-Luis Lisani, Ana-Belen Petro, and Catalina Sbert. Color and Contrast Enhancement by Controlled Piecewise Affine Histogram Equalization. *Image Processing On Line*, 2:243–265, 2012. <http://dx.doi.org/10.5201/ipol.2012.lps-pae>.
- [72] sh1r0. Caffe-Android-Lib. <https://github.com/sh1r0/caffe-android-lib>.
- [73] Might Be Evil. <http://mightbeevil.org/>.
- [74] <https://cvhci.anthropomatik.kit.edu/~baeuml/projects/a-universal-labeling-tool-for-computer-vision-sloth/>.
- [75] Monsoon Power Monitor. <https://www.msoon.com/LabEquipment/PowerMonitor>.
- [76] Nvidia. Nvidia Shield Tablet K1. <https://shield.nvidia.com/tablet/k1>.
- [77] NVIDIA Jetson TX2 Delivers Twice the Intelligence to the Edge. <https://devblogs.nvidia.com/parallelforall/jetson-tx2-delivers-twice-intelligence-edge/>.
- [78] J. Tu, A. D. Amo, Y. Xu, L. Guan, M. Chang, and T. Sebastian. A fuzzy bounding box merging technique for moving object detection. In *NAFIPS*, 2012.
- [79] Y. Liu, H. Li², and X. Wang. Learning deep features via congenerous cosine loss for person recognition. In *arXiv preprint arXiv:1702.06890*, 2017.
- [80] Jetson TX2 Developer Kit. <https://developer.nvidia.com/embedded/buy/jetson-tx2-devkit>.
- [81] Keysight B2961A 6.5 Digit Low Noise Power Source. <http://www.keysight.com/en/pd-2149890-pn-B2961A/65-digit-low-noise-power-source>.
- [82] Highlight. <http://highlig.ht/>. Last accessed: December 2013.
- [83] Facebook nearby friends. <https://techcrunch.com/2014/04/17/facebook-nearby-friends/>. Last accessed: April 2014.
- [84] Yikyak. <https://www.yikyak.com>. Last accessed: August 2016.
- [85] Whisper. <http://whisper.sh/>. Last accessed: March 2014.
- [86] AllJoyn. <http://www.alljoyn.org>.
- [87] FireChat. <https://itunes.apple.com/us/app/firechat/id719829352?mt=8>. Last accessed: March 2014.

- [88] Justin Manweiler, Ryan Scudellari, and Landon P. Cox. SMILE: Encounter-based trust for mobile social services. In *CCS*, 2009.
- [89] Landon P. Cox, Angela Dalton, and Varun Marupadi. SmokeScreen: Flexible privacy controls for presence-sharing. In *MobiSys*, 2007.
- [90] Lokast. <http://www.lokast.com>.
- [91] Hagggle. <http://www.hagggleproject.org>.
- [92] Jing Su, James Scott, Pan Hui, Jon Crowcroft, Eyal De Lara, Christophe Diot, Ashvin Goel, Meng How Lim, and Eben Upton. Hagggle: seamless networking for mobile applications. In *Proceedings of the 9th international conference on Ubiquitous computing, UbiComp '07*, 2007.
- [93] Ben Dodson, Ian Vo, T.J. Purtell, Aemon Cannon, and Monica Lam. Musubi: disintermediated interactive social feeds for mobile devices. In *Proceedings of the 21st international conference on World Wide Web, WWW '12*, 2012.
- [94] Tile. <http://www.thetileapp.com/>. Last accessed: September 2013.
- [95] P Jappinen, ILMARI Laakkonen, VILLE Latva, and A Hamalainen. Bluetooth device surveillance and its implications. *WSEAS Transactions on Information Science and Applications*, 1:1056–1060, 2004.
- [96] iOS 7 AirDrop. <http://support.apple.com/kb/HT5887>. Last accessed: January 2014.
- [97] Android Beam. <http://developer.android.com/guide/topics/connectivity/nfc/nfc.html#p2p>. Last accessed: June 2013.
- [98] Near Field Communication – Interface and Protocol (ISO/IEC 18092:2013). http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=56692. Last accessed: September 2013.
- [99] Wi-Fi Direct. <http://www.wi-fi.org/discover-and-learn/wi-fi-direct>.
- [100] Friday: automated journal. <http://www.fridayed.com/>. Last accessed: October 2013.
- [101] Memoto: automatic lifelogging camera. <http://memoto.com/>. Last accessed: September 2013.

- [102] Nadav Aharony, Wei Pan, Cory Ip, Inas Khayal, and Alex Pentland. Social fMRI: Investigating and shaping social mechanisms in the real world. *Pervasive Mob. Comput.*, 7(6), December 2011.
- [103] Yue-Hsun Lin, Ahren Studer, Hsu-Chin Hsiao, Jonathan M. McCune, King-Hang Wang, Maxwell Krohn, Phen-Lan Lin, Adrian Perrig, Hung-Min Sun, and Bo-Yin Yang. Spate: small-group pki-less authenticated trust establishment. In *Proceedings of the 7th international conference on Mobile systems, applications, and services, MobiSys '09*, 2009.
- [104] Wolfgang Apolinarski, Marcus Handte, Muhammad Umer Iqbal, and Pedro José Marrón. Secure interaction with piggybacked key-exchange. *Pervasive Mob. Comput.*, 10, February 2014.
- [105] Ben Greenstein, Damon McCoy, Jeffrey Pang, Tadayoshi Kohno, Srinivasan Seshan, and David Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *MobiSys*, 2008.
- [106] Bluetooth Specification Core Version 4.0. https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737. Last accessed: March 2014.
- [107] Saikat Guha, Mudit Jain, and Venkata N. Padmanabhan. Koi: a location-privacy platform for smartphone apps. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, NSDI'12*, 2012.
- [108] Peter Hornyack, Seungyeop Han, Jaeyeon Jung, Stuart Schechter, and David Wetherall. These aren't the droids you're looking for: retrofitting android to protect data from imperious applications. In *CCS*, 2011.
- [109] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying location privacy. In *S&P*, 2011.
- [110] Michaela Goetz and Suman Nath. Privacy-aware personalization for mobile advertising. Technical report.
- [111] Carl A. Gunter, Michael J. May, and Stuart G. Stubblebine. A formal privacy system and its application to location based services. In *Proceedings of the 4th international conference on Privacy Enhancing Technologies, PET'04*, 2005.
- [112] Panos Kalnis, Gabriel Ghinita, Kyriakos Mouratidis, and Dimitris Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE Trans. on Knowl. and Data Eng.*, 19(12), December 2007.

- [113] Baik Hoh, Marco Gruteser, Ryan Herring, Jeff Ban, Daniel Work, Juan-Carlos Herrera, Alexandre M. Bayen, Murali Annamaram, and Quinn Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *Proceedings of the 6th international conference on Mobile systems, applications, and services, MobiSys '08*, 2008.
- [114] Mehedi Bakht, Matt Trower, and Robin Hilary Kravets. Searchlight: Won't you be my neighbor? In *MobiCom*, 2012.
- [115] Bo Han and Aravind Srinivasan. ediscovery: Energy efficient device discovery for mobile opportunistic communications. In *Proceedings of the 20th IEEE International Conference on Network Protocols (ICNP)*, ICNP '12, 2012.
- [116] Arvind Kandhalu, Karthik Lakshmanan, and Rangunathan (Raj) Rajkumar. U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN '10*, 2010.
- [117] Prabal Dutta and David Culler. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In *Proceedings of the 6th ACM conference on Embedded network sensor systems, SenSys '08*, 2008.
- [118] Wei Wang, Vikram Srinivasan, and Mehul Motani. Adaptive contact probing mechanisms for delay tolerant applications. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking, MobiCom '07*, 2007.
- [119] Stefan Saroiu and Alec Wolman. Enabling new mobile applications with location proofs. In *Proceedings of the 10th workshop on Mobile Computing Systems and Applications, HotMobile '09*, 2009.
- [120] Vincent Lenders, Emmanouil Koukoumidis, Pei Zhang, and Margaret Martonosi. Location-based trust for mobile user-generated content: applications, challenges and implementations. In *Proceedings of the 9th workshop on Mobile computing systems and applications, HotMobile '08*, 2008.
- [121] Bryan Ford, Jacob Strauss, Chris Lesniewski-Laas, Sean Rhea, Frans Kaashoek, and Robert Morris. Persistent personal names for globally connected mobile devices. In *Proceedings of the 7th symposium on Operating systems design and implementation, OSDI '06*, 2006.
- [122] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *MobiCom*, 2008.

- [123] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6), nov 1976.
- [124] Paarijaat Aditya, Viktor Erdélyi, Matthew Lentz, Elaine Shi, Bobby Bhattacharjee, and Peter Druschel. Encore: Private, context-based communication for mobile social apps. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '14, pages 135–148, New York, NY, USA, 2014. ACM.
- [125] Mailinator: Free disposable email. <http://mailinator.com/>. Last accessed: January 2014.
- [126] Randy Baden, Adam Bender, Neil Spring, Bobby Bhattacharjee, and Daniel Starin. Persona: an online social network with user-defined privacy. In *Proceedings of the ACM SIGCOMM conference on Data communication*, SIGCOMM '09, 2009.
- [127] Wai-Tian Tan, Mary Baker, Bowon Lee, and Ramin Samadani. The sound of silence. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, 2013.
- [128] Claude Castelluccia and Pars Mutaf. Shake them up!: a movement-based pairing protocol for CPU-constrained devices. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, MobiSys '05, 2005.
- [129] Rene Mayrhofer and Hans Gellersen. Shake well before use: authentication based on accelerometer data. In *Proceedings of the 5th international conference on Pervasive computing*, PERVASIVE'07, 2007.
- [130] ARM. Arm trustzone. <https://www.arm.com/products/security-on-arm/trustzone>.
- [131] Give and get the photos you care about. <https://www.blog.google/products/photos/give-and-get-photos-you-care-about/>.
- [132] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17, August 1981.
- [133] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *CVPR*, 2017.
- [134] Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *arXiv preprint arXiv:1504.06852*, 2015.

- [135] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: dense correspondence across difference scenes. In *ECCV*, 2008.
- [136] S. E. Tranter and D. A. Reynolds. An overview of automatic speaker diarization systems. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5), Sept 2006.
- [137] Y. Lukic, C. Vogt, O. Durr, and T. Stadelmann. Speaker identification and clustering using convolutional neural networks. In *MLSP*, pages 1–6, Sept 2016.
- [138] Monali Pimpale, Shanthi Therese, and Vinayak Shinde. A survey on: Sound source separation methods. *International Journal of Computer Engineering In Research Trends (IJCERT)*, 3, Nov 2016.
- [139] Ferdinand Brasser, Daeyoung Kim, Christopher Liebchen, Vinod Ganapathy, Liviu Iftode, and Ahmad-Reza Sadeghi. Regulating arm trustzone devices in restricted spaces. In *ACM International Conference on Mobile Systems, Applications, and Services (Mobisys)*, 2016.
- [140] Matthew Lentz, Rijurekha Sen, Peter Druschel, and Bobby Bhattacharjee. Secloak: Arm trustzone-based mobile peripheral control. In *ACM International Conference on Mobile Systems, Applications, and Services (Mobisys)*, 2018.
- [141] S. J. Oh, R. Benenson, M. Fritz, and B. Schiele. Faceless person recognition; privacy implications in social media. In *European Conference on Computer Vision (ECCV)*, 2016.
- [142] L. Zhang T. Sim. Controllable face privacy. In *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2015.
- [143] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *arXiv preprint arXiv:1710.10196*, 2017.
- [144] Content-Aware Fill. <https://research.adobe.com/project/content-aware-fill/>.
- [145] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, 2017.
- [146] Key escrow. https://en.wikipedia.org/wiki/Key_escrow.
- [147] M. Blaze. Key escrow from a safe distance: Looking back at the clipper chip. In *ACSAC*, 2011.

- [148] H. Abelson, R. Anderson, S. M. Bellovin, J. Benaloh, M. Blaze, W. Diffie, J. Gilmore, P. G. Neumann, R. L. Rivest, J. I. Schiller, and B. Schneier. The Risks of Key Recovery, Key Escrow, and Trusted Third-Party Encryption. https://www.schneier.com/academic/archives/1997/04/the_risks_of_key_rec.html. Last accessed: September 2017.